

Internet of Things

Jordi Mongay Batalla

George Mastorakis

Constandinos X. Mavromoustakis

Evangelos Pallis *Editors*

Beyond the Internet of Things

Everything Interconnected

 Springer

Internet of Things

Technology, Communications and Computing

Series editors

Giancarlo Fortino, Rende (CS), Italy

Antonio Liotta, Eindhoven, The Netherlands

More information about this series at <http://www.springer.com/series/11636>

Jordi Mongay Batalla · George Mastorakis
Constandinos X. Mavromoustakis
Evangelos Pallis
Editors

Beyond the Internet of Things

Everything Interconnected

 Springer

Editors

Jordi Mongay Batalla
National Institute of Telecommunications
Warsaw
Poland

Constandinos X. Mavromoustakis
Department of Computer Science
University of Nicosia
Nicosia
Cyprus

George Mastorakis
Department of Commerce and Marketing
Technological Educational Institute of Crete
Crete
Greece

Evangelos Pallis
Technological Educational Institute of Crete
Crete
Greece

ISSN 2199-1073

Internet of Things

ISBN 978-3-319-50756-9

DOI 10.1007/978-3-319-50758-3

ISSN 2199-1081 (electronic)

ISBN 978-3-319-50758-3 (eBook)

Library of Congress Control Number: 2016959252

© Springer International Publishing AG 2017

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature

The registered company is Springer International Publishing AG

The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*To Sara, whose uncertain smile makes us so
happy*

Jordi Mongay Batalla

*To my son Nikos, who always makes me
proud*

George Mastorakis

*To my wife Afrodyte for her unconditional
understanding and support*

Constandinos X. Mavromoustakis

*To Vasiliki and Meletios for the time I did not
spend with them, working for a book that they
will never read*

Evangelos Pallis

Contents

Part I Challenges Beyond the Internet of Things

Context-Aware Systems: Technologies and Challenges in Internet of Everything Environments	3
Everton de Matos, Leonardo Albernaz Amaral and Fabiano Hessel	
Enabling User Context Utilization in the Internet Communication Protocols: Motivation, Architecture and Examples	29
Yu Lu	
Security Challenges of the Internet of Things	53
Musa G. Samaila, Miguel Neto, Diogo A.B. Fernandes, Mário M. Freire and Pedro R.M. Inácio	

Part II Technologies for Connecting Everything

A Novel Machine to Machine Communication Strategy Using Rateless Coding for the Internet of Things	85
Boulos Wadiah Khoueiry and M. Reza Soleymani	
Energy-Efficient Network Architecture for IoT Applications	119
P. Sarwesh, N. Shekar V. Shet and K. Chandrasekaran	
ID-Based Communication for Access to Sensor Nodes	145
Mariusz Gajewski, Waldemar Latoszek, Jordi Mongay Batalla, George Mastorakis, Constandinos X. Mavromoustakis and Evangelos Pallis	
QoS/QoE in the Heterogeneous Internet of Things (IoT)	165
Krzysztof Nowicki and Tadeus Uhl	

Part III Applicability of Interconnecting Everything

Integration of Internet of Everything (IoE) with Cloud	199
Sarbani Roy and Chandreyee Chowdhury	

Multimodal Low-Invasive System for Sleep Quality Monitoring and Improvement	223
--	-----

Fábio Manoel Franca Lobato, Damares Crystina Oliveira de Resende, Roberto Pereira do Nascimento, André Luis Carvalho Siqueira, Antonio Fernando Lavareda Jacob, Jr. and Ádamo Lima de Santana

On Real Time Implementation of Emotion Detection Algorithms in Internet of Things	243
--	-----

Sorin Zoican

Recognizing Driving Behaviour Using Smartphones	269
--	-----

Prokopis Vavouranakis, Spyros Panagiotakis, George Mastorakis, Constandinos X. Mavromoustakis and Jordi Mongay Batalla

Part IV New Horizons: Large Scenarios

Cloud Platforms for IoE Healthcare Context Awareness and Knowledge Sharing	303
---	-----

Alireza Manashty and Janet Light Thompson

Survey on Technologies for Enabling Real-Time Communication in the Web of Things	323
---	-----

Piotr Krawiec, Maciej Sosnowski, Jordi Mongay Batalla, Constandinos X. Mavromoustakis, George Mastorakis and Evangelos Pallis

Crowd-Driven IoT/IoE Ecosystems: A Multidimensional Approach	341
---	-----

Xenia Ziouvelou, Panagiotis Alexandrou, Constantinos Marios Angelopoulos, Orestis Evangelatos, Joao Fernandes, Nikos Loumis, Frank McGroarty, Sotiris Nikolettseas, Aleksandra Rankov, Theofanis Raptis, Anna Ståhlbröst and Sebastien Ziegler

Improving Quality of Life with the Internet of Everything	377
--	-----

Despina T. Meridou, Maria-Eleftheria Ch. Papadopoulou, Andreas P. Kapsalis, Panagiotis Kasnesis, Athanasios I. Delikaris, Charalampos Z. Patrikakis, Iakovos S. Venieris and Dimitra I. Kaklamani

Introduction

The networked connection of people, things, processes and data is called the Internet of Everything (IoE). It provides high revenues to many companies due to the increase of work efficiency, as well as to the increase of security and comfort of the workers. The sector-specific infrastructures, where the IoE is successfully implemented are smart grid, critical infrastructure management and smart meters, among others. Nonetheless, the increase of revenues is going to multiply in public and private sectors due to IoE deployment together with a big contribution to the well-being of people. IoE is based on near Internet ubiquity and includes three types of connections: machine-to-machine, person-to-machine and person-to-person. Machine-to-machine is closely related to security, including civil security (e.g., security in the road, disaster alert, etc.) and military security. Person-to-machine communication brings an unquestionable increase of well-being in home automation systems but also is fundamental for intelligent parking, patient monitoring and disaster response, among others. At last, person-to-person connection is already changing the inter-personal relations, which are becoming more multimedia and located in the social networks. IoE will increase the scenarios of person-to-person networked communication as, for example, telework, networked learning and telemedicine.

The future of the implementation of the IoE depends on the effective solution to a number of technical challenges that this paradigm introduces. These challenges include sensor capabilities improvement and sensor miniaturization (many hardware companies as Intel and Qualcomm are increasing the research and production of improved sensors and tiny chips for the application in all the aspects of our life), Big Data treatment and efficient remote data management (by introducing new remote management oriented architectures), as well as the open and secure composition of processes, which may be easily implemented into the IoE scenarios. Some initiatives try to build IoE from scratch (e.g., some infrastructures for smart cities proposed in China), but the normal trend is to group together specific use cases of the IoE, cloud computing and all-as-a-service communication frameworks. In fact, the approach of IoE is to find the potential benefits of the interaction of the existing infrastructure, in order to build extensive ecosystems for increasing

the number of services and their value. The backbone of the IoE is the sum of the existing technologies: fiber and mobile high-speed access to the Internet, GPS, multimedia devices (video cameras, end users' terminals), wired and wireless sensor networks, cloud computing. The management of the IoE should be distributed at different layers. Privacy and authorization and authentication should be managed at the application level (i.e., communication between processes). Instead, highly resource requesting security processes should be provided at the network level due to rather low complexity required for sensors and things. All the features related to security and privacy should be controlled by rules and norms at different levels: international and national law, Internet operator's practices, rules of companies, so the security and privacy behavior of the IoE will be the interaction of such rules and norms. Other management and control functionalities will be inserted in the IoE processes in such a way that there will be no difference between processes giving service out of the networked environment (i.e., to the end users) and inside. At last, the high degree of management distribution will be seen as self-capability of IoE management.

In this context, the major subjects of the proposed book cover modeling, analysis and efficient management of information in IoE applications and architectures. It addresses the major new technological developments in the field and will reflect current research trends, as well as industry needs. This book comprises a good balance between theoretical and practical issues, covering case studies, experience and evaluation reports and best practices in utilizing IoE applications. It also provides technical/scientific information about various aspects of IoE technologies, ranging from basic concepts to research-grade material, and including future directions. Scientific research provided in these pages comes from different research projects provided by eminent scientists, one of these projects is IDSECOM project, which is finalizing the activities just in these months.

The book is divided into four parts: (I) Challenges Beyond the Internet of Things, (II) Technologies for Connecting Everything, (III) Applicability of Interconnecting Everything, and (IV) New Horizons: Large Scenarios. In Part I, motivation and challenges of the internet of everything are exposed under the examples of context-awareness and security enhancement. Part II exposes new technologies in all levels: macro, micro and nano for implementing energy-efficient and high-quality communication between devices. At higher level, the Internet of Everything opens new applications thanks to the connectivity with the cloud and ubiquitous of sensors. Novel applications are presented in Part III, whereas Part IV presents extended platforms for connecting everything, including access to cloud, individual processes (e.g., security), and human interaction.

Jordi Mongay Batalla
George Mastorakis
Constandinos X. Mavroumoustakis
Evangelos Pallis

Part I
Challenges Beyond the Internet of Things

Context-Aware Systems: Technologies and Challenges in Internet of Everything Environments

Everton de Matos, Leonardo Albernaz Amaral and Fabiano Hessel

Abstract The Internet of Things (IoT) and Internet of Everything (IoE) paradigms have emerged in the last years, thus generating new challenges in the pervasive computing area. IoT is a computing paradigm that has been recognized for allowing the connection of the physical and virtual worlds by giving processing power to the daily “things”. IoE goes beyond the IoT by breaking the barrier of just “things”. In IoE, the people, data and processes also make part of the connected world. Context awareness has becoming an important feature in IoT and IoE scenarios. Automatic decision making, sensitivity to context, automatic notification to the user, just to name a few, are some examples of situations where a context-aware system is needed in these environments where the characteristics of the data sources are undergoing constant change. In this chapter we present the context-aware definitions and architecture in IoE and its evolution from IoT. Moreover, we present the context-aware life-cycle phases, which is the process done in order to have context information. In addition, we also analyze the current context-aware approaches of IoT/IoE systems, and present some challenges related to context-aware IoE systems.

1 Introduction

In the last years a computing paradigm called Internet of Things (IoT) has gained significant attention. The basic idea of IoT is the pervasive presence around us of a variety of things or objects (e.g., RFID tags, sensors, etc.) that cooperate with their neighbors to reach common goals [3]. By embedding mobile networking and information processing capability into a wide array of gadgets and everyday items

E. de Matos (✉) · L.A. Amaral · F. Hessel
Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, Brazil
e-mail: everton.matos.001@acad.pucrs.br

L.A. Amaral
e-mail: leonardo.amaral@acad.pucrs.br

F. Hessel
e-mail: fabiano.hessel@pucrs.br

enabling new forms of communication between people and things, and between things themselves, the Internet of Things has been adding new dimensions to the world of information and communication technology [5]. It promises to create a world where all the objects around us are connected to the Internet and communicate with each other with minimum human intervention.

Beyond the IoT, the concept of Internet of Everything (IoE) has also gained prominence in network and communication scenarios. In addition to the “Things”, IoE connects people, data, and processes in networks of billions or even trillions of connections [18]. These connections create vast amounts of data, some of these data that we never had access to before. When these data are analyzed and used intelligently, the possibilities seem endless.

There is a common sense that the data providers in IoE environments will generate a lot of data, and they will only be useful if we could analyze, interpret and understand them [47]. In this sense, context-aware computing has played an important role in tackling this challenge in previous paradigms, such as mobile and pervasive computing, which lead us to believe that it would continue to be successful in the IoE as well [39]. Mobile devices can be part of IoE scenarios and their characteristics are constantly changing (e.g., status, location). Context-aware approaches allow us to discover and store context information linked to these devices. In this sense, context-awareness became a fundamental feature of IoE in order to have a fully automated environment and improve the user’s Quality of Experience (QoE) [31].

The concept of context is attached to the information that will be used to characterize the situation of an entity. In this sense, a system becomes context-aware if it uses the context in order to provide new information to user [2]. Taking into account these definitions, an IoE environment needs a context-aware system to be aware of the environment in order to help the user by providing these information in the most useful way. In the context-awareness area, there is a set of methods to build context. These methods are organized in phases that the systems must follow to produce context information that characterizes the context life-cycle of an information [39].

The main contribution of this chapter is to present a discussion about the context-aware systems technologies and challenges in IoE environments in order to provide a view of what can be the best technologies to fit with the necessities of IoE environments and what is the new trends in the area. We will also argue about the context life-cycle, we will show a detailed view of all the phases and the most useful technologies. In addition, we will identify some existing work related to context-awareness in IoE environments and also how we can have a fully functional platform respecting the requirements and challenges of context-aware systems in IoE environments.

The remainder of this paper is organized as follows: Sect. 2 provides the concepts of the IoT evolution into IoE. Section 3 provides a theoretical background about context, we also present the context life-cycle definitions and techniques. Section 4 provides an overview of the characteristics of the systems that produce context information. Section 5 provides a study of some related work. Section 6

shows how context is present in IoE environments, moreover we show the technologies and challenges involving this issue. We conclude this chapter with a summary in Sect. 7.

2 From Internet of Things (IoT) to Internet of Everything (IoE)

Internet of Things (IoT) is a novel computing paradigm that is rapidly gaining space in scenarios of modern communication technologies. The idea of the IoT is the pervasive presence of a variety of things or objects (e.g., RFID tags, sensors, actuators, smart phones, smart devices, etc.), that are able to interact with each other and cooperate with their neighbors to reach common goals through unique addressing schemes and reliable communication media over the Internet [3, 21].

During the past decade, the IoT has gained significant attention in academia as well as industry. The main reasons behind this interest are the capabilities that the IoT will offer [27]. It promises to create a world where all the objects (also called smart objects [30]) around us are connected to the Internet and communicate with each other with minimum human intervention. The ultimate goal is to create “a better world for human beings”, where objects around us know what we like, what we want, what we need, and act accordingly without explicit instructions [17].

The Intranet is being extended to smart things [30] with a higher scalability, pervasiveness, and integration into the Internet Core. This extension is leading to reach a real IoT, where things are first class citizens in the Internet, and they do not need to relay any more on a gateway, middleware, proxy, or broker. IoT drives towards integrating everything into the Internet Core, this trend is the denominated Internet of Everything (IoE). The integration of everything is motivated by the market wish to have all processes remotely accessible through a uniform way [28].

The IoT idea implied other concepts, such as Internet of Service (IoS), Internet of Everything (IoE), Web of Things (WoT), which of course represent the IoT. When we consider the relations M2M (Man to Man), M2T (Man to Thing), M2P (Man to People), P2P (People to People), and D2D (Device to Device), we ultimately reach the IoE [49]. IoE is a new Internet concept that tries to connect everything that can be connected to the Internet, where everything refers to people, cars, televisions (TVs), smart cameras, microwaves, sensors, and basically anything that has Internet-connection capability [1].

The IoE connects people, data, things, and processes in networks of billions or even trillions of connections. These connections create vast amounts of data, some of it data we’ve never had access before. When this data is analyzed and used intelligently, the possibilities seem endless [18].

Today, less than 1% of things that could be connected are connected to the Internet or intelligent systems. Projections show that by 2017, 3.5 billion people will be connected to the Internet, 64% of them via mobile devices [13]. People and connected things will generate massive amounts of data, an estimated 40 trillion

gigabytes, that will have a significant impact on daily life [1]. It will enable faster response times to medical or public safety emergencies and save lives, it will improve the quality of citizen life by providing direct and personal services from the government, and it will uncover new information about how our cities work, thus enabling city leaders to use resources more efficiently and save money while providing superior services. There are three key ways in which the IoE will significantly impact our lives, as described in the following examples [13]:

- **The IoE will automate connections:** Today, people must proactively connect to the network or Internet via mobile devices like smartphones and tablets and to other people on the network via social media websites. Citizens must proactively call a certain phone number for an enterprise complaint or for an emergency. Imagine if people were connected automatically to systems of services instead. Wearable computers in clothing or watches, or sensors in pills that are swallowed, could automatically send patient information to doctors and nurses. This would allow a sick or an elderly person to manage his or her healthcare from home rather than a hospital or nursing home, getting automatic reminders to take medicine or immediate preventive care for changes in health status. For example, weight gain in cardiac patients is often an early indicator of returning heart problems. Connected scales from the home can be used to alert a doctor of a change in patient weight so that quick action can be taken to prevent another heart attack.
- **The IoE will enable fast personal communications and decision making:** Now imagine that intelligence is embedded within sensors or devices. This means the device itself will filter out relevant information and even apply analytics, so in the case of the connected scale, only when a certain threshold of weight gain is crossed will doctors and nurses be alerted. This type of data not only will enable faster, better decision making but also will help government workers, doctors, and citizens more efficiently manage their time. Instead of doctors searching through files or ordering a battery of tests, information would be sent to them directly from patients to help make decisions. Patients will have faster response times from doctors based on such highly personalized information. This is another example of how the Internet of Everything will completely change the types of services that are offered and also how they are delivered to citizens.
- **The IoE will uncover new information:** With the deployment of so many sensors and other information-gathering devices, city managers will be able to understand their city as never before. An interesting example is the use of acoustic sensors that are calibrated to detect gunshots. Some cities in the United States have deployed these sensors in areas of gun violence and discovered some shocking information. Police departments had historically assumed that residents called the police 80% of the time when shots were heard. These police departments were operating on highly inaccurate information about the level of gun violence in certain neighborhoods. With this new information, police can now plan their patrols differently and better target areas to reduce gun violence.

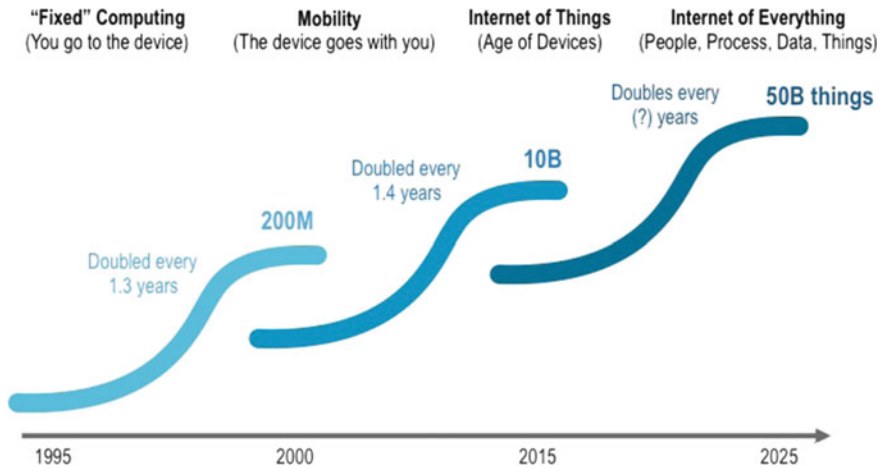


Fig. 1 Internet growth is occurring in waves [32]

As things add capabilities like context awareness, increased processing power, and energy independence, and as more people and new types of information are connected, IoT becomes an Internet of Everything—a network of networks where billions or even trillions of connections create unprecedented opportunities as well as new risks (see Fig. 1, extracted from [32]).

IoE brings together people, process, data, and things to make networked connections more relevant and valuable than ever before—turning information into actions that create new capabilities, richer experiences, and unprecedented economic opportunity for businesses, individuals, and countries (see Fig. 2) [18]. To better understand this definition, we must first break down IoE's individual components.

- People:** In IoE, people will be able to connect to the Internet in innumerable ways. Today, most people connect to the Internet through their use of devices (such as PCs, tablets, TVs, and smartphones) and social networks. As the Internet evolves toward IoE, we will be connected in more relevant and valuable ways. For example, in the future, people will be able to swallow a pill that senses and reports the health of their digestive tract to a doctor over a secure Internet connection. In addition, sensors placed on the skin or sewn into clothing will provide information about a person's vital signs. According to Gartner [32], people themselves will become nodes on the Internet, with both static information and a constantly emitting activity system.
- Data:** With IoT, devices typically gather data and stream it over the Internet to a central source, where it is analyzed and processed. As the capabilities of things connected to the Internet continue to advance, they will become more intelligent by combining data into more useful information. Rather than just reporting raw data, connected things will soon send higher-level information back to machines, computers, and people for further evaluation and decision making. This

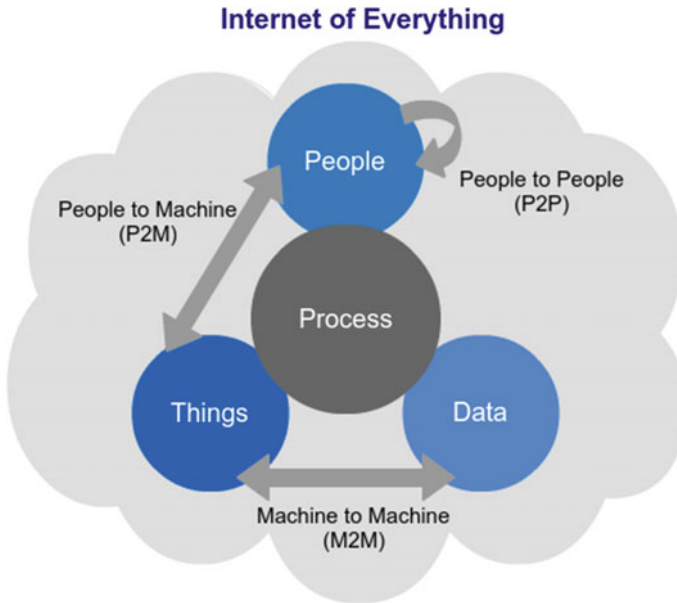


Fig. 2 The what, where, and how of the Internet of Everything

transformation from data to information in IoE is important because it will allow us to make faster, more intelligent decisions, as well as control our environment more effectively.

- **Things:** This group is made up of physical items like sensors, consumer devices, and enterprise assets that are connected to both the Internet and each other. In IoE, these things will sense more data, become context-aware, and provide more experiential information to help people and machines make more relevant and valuable decisions. Examples of “things” in IoE include smart sensors built into structures like bridges, and disposable sensors that will be placed on everyday items such as milk cartons [18].
- **Process:** Process plays an important role in how each of these entities—people, data, and things—work with the others to deliver value in the connected world of IoE. With the correct process, connections become relevant and add value because the right information is delivered to the right person at the right time in the appropriate way.

2.1 Architecture

Implementation of IoE environments is usually based on a standard architecture derived from IoT. This architecture consists of several layers [5, 18]: from the data

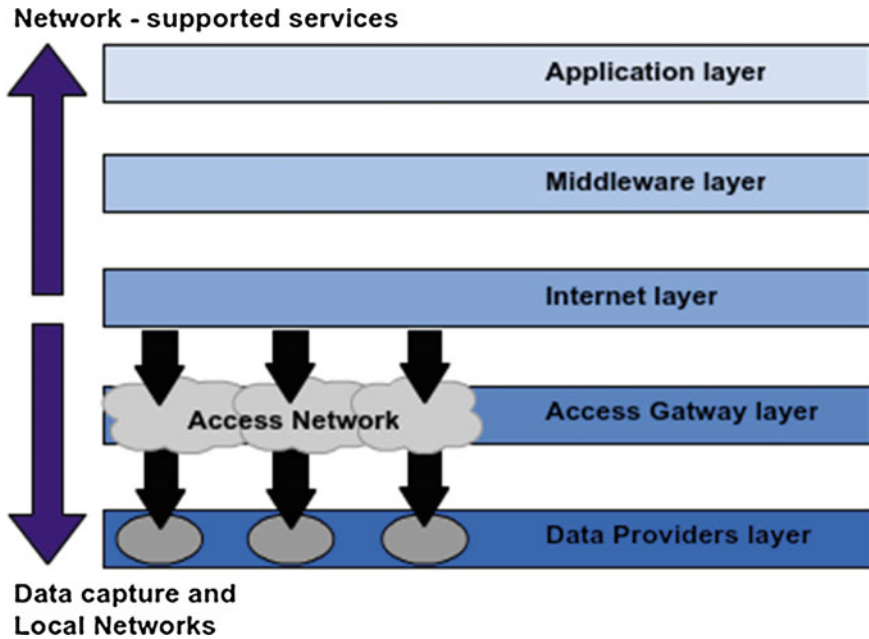


Fig. 3 Layered architecture of Internet of Everything

acquisition layer at the bottom to the application layer at the top. Figure 3 presents the generic architecture for IoE [3].

The two layers at the bottom contribute to data capturing while the two layers at the top are responsible for data utilization in applications. Next, we present the functionality of these layers [5]:

- **Data providers layer:** This layer consists of hardware components such as sensor networks, embedded systems, RFID tags and readers or other IoE devices in different forms. Moreover, in this layer is also present other components, like people information, that is also an IoE entity that provides data to the environment. These entities are the primary data sources deployed in the field. Many of these elements provide identification and information storage (e.g. RFID tags), information collection (e.g. sensor networks), information processing (e.g. embedded edge processors), communication, control and actuation. However, identification and information collection are the primary goals of these entities, leaving the processing activities for the upper layers.
- **Access gateway layer:** The first stage of data handling happens at this layer. It takes care of message routing, publishing and subscribing, and also performs cross platform communication, if required.
- **Middleware layer:** This layer acts as an interface between the hardware layer at the bottom and the application layer at the top. It is responsible for critical functions such as device management and information management, and also

takes care of issues like data filtering, data aggregation, semantic analysis, access control, and information discovery.

- **Application layer:** This layer at the top of the stack is responsible for the delivery of various services to different users/applications in IoE environments. The applications can be from different industry verticals such as: manufacturing, logistics, retail, environment, public safety, healthcare, food and drug, etc.

2.2 *Characteristics and Environments*

IoT allows communication among very heterogeneous devices connected by a very wide range of networks through the Internet infrastructure. IoT devices and resources are any kind of device connected to Internet, from existing devices, such as servers, laptops, and personal computers, to emerging devices such as smart phones, smart meters, sensors, identification readers, and appliances [28].

In addition to the physical devices, IoT is also enriched with the cybernetic resources and Web-based technologies. For that purpose, IoT is enabled with interfaces based on Web Services such as RESTful architecture and the novel protocol for Constrained devices Applications Protocol (CoAP) [43]. These interfaces enable the seamless integration of the IoT resources with information systems, management systems, and the humans. Reaching thereby a universal and ubiquitous integration among human networks (i.e., society), appliance networks, sensor networks, machine networks, and, in definitive, everything networks [28].

Beside these devices, the People and Data (see Fig. 2) can also make part of this connection, thus we have the IoE. IoE offers several advantages and new capabilities for a wide range of application areas. For example, nowadays IoE is finding applications for the development of Smart Cities, starting with the Smart Grid, Smart Lighting and transportation with new services such as Smart Parking and the Bicycle Sharing System [20] for building sustainable and efficiently smart ecosystems [28].

The application of the IoE is not limited to high scale deployments such as the locations in Smart Cities, elsewhere it can also be considered for consumer electronics, vehicular communications, industrial control, building automation, logistic, retail, marketing, and healthcare [28].

3 **Context-Aware Life-Cycle**

Context is considered any information that can be used to characterize the situation of an entity. Entity is a person, place, or computing device (also called thing) that is relevant to the interaction between a user and an application, including the user and the application themselves. A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the

user's task [2, 34]. In this way, an IoE ecosystem requires a context-aware mechanism to be aware of the environment situation in order to help the user in the most useful way. In various cases, the context-aware becomes a feature of IoE systems.

Different researchers have identified context types based of different perspectives. Abowd et al. [2] introduced one of the leading mechanisms of defining context types. They identified location, identity, time, and activity as the primary context types. Further, they defined secondary context as the context that can be found using primary context [39]. For example, given primary context such as a person's identity, we can acquire many pieces of related information such as phone numbers, addresses, email addresses, etc. Some examples defined by [39] are:

- **Primary context:** Any information retrieved without using existing context and without performing any kind of sensor data fusion operations (e.g. GPS sensor readings as location information).
- **Secondary context:** Any information that can be computed using primary context. The secondary context can be computed by using sensor data fusion operations or data retrieval operations such as web service calls (e.g. identify the distance between two sensors by applying sensor data fusion operations on two raw GPS sensor values). Further, retrieved context such as phone numbers, addresses, email addresses, birthdays, list of friends from a contact information provider based on a personal identity as the primary context can also be identified as secondary context.

A set of methods is mandatory in order to obtain the context of an entity. Furthermore, there is a set of actions, organized in phases, that characterizes the context life-cycle of an information. Perera et al. [39] proposed a life-cycle and explained how acquisition, modelling, reasoning, and distribution of context should occur.

3.1 Context Acquisition

In acquisition process, context needs to be acquired from various information sources. These sources can be physical or virtual devices. The techniques used to acquire context can vary based on responsibility, frequency, context source, sensor type, and acquisition process [39].

- (1) *Based on Responsibility:* Context acquisition can be primarily accomplished using two methods [40]: push and pull.
 - **Push:** The physical or virtual sensor pushes data to the data consumer which is responsible to acquiring sensor data periodically or instantly. Periodical or instant pushing can be employed to facilitate a publish and subscribe model.

- Pull: The data consumers make a request from the sensor hardware periodically or instantly to acquire data.
- (2) *Based on Frequency*: There are two different types: Instant and Interval.
- Instant: These events occur instantly. The events do not span across certain amounts of time. In order to detect this type of event, sensor data needs to be acquired when the event occurs. Both push and pull methods can be employed.
 - Interval: These events span in a certain period of time. In order to detect this type of event, sensor data needs to be acquired periodically. Both push and pull methods can be employed.
- (3) *Based on Source*: Context acquisition methods can be organized into three categories [12].
- Acquire directly from sensor hardware: In this method, context is directly acquired from the sensor by communicating with the sensor hardware and related APIs. Software drivers and libraries need to be installed locally.
 - Acquire through a middleware infrastructure: In this method, sensor (context) data is acquired by middleware solutions. The applications can retrieve sensor data from the middleware and not from the sensor hardware directly.
 - Acquire from context servers: In this method, context is acquired from several other context storage types (e.g. databases, web services) by different mechanisms such as web service calls.
- (4) *Based on Sensor Types*: In general usage, the term ‘sensor’ is used to refer the tangible sensor hardware devices. However, among the technical community, sensors are referred as any data source that provides relevant context. Therefore, sensors can be divided into three categories [26]: physical, virtual, and logical.
- Physical sensors: These are the most commonly used type of sensors. These sensors generate data by themselves. Most of the devices we use today are equipped with a variety of physical sensors (e.g. temperature, humidity, microphone, touch).
 - Virtual sensors: These sensors do not necessarily generate data by themselves. Virtual sensors retrieve data from many sources and publish it as sensor data (e.g. calendar, contact number directory, twitter statuses, email, and chat applications). These sensors do not have a physical presence.
 - Logical sensors (also called software sensors): They combine physical sensors and virtual sensors in order to produce more meaningful information. A web service dedicated to providing weather information can be called a logical sensor.
- (5) *Based on Acquisition Process*: Here are three ways to acquire context: sense, derive, and manually provided.

- **Sense:** The data is sensed through sensors, including the sensed data stored in databases (e.g. retrieve temperature from a sensor, retrieve appointments details from a calendar).
- **Derive:** The information is generated by performing computational operations on sensor data. These operations could be as simple as web service calls or as complex as mathematical functions running over sensed data (e.g. calculate distance between two sensors using GPS coordinates).
- **Manually provided:** Users provide context information manually via pre-defined settings options such as preferences (e.g. understanding that user doesn't like to receive event notifications between 10 pm to 6 am).

3.2 Context Modeling

Context modeling is organized in two steps [7]. First, new context information needs to be defined in terms of attributes, characteristics, and relationships with previously specified context. In the second step, the outcome of the first step needs to be validated and the new context information needs to be merged and added to the existing context information repository. Finally, the new context information is made available to be used when needed.

The most popular context modeling techniques are surveyed in [11, 44]. These surveys discuss a number of systems that have been developed based on the following techniques. Each technique has its own strengths and weaknesses.

- (1) *Key-Value Modelling:* In the key-value each data has a key. The key-value technique is an application oriented and application bounded technique that suits the purpose of temporary storage such as less complex application configurations and user preferences. It models context information as key-value pairs in different formats such as text files and binary files. This is the simplest form of context representation among all the other techniques. They are easy to manage when they have smaller amounts of data. However, key-value modelling is not scalable and not suitable to store complex data structures.
- (2) *Markup Scheme Modelling (Tagged Encoding):* It models data using tags. Therefore, context is stored within tags. This technique is an improvement over the key-value modelling technique. The advantage of using markup tags is that it allows efficient data retrieval. Markup schemes such as XML are widely used in almost all application domains to store data temporarily, transfer data among applications, and transfer data among application components. In contrast, markup languages do not provide advanced expressive capabilities which allow reasoning.
- (3) *Graphical Modelling:* It models context with relationships. Some examples of this modelling technique are Unified Modelling Language (UML) [45] and Object Role Modelling (ORM) [36]. Actual low-level representation of the graphical modelling technique could be varied. For example, it could be a SQL

database, noSQL database, etc. Further, as we are familiar with databases, graphical modelling is a well-known, easy to learn, and easy to use technique. Databases can hold massive amounts of data and provide simple data retrieval operations, which can be performed relatively quickly. In contrast, the number of different implementations makes it difficult with regards to interoperability.

- (4) *Object Based Modelling*: Object based (or object oriented) concepts are used to model data using class hierarchies and relationships. Object oriented paradigm promotes encapsulation and re-usability. As most of the high-level programming languages support object oriented concepts, modelling can be integrated into context-aware systems easily. Object based modelling is suitable to be used as an internal, non-shared, code based, run-time context modelling, manipulation, and storage mechanism. Validation of object oriented designs is difficult due to the lack of standards and specifications.
- (5) *Logic Based Modelling*: Facts, expressions, and rules are used to represent information about the context. Rules are primarily used to express policies, constraints, and preferences. It provides much more expressive richness compared to the other models discussed previously. Therefore, reasoning is possible up to a certain level. Logic based modelling allows new high-level context information to be extracted using low-level context.
- (6) *Ontology Based Modelling*: The context is organized into ontologies using semantic technologies. A number of different standards and reasoning capabilities are available to be used depending on the requirement. A wide range of development tools and reasoning engines are also available. However, context retrieval can be computationally intensive and time consuming when the amount of data is increased.

3.3 Context Reasoning

Context reasoning can be defined as a method of deducing new knowledge based on the available context [8]. It can also be explained as a process of giving high-level context deductions from a set of contexts [22]. Reasoning is also called inferencing. Broadly the reasoning can be divided into three phases [35].

- Context pre-processing: This phase cleans the collected sensor data. Due to inefficiencies in sensor hardware and network communication, collected data may be not accurate or missing. Therefore, data needs to be cleaned by filling missing values, removing outliers, validating context via multiple sources, and many more.
- Sensor data fusion: It is a method of combining sensor data from multiple sensors to produce more accurate, more complete, and more dependable information that could not be achieved through a single sensor [24].
- Context inference: It is a method of generation of high-level context information using lower-level context. The inferencing can be done in a single interaction or

in multiple interactions. For example in a situation where the context is represented as tuples (e.g. Who: Leonardo, What: walking: 1 km/h, Where: Porto Alegre, When: 2016-01-05:11.30 am). This low-level context can be inferred through a number of reasoning mechanisms to generate the final results. For example, in the first iteration, longitude and latitude values of a GPS sensor may be inferred as Rei do Cordeiro restaurant in Porto Alegre. In the next iteration Rei do Cordeiro restaurant in Porto Alegre may be inferred as Leonardo's favourite restaurant. Each iteration gives more accurate and meaningful information.

In [39], context reasoning techniques are classified into six categories: supervised learning, unsupervised learning, rules, fuzzy logic, ontological reasoning, and probabilistic reasoning.

- (1) *Supervised learning*: In this category of techniques, we first collect training examples. Then we label them according to the results we expect. Then we derive a function that can generate the expected results using the training data. Decision tree is a supervised learning technique where it builds a tree from a dataset that can be used to classify data.
- (2) *Unsupervised learning*: This category of techniques can find hidden structures in unlabeled data. Due to the use of no training data, there is no error or reward signal to evaluate a potential solution.
- (3) *Rules*: This is the simplest and most straightforward method of reasoning. Rules are usually structured in an IF-THEN-ELSE format. Rules are expected to play a significant role in the IoE, where they are the easiest and simplest way to model human thinking and reasoning in machines.
- (4) *Fuzzy logic*: This allows approximate reasoning instead of fixed and crisp reasoning. Fuzzy logic is similar to probabilistic reasoning but confidence values represent degrees of membership rather than probability [42]. In traditional logic theory, acceptable truth values are 0 or 1. In fuzzy logic partial truth values are acceptable. It allows real world scenarios to be represented more naturally; as most real world facts are not crisp.
- (5) *Ontology based*: It is based on description logic, which is a family of logic based knowledge representations of formalisms. The advantage of ontological reasoning is that it integrates well with ontology modelling. In contrast, a disadvantage is that ontological reasoning is not capable of finding missing values or ambiguous information where statistical reasoning techniques are good at that. Rules can be used to minimize this weakness by generating new context information based on low-level context.
- (6) *Probabilistic logic*: This category allows decisions to be made based on probabilities attached to the facts related to the problem. This technique is used to understand occurrence of events. For example, it provides a method to bridge the gap between raw GPS sensor measurements and high level information such as a user destination, mode of transportation, calendar based observable evidence such as user calendar, weather, etc.

3.4 *Context Distribution*

Finally, context distribution is a fairly straightforward task. It provides methods to deliver context to the consumers. From the consumer perspective this task can be called context acquisition. There are two methods that are commonly used in context distribution [39]:

- **Query:** Context consumer makes a request in terms of a query, so the context management system can use that query to produce results.
- **Subscription (also called publish/subscribe):** Context consumer can be allowed to subscribe to a context management system by describing the requirements. The system will then return the results periodically or when an event occurs. In other terms, consumers can subscribe for a specific sensor or to an event.

4 *Context-Aware Systems*

Context-awareness involves acquisition of contextual information, modelling of these information, reasoning about context, and distribution of context. A system for context-awareness would provide support for each of these tasks. It would also define a common model of context, which all agents can use in dealing with context. Moreover, it would ensure that different agents in the environment have a common semantic understanding of contextual information.

4.1 *Architecture Overview*

In terms of architecture, some authors have identified and comprehensively discussed some design principles related to context-aware systems [39]. We summarize the findings below with brief explanations. This list is not intended to be exhaustive. Only the most important design aspects are considered.

- **Architecture layers and components:** The functionalities need to be divided into layers and components in a meaningful manner. Each component should perform a very limited amount of the task and should be able to perform independently up to a large extent.
- **Scalability and extensibility:** The component should be able to be added or removed dynamically. For example, new functionalities (i.e. components) should be able to be added without altering the existing components (e.g. Open Services Gateway initiative). The component needs to be developed according to standards across the solutions, which improves scalability and extensibility (e.g. plug-in architectures).

- **Application Programming Interface (API):** All the functionalities should be available to be accessed via a comprehensive easy to learn and easy to use API. This allows the incorporation of different solutions very easily. Further, API can be used to bind context management frameworks to applications. Interoperability among different IoE solutions heavily depends on API and their usability.
- **Mobility support:** In the IoE, most devices would be mobile, where each one has a different set of hardware and software capabilities. Therefore, context-aware frameworks should be developed in multiple versions, which can run on different hardware and software configurations (e.g. more capabilities for server level software and less capabilities for mobile phones).
- **Monitoring and detect event:** Events play a significant role in the IoE, which is complemented by monitoring. Detecting an event triggers an action autonomously in the IoE paradigm. This is how the IoE will help humans carry out their day-to-day work easily and efficiently. Detecting events in real-time is a major challenge for context-aware frameworks in the IoE paradigm.

4.2 Systems Features

The context-aware systems must have several features to deal with the context information production. First we will introduce some of these features, and in the Sect. 5 a comparison table of systems regarding these features will be shown. The most important features are surveyed by Perera et al. [39] and explained in the follow items:

- (1) **Modelling:** It has been discussed in detail in Sect. 3.2. The following abbreviations are used to denote the context modeling techniques employed by the system: key-value modelling (K), markup Schemes (M), graphical modelling (G), object oriented modelling (Ob), logic-based modelling (L), and ontology-based modelling (On).
- (2) **Reasoning:** It has been discussed in detail in Sect. 3.3. The following abbreviations are used to denote the reasoning techniques employed by the system: supervised learning (S), unsupervised learning (U), rules (R), fuzzy logic (F), ontology-based (O), and probabilistic reasoning (P).
- (3) **Distribution:** It has been discussed in detail in Sect. 3.4. The following abbreviations are used to denote the distribution techniques employed by the system: publish/subscribe (P) and query (Q).
- (4) **History and Storage:** Storing context history is critical in both traditional context-aware computing and IoE [16]. Historic data allows sensor data to be better understood. Specifically, it allows user behaviors, preferences,

patterns, trends, needs, and many more to be understood. The symbol (✓) is used to denote that context history functionality is facilitated and employed by the system.

- (5) **Knowledge Management:** Most of the tasks that are performed by IoE systems solutions require knowledge in different perspectives, such as knowledge on sensors, domains, users, activities, and many more. Knowledge can be used for tasks such as automated configuration of sensors to IoE system, automatic sensor data annotation, reasoning, and event detection. The symbol (✓) is used to denote that knowledge management functionality is facilitated and employed by the system in some perspective.
- (6) **Event Detection:** IoE envisions many types of communication. Most of these interactions are likely to occur based on an event. An occurrence of event is also called an event trigger. Once an event has been triggered, a notification or action may be executed. For example, detecting current activity of a person or detecting a meeting status in a room, can be considered as events. Mostly, event detection needs to be done in real-time. However, events such as trends may be detected using historic data. The symbol (✓) is used to denote that event detection functionality is facilitated and employed by the system in some perspective.
- (7) **Level of Context Awareness:** Context-awareness can be employed at two levels: low (hardware) level and high (software) level. At the hardware level, context-awareness is used to facilitate tasks such as efficient routing, modelling, reasoning, storage, and event detection [25]. The software level has access to a broader range of data and knowledge as well as more resources, which enables more complex reasoning to be performed. The following abbreviations are used to denote the level of context awareness facilitated and employed by the system: high level (H) and low level (L).
- (8) **Data Source Support:** There are different sources that are capable of providing context. The (P) denotes that the solution supports only physical sensors. Software sensors (S) denotes that the solution supports either virtual sensors, logical sensors or both. The (A) denotes that the solution supports all kinds of data sources (i.e. physical, virtual, and logical). The (M) denotes that the solution supports mobile devices.
- (9) **Quality of Context:** It denotes the presence of conflict resolution functionality using (C) and context validation functionality using (V). Conflict resolution is critical in the context management domain [19]. Context validation ensures that collected data is correct and meaningful. Possible validations are checks for range, limit, logic, data type, cross-system consistency, uniqueness, cardinality, consistency, data source quality, security, and privacy.
- (10) **Data Processing:** Are denoted the presence of context aggregation functionality using (A) and context filter functionality using (F). Context filter functionality makes sure that the reasoning engine processes only important data. Filtering functionality can be presented in different solutions with in

different forms: filter data, filter context sources, or filter events. Aggregation can just collect similar information together. This is one of the simplest forms of aggregation of context.

- (11) **Dynamic Composition:** IoE solutions must have a programming model that allows dynamic composition without requiring the developer or user to identify specific sensors and devices. Software solutions should be able to understand the requirements and demands on each situation, then organize and structure its internal components according to them. The symbol (✓) denotes the presence of dynamic composition functionality at the system in some form.
- (12) **Real-Time Processing:** Most of the interactions are expected to be processed in real-time in IoE. This functionality has been rarely addressed by the research community in the context-aware computing domain. The symbol (✓) denotes the presence of real-time processing functionality in some form.
- (13) **Registry Maintenance and Lookup Services:** The (✓) symbol is used to denote the presence of registry maintenance and lookup services functionality in the system. This functionality allows different components such as context sources, data fusion operators, knowledge bases, and context consumers to be registered.

5 Related Work

Some systems provide context-aware functions to IoT and IoE environments. This Section presents some examples of these systems and a brief review about their context-aware features based on systems features presented at Sect. 4.2.

Tables 1 and 2 presents a comparison between systems with context-aware features. The items (features) used for the comparison are: (1) Modelling, (2) Reasoning, (3) Distribution, (4) History and Storage, (5) Knowledge Management, (6) Event Detection, (7) Level of Context Awareness, (8) Data Source Support, (9) Quality of Context, (10) Data Processing, (11) Dynamic Composition,

Table 1 Context life-cycle phases implemented in IoE systems

IoE system	(1)	(2)	(3)
Hydra	K, On, Ob	R, O	Q
COSMOS	Ob	R	Q
SALES	M	R	Q
C-Cast	M	R	P, Q
CoMiHoc	Ob	R, P	Q
MidSen	K	R	P, Q
CARISMA	M	R	Q
ezContext	K, Ob	R	Q
Feel@Home	G, On	O	P, Q

Table 2 Context features implemented in IoE systems

IoE system	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
Hydra	✓	✓	✓	H	P	V	–	–	–	–
COSMOS	–	–	✓	H	P	–	A	✓	–	✓
SALES	–	–	✓	L	P	–	F	–	–	✓
C-Cast	✓	–	✓	H	A	–	–	–	–	✓
CoMiHoc	–	✓	✓	H	A	V	–	–	–	–
MidSen	–	✓	✓	H	P	–	–	–	–	✓
CARISMA	–	–	–	H	M	C	–	–	–	–
ezContext	✓	✓	✓	H	A	–	A	–	–	✓
Feel@Home	–	✓	✓	H	A	–	–	–	–	✓

(12) Real-Time Processing, and (13) Registry Maintenance and Lookup Services. We only analyzed systems that provide details of these features in the literature. The definition of each item is given at the Sect. 4.2.

Hydra [4] is a system that comprises a Context-Aware Framework that is responsible for connecting and retrieving data from sensors, context management and context interpretation. A rule engine called Drools [29] has been employed as the core context reasoning mechanism. COSMOS [14] is a system that enables the processing of context information in ubiquitous environments. COSMOS consists of three layers: context collector (collects information), context processing (derives high level information), and context adaptation (provides context access to applications). Therefore, COSMOS follows distributed architecture which increases the scalability of the system.

SALES [15] is a context-aware system that achieves scalability in context dissemination. The XML schemes are used to store and transfer context. C-Cast [41] is a system that integrates WSN into context-aware systems by addressing context acquisition, dissemination, representation, recognizing, and reasoning about context and situations. The data history can be used for context prediction based on expired context information.

CoMiHoc [46] is a framework that supports context management and situation reasoning. CoMiHoc architecture comprises six components: context provisioner, request manager, situation reasoner, location reasoner, communication manager, and On-Demand Multicast Routing Protocol (ODMRP). MidSen [37], as C-Cast, is a context-aware system for WSN. MidSen is based on Event-Condition-Action (ECA) rules. The system proposes a complete architecture to enable context awareness in WSN.

CARISMA [9] is focused on mobile systems where they are extremely dynamic. Adaptation is the main focus of CARISMA. Context is stored as application profiles (XML based), which allows each application to maintain meta-data. The framework ezContext [33] provides automatic context life cycle management. The ezContext

comprises several components that provides context, retrieves context, modelling and storage context. Feel@Home [23] is a context management framework that supports interaction between different domains. Feel@Home decides what the relevant domain needs to be contacted to answer the user query. Then, the framework redirects the user query to the relevant domain context managers. Feel@Home consists of context management components responsible for context reasoning and store context.

The first feature to be analyzed in Table 1 (first column) is related to systems context modeling feature. The most popular modeling approaches in the comparison were markup schemes, key-value, and object-oriented modeling. Modeling through key-value is made by Hydra for simplicity of use [4]. CARISMA uses markup schemes because the way it models the context can be easily understood, both by machines and by human [9].

In reasoning and distribution (2 and 3 respectively) almost all analyzed systems seem to have a consensus regarding which technologies to use. With respect to reasoning, the most of analyzed systems use rules as a tool. A study by [39] showed that rules is the most popular method of reasoning used by systems. Hydra besides rules also uses ontologies as a promising technology [48]. On the other hand, Feel@Home makes use only of ontologies. To supply context distribution all analyzed systems use query. However, some systems as C-Cast, MidSen, and Feel@Home also offer the possibility of using publish/subscribe as a plus.

In Table 2 the function of history and storage (4) is a differential of the analyzed systems. Only three have this feature. For C-Cast, the history can be used for context prediction based on expired context information [41]. Another differential feature is knowledge management (5). One of the few that provides this functionality is CoMiHoc. In this system, the knowledge is required to overcome the limitations of the environment and to provide reliable support for the applications [46]. Detection of events (6) is a feature provided by almost all systems. When specific context events occur, event detection takes action such as shutting down if the battery is low [4].

In terms of level of context awareness (7), only one system has a low level, which works with the context in hardware. All other analyzed systems work with context in terms of software, which allows a greater capacity for reasoning [39]. Regarding data source support (8), most of analyzed systems support physical sensors. CARISMA supports mobile sensors because it is a specific solution for this area [9]. A better alternative is to support the largest possible range of different sensors, since IoE provides a heterogeneous environment [3].

A comparison was made between systems on quality of context (9). Only three of the analyzed systems control quality of context, and two of them control through validation. In CoMiHoC validation is integrated into the communication protocol [46]. Data processing (10) is another analyzed functionality. Only three systems perform some kind of processing. SALES uses filtering techniques to reduce traffic [15].

Another feature compared between systems was dynamic composition (11). This is only attended by COSMOS [14]. The real-time processing (12) becomes a challenge of future context-aware systems, as none of the analyzed systems had this feature. Finally, the last item used for systems analysis was registry maintenance and lookup services (13). Many of the compared systems have this feature. Through it, the systems can have a history of performed processes, thus making easy future operations [39].

6 Context-Awareness in IoE

Data alone are not very interesting or useful. It is when data can be used and become actionable that it can change processes and have direct positive impact on people's lives. The IoE generates data, and adding analysis turns those data into information. Aggregated data become information that, when analyzed, become knowledge. Knowledge can lead to context and informed decision-making, which at the highest level is wisdom (Fig. 4) [38].

Data for critical decision-making though the IoE can create approximately US \$14.4 trillion dollars of added value in the US commercial sector over the next 10 years across a wide range of industries [38]. This opportunity exists in the form of new value created by technology innovation, market share gains, and increasing competitive advantage. Similarly researches indicates that data analytics were responsible for an improvement in business performance of companies. Capturing these gains, however, may require concurrent investment in resources to manage the rise in data [18].

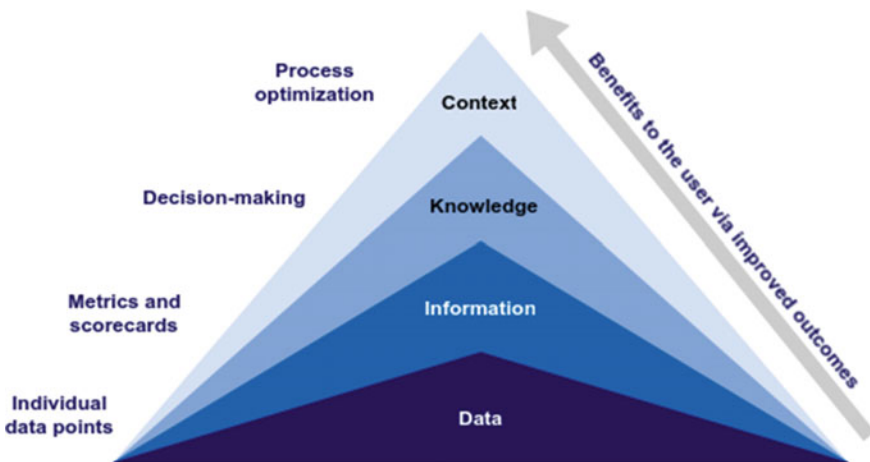


Fig. 4 Turning data into context

6.1 *Technologies and Challenges*

In this section our objective is to discuss eight unique challenges in the IoE where novel techniques and solution may need to be employed [38, 39].

- (1) Automated configuration of data providers: In traditional pervasive/ubiquitous computing, we connect only a limited number of data providers to the applications (e.g. smart farm, smart home) [6]. In contrast, the IoE envisions billions of data providers to be connected together over the Internet. As a result, a unique challenge would arise on connection and configuration of data providers to applications. There has to be an automated or at least semi-automated process to connect data providers to applications.
- (2) Context discovery: Once we connect data providers to a software solution, there has to be a method to understand the data produced by the data providers and the related context automatically. There are many types of context that can be used to enrich data. However, understanding sensor data and appropriately annotating it automatically in a paradigm such as IoE, where application domains vary widely, is a challenging task.
- (3) Acquisition, modelling, reasoning, and distribution: No single technique would serve the requirements of the IoE. Incorporating and integrating multiple techniques has shown promising success in the field. Some of the early work, such as [7, 10], have discussed the process in detail. However, due to the immaturity of the field of IoE, it is difficult to predict when and where to employ each technique. Therefore, it is important to define and follow a standard specification so different techniques can be added to the solutions without significant effort.
- (4) Selection of data providers: It is clear that we are going to have access to billions of data providers. In such an environment, there could be many different alternative data providers to be used. For example, in some cases, there will be many similar data providers in a complex environment like a smart city.
- (5) Security, privacy, and trust: The advantage of context is that it provides more meaningful information that will help us to understand a situation or data. At the same time, it increases the security threats due to possible misuse of the context (e.g. identity, location, activity, and behavior). In the IoE, security and privacy need to be protected in several layers: sensor hardware layer, sensor data communication (protocol) layer, context annotation and context discovery layer, context modelling layer, and the context distribution layer. IoE is a community based approach where the acceptance of the users (e.g. general public) is essential. Therefore, security and privacy protection requirements need to be carefully addressed in order to win the trust of the users.
- (6) Scalability: The growth of mobile data traffic will require greater radio spectrum to enable wireless M2M, as well as people-to-people (P2P) and people-to-machine (P2M), connectivity. Ensuring device connectivity and sufficient bandwidth for all of the uses of wireless sensors will require careful

planning. Moreover, the context to be processed will grow, and the context systems will need to adapt to this scenario keeping the reliability.

- (7) **Reliability:** As more critical processes are conducted as part of the IoE, the need for reliability increases. Healthcare applications that require instant communication between end-users and medical professionals, safety and security applications, utility functions, and industrial uses are some examples where continuous, uninterrupted, real-time communications require reliable and redundant connectivity. The context systems will be present in these fields, and they must work correctly in these critical scenarios.
- (8) **Context Sharing and Interoperability:** This is largely neglected in the context-aware systems domain. Most of the systems solutions or architectures are designed to facilitate applications in isolated factions. Inter-systems communication is not considered to be a critical requirement. However, in the IoE, there would be no central point of control. Different systems developed by different parties will be employed to connect to sensors, collect, model, and reason context. Therefore, sharing context information between different kinds of systems or different instances of the same systems is important. Standards and interoperability issues span both the technical and architectural domains. In this sense, an interoperability between systems will be required.

7 Summary

The use of mobile communication networks has increased significantly in the past decades. The proliferation of smart devices (e.g. data providers) and the resulting exponential growth in data traffic has increased the need for higher capacity wireless networks. In addition, new paradigms are emerging, like Internet of Things (IoT) and Internet of Everything (IoE). With these paradigms, billions of data providers will be connected to the Internet in next years. The attention is now shifting toward the next set of innovations in architecture, technologies, and systems that will address the capacity and service demands envisioned for this evolutionary wave. These innovations are expected to form the so called fifth generation of communications systems.

Can be identified through literature that there are significant amount of systems for data management related to IoE, sensor networks, and pervasive/ubiquitous computing. However, unless the system can analyze, interpret, and understand these data, it will keep useless and without meaning for the users and applications. The context is used to give meaning to these data. A context-aware feature is required to the systems in order to address this challenge.

As can be seen during this chapter, there are some systems with different architectures that have context-aware features, thus enabling a sensing-as-a-service platform. The system features can vary in different ways, in addition to the modules

that compose it. On the other hand, all the systems may follow the four phases of context life-cycle (acquisition, modelling, reasoning, and distribution) in order to produce context information.

The new requirements imposed by IoE will drive to new context-aware challenges. The systems will aim to produce context in the most efficient way. Moreover, there are many challenges involving the process as well, like: automated configuration of data providers, context discovery, context life-cycle phases, selection of data providers, security issues, scalability, reliability, context sharing and interoperability. These challenges will force new directions to the context-aware systems of the future IoE environments.

Acknowledgments Our thanks to CAPES/CNPq for the funding within the scope of the project number 384843/2015-8.

References

1. Abdelwahab, S., Hamdaoui, B., Guizani, M., Rayes, A.: Enabling smart cloud services through remote sensing: An internet of everything enabler. *Internet of Things Journal*, IEEE 1 (3), 276–288 (2014)
2. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggle, P.: Towards a better understanding of context and context-awareness. In: *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, pp. 304–307. Springer (1999). URL <http://dl.acm.org/citation.cfm?id=647985.743843>
3. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. *Computer Networks* 54(15), 2787–2805 (2010). doi:<http://dx.doi.org/10.1016/j.comnet.2010.05.010>
4. Badii, A., Crouch, M., Lallah, C.: A context-awareness framework for intelligent networked embedded systems. In: *Advances in Human-Oriented and Personalized Mechanisms, Technologies and Services (CENTRIC), 2010 Third International Conference on*, pp. 105–110. IEEE (2010)
5. Bandyopadhyay, D., Sen, J.: Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications* 58(1), 49–69 (2011). doi:[10.1007/s11277-011-0288-5](https://doi.org/10.1007/s11277-011-0288-5). URL <http://dx.doi.org/10.1007/s11277-011-0288-5>
6. Batalla, J.M., Mastorakis, G., Mavroumoustakis, C.X., Z'urek, J.: On cohabitating networking technologies with common wireless access for home automation systems purposes. In: *IEEE Wireless Communications Magazine (To be published)*. IEEE (2016)
7. Bettini, C., Brdiczka, O., Henriksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing* 6(2), 161–180 (2010)
8. Bikakis, A., Patkos, T., Antoniou, G., Plexousakis, D.: A survey of semantics-based approaches for context reasoning in ambient intelligence. In: *Constructing ambient intelligence*, pp. 14–23. Springer (2008)
9. Capra, L., Emmerich, W., Mascolo, C.: Carisma: Context-aware reflective middleware system for mobile applications. *Software Engineering*, IEEE Transactions on 29(10), 929–945 (2003)
10. Chellouche, S.A., Négru, D., Arnaud, J., Batalla, J.M.: Context-aware multimedia services provisioning in future internet using ontology and rules. In: *Network of the Future (NOF), 2014 International Conference and Workshop on the*, pp. 1–5 (2014). doi:[10.1109/NOF.2014.7119778](https://doi.org/10.1109/NOF.2014.7119778)

11. Chen, G., Kotz, D., et al.: A survey of context-aware mobile computing research. Tech. rep., Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College (2000)
12. Chen, H., Finin, T., Joshi, A., Kagal, L., Perich, F., Chakraborty, D.: Intelligent agents meet the semantic web in smart spaces. *Internet Computing*, IEEE 8(6), 69–79 (2004). doi:[10.1109/MIC.2004.66](https://doi.org/10.1109/MIC.2004.66)
13. Clarke, R.Y.: Smart cities and the internet of everything: The foundation for delivering nextgeneration citizen services. Alexandria, VA, Tech. Rep (2013)
14. Conan, D., Rouvroy, R., Seinturier, L.: Scalable processing of context information with cosmos. In: *Distributed Applications and Interoperable Systems*, pp. 210–224. Springer (2007)
15. Corradi, A., Fanelli, M., Foschini, L.: Implementing a scalable context-aware middleware. In: *Computers and Communications*, 2009. ISCC 2009. IEEE Symposium on, pp. 868–874. IEEE (2009)
16. Dey, A.K., Abowd, G.D., Salber, D.: A context-based infrastructure for smart environments. In: *Managing Interactions in Smart Environments*, pp. 114–128. Springer (2000)
17. Dohr, A., Modre-Opsrian, R., Drobits, M., Hayn, D., Schreier, G.: The internet of things for ambient assisted living. In: *Information Technology: New Generations (ITNG)*, 2010 Seventh International Conference on, pp. 804–809. Ieee (2010)
18. Evans, D.: The internet of things: How the next evolution of the internet is changing everything. CISCO white paper 1, 14 (2011)
19. Filho, J., Agoulmine, N.: A quality-aware approach for resolving context conflicts in contextaware systems. In: *Embedded and Ubiquitous Computing (EUC)*, 2011 IFIP 9th International Conference on, pp. 229–236 (2011). doi:[10.1109/EUC.2011.9](https://doi.org/10.1109/EUC.2011.9)
20. Froehlich, J., Neumann, J., Oliver, N.: Measuring the pulse of the city through shared bicycle programs. *Proc. of UrbanSense08* pp. 16–20 (2008)
21. Giusto, D., Iera, A., Morabito, G.: *The Internet of Things*. Springer (2010)
22. Guan, D., Yuan, W., Lee, S., Lee, Y.K.: Context selection and reasoning in ubiquitous computing. In: *Intelligent Pervasive Computing*, 2007. IPC. The 2007 International Conference on, pp. 184–187 (2007). doi:[10.1109/IPC.2007.102](https://doi.org/10.1109/IPC.2007.102)
23. Guo, B., Sun, L., Zhang, D.: The architecture design of a cross-domain context management system. In: *Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2010 8th IEEE International Conference on, pp. 499–504. IEEE (2010)
24. Hall, D., Llinas, J.: An introduction to multisensor data fusion. *Proceedings of the IEEE* 85(1), 6–23 (1997). doi:[10.1109/5.554205](https://doi.org/10.1109/5.554205)
25. Huaifeng, Q., Xingshe, Z.: Context aware sensornet. In: *Proceedings of the 3rd International Workshop on Middleware for Pervasive and Ad-hoc Computing*, MPAC’05, pp. 1–7. ACM, New York, NY, USA (2005). doi:[10.1145/1101480.1101489](https://doi.org/10.1145/1101480.1101489). URL <http://doi.acm.org/10.1145/1101480.1101489>
26. Indulska, J., Sutton, P.: Location management in pervasive systems. In: *Proceedings of the Australasian Information Security Workshop Conference on ACSW Frontiers 2003—Volume 21, ACSW Frontiers’03*, pp. 143–151. Australian Computer Society, Inc., Darlinghurst, Australia, Australia (2003). URL <http://dl.acm.org/citation.cfm?id=827987.828003>
27. Institutes, C.: Smart networked objects and internet of things. Carnot Institutes’ Information Communication Technologies and Micro Nano Technologies alliance, White Paper (2011)
28. Jara, A.J., Ladid, L., Skarmeta, A.: The internet of everything through ipv6: An analysis of challenges, solutions and opportunities. *J. Wirel. Mob. Netw. Ubiqu. Comput. Dependable Appl* 4, 97–118 (2013)
29. jboss.org: Drools—the business logic integration platformn. <http://www.jboss.org/drools> (2001). Accessed: 2015-05-15
30. Kortuem, G., Kawsar, F., Fitton, D., Sundramoorthy, V.: Smart objects as building blocks for the internet of things. *Internet Computing*, IEEE 14(1), 44–51 (2010)

31. Kryftis, Y., Mavromoustakis, C.X., Mastorakis, G., Pallis, E., Batalla, J.M., Rodrigues, J.J.P. C., Dobre, C., Kormentzas, G.: Resource usage prediction algorithms for optimal selection of multimedia content delivery methods. In: 2015 IEEE International Conference on Communications (ICC), pp. 5903–5909 (2015). doi:[10.1109/ICC.2015.7249263](https://doi.org/10.1109/ICC.2015.7249263)
32. Mahoney, J., LeHong, H.: Innovation insight: the ‘internet of everything’ innovation will transform business. Gartner Online, January 3 (2012)
33. Martín, D., Lamsfus, C., Alzua, A.: Automatic context data life cycle management framework. In: Pervasive Computing and Applications (ICPCA), 2010 5th International Conference on, pp. 330–335. IEEE (2010)
34. Matos, E., Amaral, L., Tiburski, R., Lunardi, W., Hessel, F., Marczak, S.: Context-aware system for information services provision in the internet of things. In: Emerging Technologies & Factory Automation, 2015. ETFA 2015. IEEE Conference on, pp. 1–4. IEEE (2015)
35. Nurmi, P., Floree, P.: Reasoning in context-aware systems. position paper. In: Department of Computer Science, University of Helsinki (2004)
36. Ormfoundation.org: The orm foundation (1989). URL <http://www.ormfoundation.org>
37. Patel, P., Jardosh, S., Chaudhary, S., Ranjan, P.: Context aware middleware architecture for wireless sensor network. In: Services Computing, 2009. SCC’09. IEEE International Conference on, pp. 532–535. IEEE (2009)
38. Pepper, R., Garrity, J.: The internet of everything: How the network unleashes the benefits of big data. The global information technology report pp. 35–42 (2014)
39. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Context aware computing for the internet of things: A survey. Communications Surveys Tutorials, IEEE 16(1), 414–454 (2014). doi:[10.1109/SURV.2013.042313.00197](https://doi.org/10.1109/SURV.2013.042313.00197)
40. Pietschmann, S., Mitschick, A., Winkler, R., Meissner, K.: Croco: Ontology-based, crossapplication context management. In: Semantic Media Adaptation and Personalization, 2008. SMAP’08. Third International Workshop on, pp. 88–93 (2008). doi:[10.1109/SMAP.2008.10](https://doi.org/10.1109/SMAP.2008.10)
41. Reetz, E.S., Tonjes, R., Baker, N.: Towards global smart spaces: Merge wireless sensor networks into context-aware systems. In: Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on, pp. 337–342. IEEE (2010)
42. Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R.H., Nahrstedt, K.: A middleware infrastructure for active spaces. IEEE pervasive computing 1(4), 74–83 (2002)
43. Shelby, Z., Hartke, K., Bormann, C.: The constrained application protocol (coap) (2014)
44. Strang, T., Linnhoff-Popien, C.: A context modeling survey. In: Workshop Proceedings (2004)
45. Uml.org: Unified modeling language (uml) (2012). URL <http://www.uml.org/>
46. Wibisono, W., Zaslavsky, A., Ling, S.: Comihoc: A middleware framework for context management in manet environment. In: Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on, pp. 620–627. IEEE (2010)
47. Zaslavsky, A., Perera, C., Georgakopoulos, D.: Sensing as a service and big data. arXiv preprint [arXiv:1301.0159](https://arxiv.org/abs/1301.0159) (2013)
48. Zhang, W., Hansen, K.M.: Towards self-managed pervasive middleware using owl/swrl ontologies. In: Fifth International Workshop on Modelling and Reasoning in Context. MRC 2008 (2008)
49. Zieliński, J.S.: Internet of Everything (IoE) in smart grid. Przegląd Elektrotechniczny 91(3), 157–159 (2015)

Enabling User Context Utilization in the Internet Communication Protocols: Motivation, Architecture and Examples

Yu Lu

Abstract The communication protocols in the Internet protocol stack never explicitly take into account the context information of its dynamic end-users, which affects protocol performance from the perspectives of both end-users and networks. The fast progress in context-aware computing combined with the sensing technologies greatly facilitates collecting and understanding the context information of Internet end-users. Proper utilization of the substantive and abstract end-user's context information provides major opportunities to strengthen today's Internet to be a context-aware, intelligent and user-centric communication system. We therefore propose a new functional module, named User-Context Module, to explicitly integrate the end-user's context information into the established Internet protocol stack. In this chapter, we present this work in three phases: (i) the module's architectural design; (ii) the module's applications; (iii) a resource management framework designed for the module.

1 Introduction

1.1 Motivation

The Internet has achieved tremendous success due to many fundamental and respected design principles for building its protocol stack, such as the layered architecture for task partitioning and end-to-end arguments for implementing protocol functionalities. One of its fundamental design principles is that the Internet serves as the communication medium between two networked hosts that desire to speak to each other [1], where networked hosts work as the delegated representative of Inter-

Y. Lu (✉)
Institute for Infocomm Research (I2R), A*STAR, Singapore 138632, Singapore
e-mail: victoryluyu@gmail.com

Y. Lu
Beijing Advanced Innovation Center For Future Education,
Beijing Normal University, Beijing, China

© Springer International Publishing AG 2017
J.M. Batalla et al. (eds.), *Beyond the Internet of Things*,
Internet of Things, DOI 10.1007/978-3-319-50758-3_2

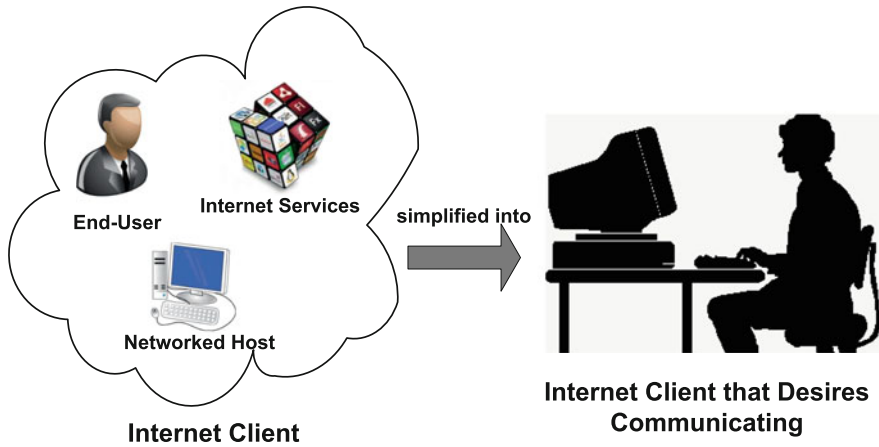


Fig. 1 Oversimplification of an Internet client

net end-users [2]. Such a traditional design principle directly results in today's Internet protocols simply regarding its end-user, host and service as one entity, namely the *Internet client*. More specifically, the Internet protocol stack conflates its dynamic end-user, networked host and running service into one oversimplified concept: *an Internet client that desires communication*. Figure 1 simply depicts such a design principle used by the Internet protocol stack and its communication protocols. Note that the end-user refers to the person who uses Internet services through a networked host. Internet services span a wide range of online services, typically including World Wide Web, file transfer as well as streaming media service.

There is no doubt that such a traditional design principle greatly decreases today's Internet complexity, but it essentially and completely excludes the end-user factor from the Internet client entity and entire Internet protocol stack. Consequently, Internet communication protocols inevitably neglect end-users' presence, preference and interactions with Internet services and hosts. As a result, the Internet protocol stack is unable to take advantage of its end-users' information, especially the context information that can be utilized in the Internet communication protocols. The absence of the end-users' context information may not only affect the underlying network performance but also decrease effectiveness of Internet services. In short, the Internet protocol stack does not explicitly take into account dynamic end-users and their context information in its architectural design, which affects its performance from the perspectives of both end-users and networks.

On the other hand, advances in context-aware computing, combined with the latest sensor technology and the cognitive psychology, greatly facilitate collecting and ascertaining context information of Internet end-users. Proper utilization of the highly abstract and substantive end-user's context information presents major opportunities to strengthen the Internet to be context-aware and user-centric. The term *context* refers to "any information that can be used to characterize the situation of an

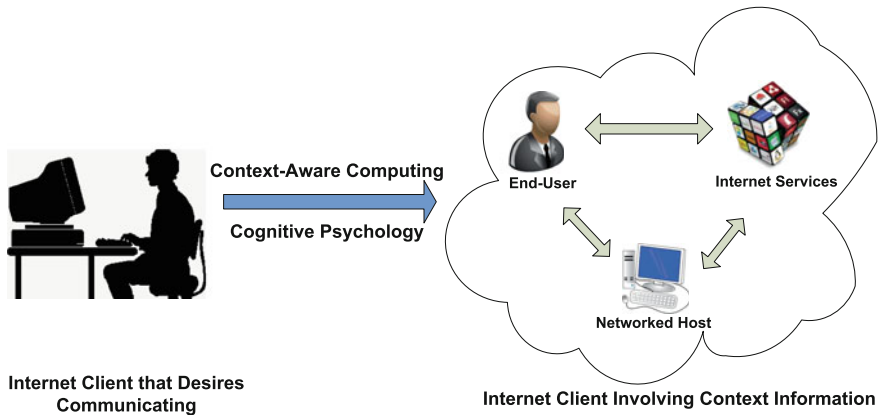


Fig. 2 De-conflation of an Internet client

entity that is considered relevant to the interaction between an end-user and the application, including the end-user and the application themselves” [3]. Briefly speaking, a context-aware system makes use of various sensors and techniques to collect a system’s physical and environmental information. The system then can adapt its operations to the collected context information to increase its usability and effectiveness. There has been an entire body of research dedicated to building context-aware systems, where the Internet always serves as a communication carrier to undertake the task of long distance data transmission. However, few prior studies consider introducing the captured end-users’ context information into the underlying communication protocols of the Internet. We target on incorporating the end-users’ context information into the Internet communication protocols in an explicit way and eventually enable the Internet to adapt its operations to its dynamic end-users. As shown in Fig. 2, the developed context-aware computing and other techniques help to extract the end-users’ context information and restore the oversimplified Internet client.

1.2 Research Challenges

Introducing end-users’ context information into the Internet communication protocols is different from building other context-aware systems, and the difficulties stem mainly from the following open issues:

1. What types of context information can be utilized by the Internet communication protocols?
2. How should the Internet communication protocols properly utilize and adapt themselves to the derived context information?
3. How to guide and incentivize the context sharing among Internet clients?

Firstly, only the highly abstract and substantive context information can be introduced into the Internet communication protocols, which should accurately reflect the dynamic changes of an end-user's real-time interaction states with the Internet. Any irrelevant or redundant context information should be excluded from the Internet protocol stack, as the Internet's key responsibility is to provide the end-to-end connectivity service. The context information should be acquired and verified from multiple and heterogeneous sources.

Secondly, the layered architecture of the Internet provides natural abstractions to deal with the functional hierarchy present in the Internet protocol stack, and thus the communication protocols running at a particular layer do not need to worry about the rest of the stack. The selected context information should be cautiously introduced into the communication protocols to avoid spoiling the integrity and modularity of the Internet architecture. Improperly introducing the context information would impair the basic functions and operations of the relevant protocols, and even lead to unintended consequences on overall performance of the entire layer.

Thirdly, when the context information available at the Internet client side, a new resource distribution mechanism is required to utilize the context information, and meanwhile incentivize the Internet clients providing their actual context information.

1.3 Contributions

To address the above mentioned research issues and challenges, we propose a functional module, called User-Context Module [4, 5], and on this chapter, we will study and exploit it in the following three aspects:

1. The basic architectural design of the User-Context Module.
2. The applications of the User-Context Module.
3. The resource distribution framework designed for the User-Context Module.

Firstly, we introduce the basic architecture of the User-Context Module, which consists of the three subsystems and can identify several fundamental categories of the end-user context information.

Secondly, we present two applications of the User-Context Module to demonstrate its operation, implementation and performance. The network experimental results show that the applications can effectively enhance the end-user's quality of experience (QoE) and meanwhile improve the underlying protocol performance.

Thirdly, we design a novel resource distribution framework that provides the context-based service differentiation and encourages clients to share their actual context information.

2 User-Context Module Design

2.1 System Architecture

With the objective of introducing the context information into the Internet communication protocols, the User-Context Module mainly operates on top of the five-layer Internet protocol stack and under the client-server architecture. As illustrated in Fig. 3, the User-Context Module consists of three indispensable subsystems: context sensing subsystem, context model subsystem and context control subsystem.

- Context sensing subsystem** mainly works at the Internet client side, and its main functionalities involve collecting and storing the basic context information. The basic context information includes the user-oriented information, e.g., whether an end-user’s eye-gaze direction currently towards the networked host screen, and the host-oriented information, e.g., which Internet service is running and displaying in the foreground of the networked host screen. The context sensing subsystem can also equip with a specifically designed interactive user interface to receive inputs from end-users. All the captured basic context information will be timely delivered to the context model subsystem for processing.
- Context model subsystem** is the place for constructing, hosting and utilizing context models to derive highly abstract and substantive context information, which can be termed as *key context information* (KCI). To build such reliable and effective context models, the data mining techniques and cognitive psychology knowledge may need to be employed. Besides the built context models, the context model subsystem also has a shared database to store and manage the delivered basic context information as well as network condition information. The context models process these delivered data and finally derive the KCI. Note that KCIs can

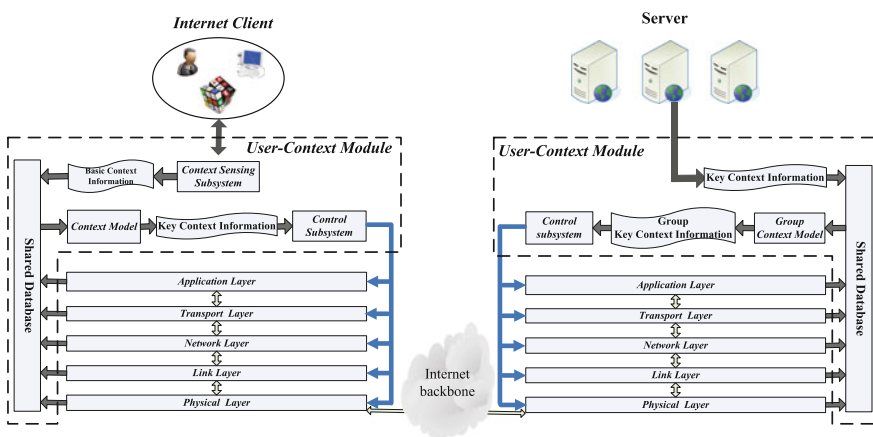


Fig. 3 System block diagram of the User-Context Module with the Internet protocol stack

be derived at both the Internet client side and the server side. The derived KCIs would be transferred and utilized in the context control subsystem.

- **Context control subsystem** is the component for directly interacting with the communication protocols based on the delivered KCIs. For different applications and usage cases, the context control subsystem may interact with distinct protocols in different layers. The context control subsystem may be implemented at either or both the Internet client side and the server side. When interacting with an Internet communication protocol, the context control subsystem does not arbitrarily change the protocol's internal architecture and logic. The context control subsystem normally only cautiously selects the suitable parameters and configurations of the target protocol, which are usually accessible and adjustable. Then it implements the corresponding *control rules* to actively tune those parameters. The *control rules* is a set of rules that specify the actions triggered by the derived KCIs.

Note that designing a context control subsystem is normally different from the traditional cross-layer design, which often merges the established protocol layers or creates novel interfaces for improving the network performance [6]. The designed context control subsystem under the User-Context Module merely tunes and optimizes the accessible parameters and configurations of the communication protocols. Accordingly implementing the context control subsystem and the entire User-Context Module would not impair the functionalities and integrity of the established layered architecture.

In short, the User-Context Module consists of these three indispensable building blocks. More importantly, it provides abundant space and flexibilities for different deployment plans and applications.

2.2 End-User Modeling

Understanding and ascertaining the context information of Internet end-users is not a simple and straightforward task. Therefore, we need to first model the end-users themselves and their interaction behaviors. Fortunately, the Human-Computer Interaction (HCI) and cognitive psychology fields [7] have prepared the frameworks and models to explain the human and their interaction behaviors. One of the widely accepted models is called the Model Human Processor (MHP) [8], and can be easily used to model the common Internet end-users.

As shown in Fig. 4, MHP consists of the perceptual subsystem, cognitive subsystem and motor subsystem. Eyes and ears with the buffer memories are the sensors of the perceptual subsystem to collect and temporarily store the external information. The buffer memories outputs symbolically coded information to the cognitive subsystem. The cognitive subsystem decides how to respond and guide the Motor subsystem to take action. The MHP also indicates that monitoring the Motor subsystem can directly help estimate the cognitive subsystem status. Furthermore, the

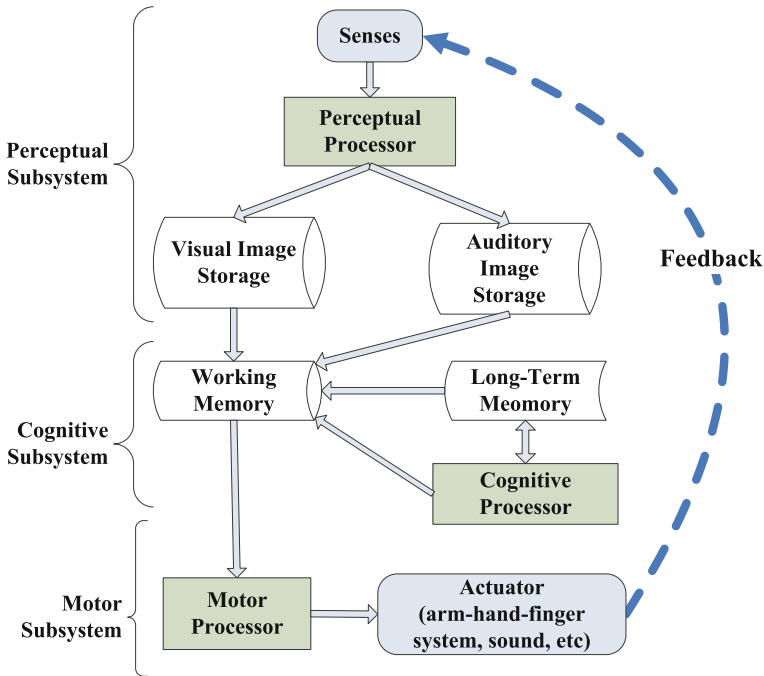


Fig. 4 Model Human Processor (MHP) framework

rationality principle of MHP demonstrates that the human behaviors are based on rational activities and would not randomly and arbitrarily change. Moreover, the *problem space principle* of MHP shows “all rational activities serve to achieve human’s explicit goals, given the task and external information are bounded by his knowledge and processing ability”. The MHP approach has demonstrated its particular strengths in modeling human interaction process, and thus we select it for the User-Context Module.

Based on the MHP model, there are many states can be defined to describe an end-user’s basic status with different Internet services. For the User-Context Module, we firstly define two generic and basic end-user states:

- (1) **User Perception State:** An end-user’s perceptual subsystem and cognitive subsystems are both working on acquiring and processing the information from the corresponding Internet service.
- (2) **User Halt State:** An end-user’s *three* subsystems, i.e., the perceptual, cognitive and motor subsystems, are all not working with the corresponding Internet service.

Given the above defined two end-user states, a running individual Internet service can be associated with only one state at a time, i.e., either the *User Perception State* or the *User Halt State*. Other possible situations of an end-user can be simply defined

as *Unidentified User State*. Note that a running individual Internet service refers to the basic unit of a service. For instance, each open tab inside a Web browser is considered as one individual Web browsing service.

2.3 Key Context Information (KCI)

As indicated earlier, the KCIs serve as the standard outputs of the context model subsystem and the direct inputs of the context control subsystem. Hence, it plays a crucial role in the User-Context Module and a variety of KCIs can be defined depending on different usage scenarios. In this article, we define two fundamental categories of the KCI:

- (1) **Communicating State (CS)**: The Internet end-user keeps in the User Perception State **AND** the corresponding Internet service keeps working.
- (2) **Inactive State (IS)**: The Internet end-user keeps in the User Halt State **OR** the corresponding Internet service stops working.

The defined two categories of the KCI are applicable to most of interaction activities between an end-user and Internet services, regardless of the end-user's identity and the type of Internet services. Moreover, they can be used as the cornerstones to further describe and define more complex KCIs. In addition, when an end-user enters into the unidentified user state, the corresponding KCI can be simply termed as *unidentified state*.

We implemented a context sensing subsystem and a context model subsystem to detect the above defined KCIs, i.e., CS and IS, between an Internet end-user and the typical Internet services including the live multimedia streaming, the Web browsing and the file transfer. Briefly speaking, the context sensing subsystem captures an end-user's eye-gaze direction by a webcam and periodically verifies whether the corresponding Internet application is receiving the mouse or keyboard inputs from the end-user. Meanwhile, it monitors whether the applications are generating visual/audio outputs. The context model subsystem simply adopts the first-order rule-based reasoning approach to ascertain the CS or the IS, and sends the real-time KCIs to the context control subsystem. Note that implementing a User-Context Module would incur computational overhead for collecting and deducing KCIs. However, the latest sensing techniques with the fully-fledged networked hosts help to reduce the overhead considerably while not sacrificing the performance of Internet services. More implementation details and cost assessment can be found in [5].

Note that the above KCIs are defined from the perspective of end-users' presence and their physical interactions with Internet services, while new KCIs can also be deduced from many other perspectives, e.g., an Internet end-user's preference, profile or social networks.

3 Applications of the User-Context Module

3.1 Application I: HTTP Case

With the proposed system architecture and the deduced KCIs, we present two exemplary applications of the User-Context Module to demonstrate its operations and performance gains. The first application shows that the context control subsystem interacts with the Application Layer's HTTP Protocol. More specifically, it actively adjusts the persistent connection timeout parameter in the HTTP protocol to improve the protocol performance in Web service from the perspectives of both end-users and networks.

The HTTP [9] is a stateless communication protocol widely used for transferring Web pages. The HTTP persistent connection mechanism allows reusing the same TCP connection to send and receive multiple HTTP requests. Such a persistent connection mechanism actively reduces network congestion and meanwhile conserves the memory and the CPU usage of networked hosts. However, the HTTP never explicitly defines a closing mechanism for its persistent connection mechanism, and only simply suggests using a simple timeout value for shutting down the persistent connections. In the practical implementations of the HTTP protocol, a fixed timeout value is often imposed, e.g., the Apache HTTP Server uses 15 s and the Microsoft IIS server adopt 120 s as the default connection timeout values. However, improper timeout value can easily impair the network and protocol performance. A small timeout value would decrease the utilization of the established persistent connections, and accordingly increases the network burden and the Web end-users' perceived latency. Meanwhile, a large fixed value would waste and even quickly exhaust the scarce resource on the Web server side (e.g. worker threads), which causes the server instable and the response latency unpredictable. Moreover, it is time-consuming and error-prone to manually configure the timeout parameter. There has been limited research work on adjusting the timeout value for the HTTP to optimize the performance of Web servers [10]. However, none of them directly solves the key issue: *in a Web session, the HTTP protocol cannot properly identify a persistent connection that is being used by a Web end-user or the connection that already transits into a long-term idle state.*

The proposed User-Context Module can be an effective and natural solution to address this problem. On the Web server side, we implemented a context control subsystem to manage the HTTP persistent connection mechanism with the following control rules:

- (1) IF the new Inactive State detected, THEN the context control subsystem immediately notifies the HTTP protocol to *terminate* the relevant connection.
- (2) IF the new Communicating State detected, THEN the context control subsystem immediately notifies the HTTP protocol to *maintain* the relevant HTTP connection and *wait* for the next KCI from the same Web client.

- (3) IF the Unidentified State detected, THEN the context control subsystem can treats it either as the Inactive State or the Communicating State (depending on the real-time server workload and the network condition).

With the above control rules, the traditional HTTP can dynamically adjust its persistent connection mechanism in terms of the Web end-users' real-time browsing behaviors. We conducted the Internet experiments to assess the performance gain on the open source Apache HTTP Server. We modified the source code of the latest Apache HTTP server and recompile it on the Linux 2.6.28 platform to test the newly implemented control rules.

The experimental results show that the User-Context Module significantly shortens the average Web response time. The network trace analysis shows that the User-Context Module effectively extends the lifecycle of the HTTP connections when the Web end-user is interacting with the corresponding Web pages, and thus it would avoid re-establishing of many unnecessary new connections. The Web response time is the time interval between the end-user sending out the HTTP request and the last object of that Web page received. Prior studies have proven that the Web response time greatly impact the end-user's quality of experience (QoE) on the Web browsing. The QoE is defined as "the overall acceptability of an application or service, as perceived subjectively by the end user" [11], which uses the opinion scores to measure the end-user's subjective perception. It has been shown that the opinion scores monotonically increases when the Web response time decreases. Therefore, the implemented User-Context Module effectively enhances the end-user's QoE on Web browsing, and the average end-user's QoE on Web browsing increases from *AVERAGE* to *GOOD* in our Internet experiments.

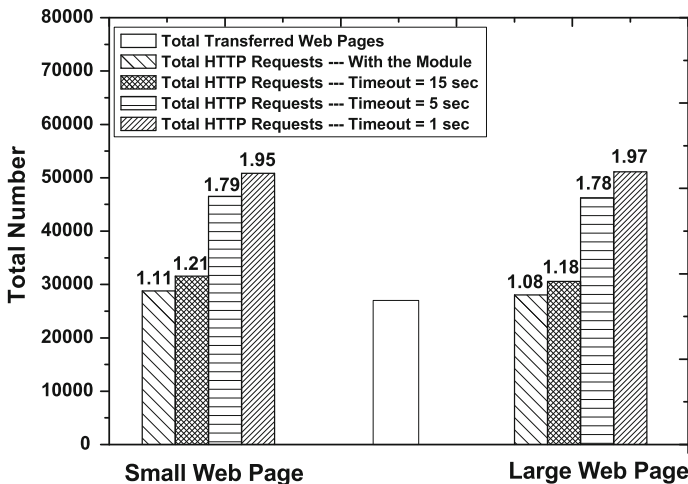


Fig. 5 Ratios of HTTP request number to transferred Web page number

Figure 5 shows the ratio of the HTTP request number in total to the successfully delivered Web page number. The User-Context Module case achieves the smallest ratio in both the small page size and the large page size groups. It is because the User-Context Module reduces the network traffic caused by unnecessary HTTP connection requests and also terminates useless HTTP connections in a timely way on the Web server side. Thus, it considerably alleviate the burden on Internet backbone and the server side. Meanwhile, it also saves the memory usage and the CPU time in the Web servers and the routers.

3.2 Application II: TCP Case

The second application shows that the context control subsystem interacts with the transport layer's TCP protocol. More specifically, it manipulates the advertised window size in TCP to actively re-distribute the access link bandwidth for prioritizing the specific Internet service and eventually enhancing the end-user's QoE.

TCP [12] is a connection-oriented reliable communication protocol, and it also assumes two networked hosts always desire to speak to each other. Therefore, an individual TCP stream often strives to maximize its own throughput unless receiver buffer overflow or network congestion occurs. It has been proven that the TCP stream with a smaller RTT can grab a much larger share of the link bandwidth when competing with the TCP stream with a larger RTT [13]. Accordingly, the TCP favors the Internet applications with shorter RTTs, regardless of end-user's preference and other influential factors. Such a property would always impair an Internet end-user's QoE, especially when some Internet applications need to be prioritized. For example, an end-user may run a file transfer application to download multiple large files and meanwhile run a live multimedia streaming application to watch online TV. However, the multimedia streaming application requires a minimum guaranteed bandwidth, while the file downloading TCP connections with smaller RTTs would take most of the limited bandwidth, where the bottleneck exists at the last mile access link.

The User-Context Module would be a natural and effective solution to empower TCP protocol providing the bandwidth prioritization for such situations. In order to achieve it at the Internet client side, the context control subsystem needs to leverage on the TCP flow control mechanism. The TCP flow control mechanism is designed for avoiding the TCP sender overflowing the TCP receiver's local buffer. More specifically, the TCP receiver maintains a variable in each TCP acknowledgment, called *advertised window*, and actively sets the advertised window size to the amount of its spare buffer. Figure 6 illustrates the TCP advertised window and its physical meaning. We therefore can manipulate the advertised window size to control the TCP sending rate, and the following simply control rules can be implemented on the Internet client side:

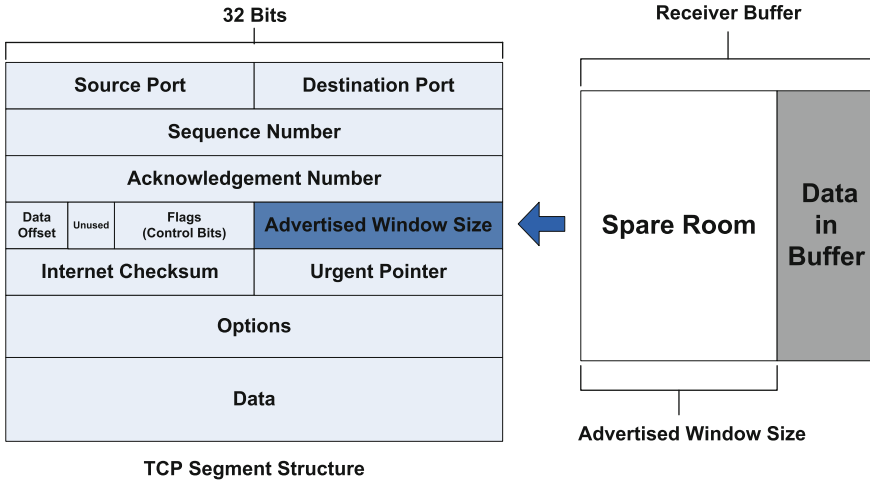


Fig. 6 Advertised window size determined by the spare room of the receiver buffer

- (1) IF the *Communicating State* between the end-user and the live multimedia application is detected, THEN the context control subsystem immediately **reduces** the advertised window size of the concurrent file transfer application until the live multimedia application's bandwidth share exceeds the required threshold.
- (2) IF the *Inactive State* between the end-user and the live multimedia application is detected or the live multimedia application is terminated, THEN the context control subsystem **increases** the advertised window size of the concurrent file transfer application to its initial value.

The above control rules show that a TCP-based delay-sensitive application can be given a higher priority to receive the bandwidth than other delay-insensitive applications. Meanwhile, the delay-insensitive applications can still run concurrently with the leftover bandwidth rather than being forcibly closed or paused. Some prior work suggests the techniques of adjusting the advertised window size to control the TCP sending rate [14]. We conducted the Internet experiment to assess the performance gain from the perspective of the end-users, where multiple participants with their own live multimedia streaming application and file transfer application are tested. The experiment results shows that in all test scenarios, the corresponding live multimedia applications are successfully given the higher priority to receive extra bandwidth, and meanwhile the average QoE value of the end-users increase from *POOR* to *AVERAGE*.

In short, both the HTTP and the TCP cases show that how the User-Context Module works to enhance the protocol performance and accordingly improve the end-user's QoE. Moreover, they demonstrate the context control subsystem designs and operations, and set typical samples for enabling user context utilization in other Internet communication protocols. The technical details and more experiment results can be found in [5].

4 Resource Distribution Framework for the User-Context Module

4.1 Motivation

In the previous sections, we have introduced the core architecture of the User-Context Module, and the two fundamental categories of the KCI have been defined and subsequently deduced by the context models built for different Internet services. The deduced KCI can be directly used to help the Internet to differentiate between the clients that are really resource-starved and the clients that are just ordinary resource consumers. The User-Context Module essentially introduces the KCI into the resource distribution process and provides service differentiation in allocating the resource. More specifically, the control subsystem adaptively allocates the limited resources to real starving Internet clients based on the real-time KCI. Such a design could effectively improve the protocol performance and enhance the end-user's QoE, which has been demonstrated in both the HTTP and the TCP cases.

On the other hand, another critical issue is to motivate the individual Internet client to provide truthful and actual context information, as normal operations of the User-Context Module require that Internet clients provide their actual KCI in a timely way. In many cases, the limited resources are located on the server side or remote end of the network, and accordingly Internet clients are required to share their KCIs with the remote resource owner. However, Internet clients are assumed to be rational and selfish in nature, and therefore they may be not willing to provide their KCIs, especially the negative ones (e.g., Inactive State), because the negative ones may lead to fewer allocated resources or a lower priority.

In this section, we present a novel resource distribution framework that provides context-driven and QoE-aware service differentiation, which means that starving clients are prioritized in resource allocation to enhance the corresponding end-user's quality of experience (QoE). Moreover, the framework actively motivates each Internet client to consistently provide its actual context information and to adopt moderate competition policies, given that all clients are selfish but rational in nature. The selfish nature results in the Internet clients competing aggressively for any limited resource over the Internet, typically including the resource held by servers.

To further aid understanding of the above-described issues, we take the World Wide Web system (Web system) as an illustrative example. Web system adopts the client-server architecture and leverages on the HTTP protocol for transferring Web pages between the Web server and the Web clients. On the Web server side, the child process usually creates multiple worker threads to handle any incoming HTTP connection requests: normally, one worker thread only serves one HTTP connection at a time on a first-come-first-served basis. Too many worker threads in Web server can easily cause thrashing in virtual memory system and considerably degrade server performance. In practice, a fixed limit is always imposed on the maximum number of worker threads: for example, the default maximum number in an Apache HTTP Server 2.2 is set to 256. Therefore, the worker threads held by the Web server always

become the limited resource in the Web system. On the Web client side, HTTP/1.1 specifies that “*Clients that use persistent connections SHOULD limit the number of simultaneous connections that they maintain to a given server. A single-user client SHOULD NOT maintain more than 2 connections with any server or proxy*”. However, most of commercial Web browsers violate this restriction: the default maximum value of Firefox is set to 6 parallel persistent connections per server, and 8 persistent connections per proxy. The Google Chrome and Internet Explorer also aggressively set at least 6 parallel persistent connections per server as their default settings. Consequently, the limited worker threads in a popular Web system are usually subjected to excessive competition from such aggressive and unconstrained Web clients.

As described in the previous sections, today’s Internet protocols and its design principle simply assumes that end-users behind their network hosts desire to communicate with the other end. Hence, the traditional Web system also assumes that each of the allocated worker thread is being used by a hungry end-users through the established HTTP connection. Accordingly, it usually handles all incoming HTTP requests equally and maintains a first-in, first-out (FIFO) queue with the drop-tail queue management [15]. To handle and control the established HTTP connections, the Web system has to adopt a fixed timeout mechanism by releasing the worker thread [10].

As shown in the last section (HTTP application case), when the User-Context Module is introduced and implemented, a Web system is able to differentiate between the worker threads that are being used by real hungry end-users and the worker threads that are just grabbed by aggressive Web browsers. With such crucial information and key knowledge, the User-Context Module can enable many service differentiation solutions. However, since providing the Inactive State (IS) information to Web server may result in fewer and even no allocated worker threads, any selfish Web client may not be willing to share such negative KCIs. We therefore propose a novel resource distribution framework with the following three explicit design objectives:

1. The framework should provide *service differentiation* in allocating limited resources in terms of the derived KCI.
2. The framework should encourage selfish but rational Internet clients to share their *actual KCIs*, especially the negative ones, such as *IS*.
3. The framework should motivate all Internet clients to adopt a *moderate competition* policy for the limited resource.

4.2 Framework Workflow

Assume that μ basic units of the limited resource are held by the server side, which is termed as *resource owner* in this framework. The limited resource may be in different types, such as worker thread, CPU time, bandwidth, etc. A finite set of clients, denoted by $P_i, i \in I = \{1, 2, \dots, N\}$, actively competes for the limited resource. All clients agree to transfer and update their KCIs to the resource owner through an

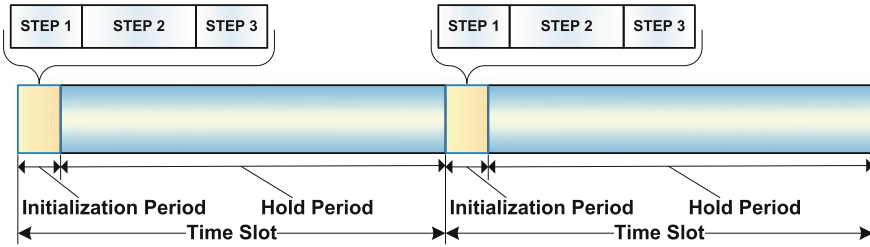


Fig. 7 Time slot divided into the Initialization Period and the Hold Period

interoperable communication mechanism. The resource owner maintains a database to store the recent updates of KCIs with the corresponding timestamp. On the resource owner side, the time domain is divided into fixed-sized time slots $T_j, j \in \{1, 2, \dots, +\infty\}$, and as shown in Fig. 7, each individual time slot is further divided into two phases: an Initialization Period and a subsequent Hold Period. The resource distribution process only occurs in each of the Initialization Period. The Initialization Period normally occupies a small portion of the entire time slot, typically 5–10 %.

Within each of the Initialization Period, the basic workflow of the resource distribution framework, i.e., the interaction steps between the resource owner and Internet clients, can be simply described as below:

1. Based on historical and current KCIs, resource owner firstly runs an algorithm, called *Willingness Update Algorithm (WUA)*, to calculate the so-called willingness value for each client. The willingness value, $w_i(T_j)$, is the amount of resource that resource owner is currently willing to provide to client P_i at the time slot T_j . After running WUA, resource owner then informs the assigned willingness value to each Internet client.
2. After obtaining the willingness value, each client, say P_i , will decide a bidding value $b_i(T_j)$ with its own strategy, and accordingly send it to the resource owner. The bidding value $b_i(T_j)$ is the resource amount that client P_i currently expects to get during time slot T_j .
3. Based on the received bidding values from STEP 2 and the willingness values from STEP 1, resource owner performs the so-called *Resource Distribution Algorithm (RDA)* to compute final resource distribution result. The result $x_i(T_j), \forall i \in I$, is the amount of the resource finally given to client P_i at the current time slot T_j .

Figure 8 illustrates the above three-step procedure, and the detailed algorithms for the *Willingness Update Algorithm* in STEP 1 and the *Resource Distribution Algorithm* in STEP 3 can be found in [4].

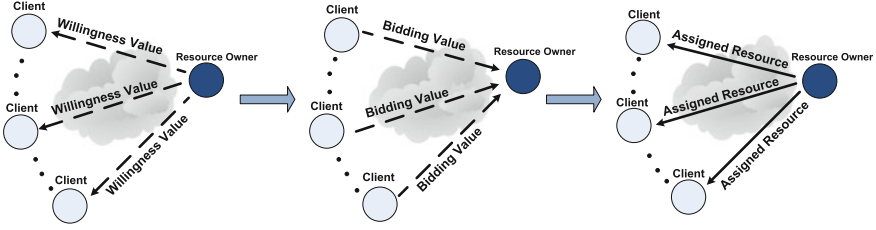


Fig. 8 Three steps in the basic workflow of the resource distribution framework

4.3 Framework Properties

The three-step basic workflow determines an interaction process between the resource owner and all of its clients. We can analyze such a process using non-cooperative game theory: all clients are the game players, and they can independently choose their own bidding strategy for maximizing their individual payoff; the implemented WUA and RDA can be regarded as the utility functions; the payoffs for the game players are the final distribution results.

Lemma 1 *Under the proposed framework and the designed WUA and RDA, any Internet client, say P_c , who bids the assigned willingness value, i.e., $b_c(T_j) = w_c(T_j)$, can be guaranteed to receive its bidding resource, i.e., $x_c(T_j) = b_c(T_j)$, regardless of other clients' bidding strategy.*

Lemma 2 *Under the proposed framework and the designed WUA and RDA, the bidding strategy profile $B^*(T_j) = \{b_c^*(T_j) : b_c^*(T_j) = w_c(T_j), \forall c \in I\}$ is the unique pure-strategy Nash equilibrium.*

Proposition *Under the proposed resource distribution framework and the designed WUA and RDA, the best policy for any client is to share its actual KCI, i.e., either the CS or the IS, and adopt a moderate bidding strategy for competing the limited resource.*

The proof of Lemmas 1, 2 and Proposition can be found in [4].

4.4 Illustrative Case and Experimental Results

To demonstrate how the framework operates in practice, we use the mentioned Web system as an exemplary application. The Web server (resource owner) has the limited number of worker threads, which are normally under demand of its clients. The individual Web client is essentially the Web browser with its end-user. Assuming that the KCIs of Web clients, namely IS and CS, is able to timely derived and transferred to the server side, accordingly at the Initialization Period of each time slot, Web

server will firstly perform the given WUA algorithm and then immediately informs the willingness values to the corresponding client. After that, each Web client has to determine how many resource (i.e., worker threads) to bid. In this case, The bidding value is actually the number of parallel HTTP connection initiated by the Web browser. Based on the Proposition 1, any rational Web client, say P_i , would act moderately to set a bidding value $b_i(T_j)$, which is close to the given willingness value $w_i(T_j)$. For example, the client P_i can choose a simple bidding strategy:

$$b_i(T_j) = \max\{1, \lceil w_i(T_j) \rceil\}, \quad (1)$$

where $\lceil \cdot \rceil$ is the ceiling function. Accordingly, the client side should automatically adjust the number of parallel HTTP connections with the Web server. Given that $x_i^r(T_{j-1})$ is the established HTTP connections during last time slot T_{j-1} , the control rules at the client side would be based on the bidding strategy (1):

- (1) IF $b_i(T_j) > x_i^r(T_{j-1})$, THEN the Web browser would immediately **initiate** $b_i(T_j) - x_i^r(T_{j-1})$ new HTTP persistent connection requests to the server.
- (2) IF $b_i(T_j) \leq x_i^r(T_{j-1})$, THEN the Web browser would take **no** action.

The above control rules indicate Web clients do not need to conduct the connection termination tasks, and the termination actions are left for the Web server.

In STEP 3, the Web server has all the bidding values and runs the designed RDA to obtain the resource distribution results, i.e., $x_i(T_j)$, $\forall i \in I$. Given that $\beta_i(T_j) = \lceil b_i(T_j) - x_i(T_j) \rceil$, the control rules below can be implemented on the Web server:

- (1) IF $\beta_i(T_j) > 0$, THEN the Web server would **gracefully terminate** $\beta_i(T_j)$ established HTTP connections with Web client P_i .
- (2) IF $\beta_i(T_j) = 0$, THEN the Web server would take **no** action for client P_i .

The above control rules actually enable Web server to get back the worker threads from the aggressive Web clients and accomplish the final distribution results. Considering the running framework and control rules, the rational commercial Web browsers (their designers) would stop arbitrarily increase the limit of parallel connections per server, and would adopt a proper competition policy similar to the moderate bidding strategy (1).

We have implemented the above described framework and the control rules on an experimental Web system. On the Web server side, we have selected Apache HTTP Server and modified its source code for the HTTP protocol and the thread pool management. On the Web client side, we use a HTTP request generator to emulate multiple Web clients. Each client switches between the CS and the IS and follows a similar state transition model given in [16], where the user sessions are exponentially distributed.

Experiment (Service Differentiation). Given 500 clients compete for 256 worker threads in the Apache HTTP server. All clients adopt the moderate bidding strategy (1) and meanwhile actively share their real KCIs. Figure 9 shows the average number of worker threads by the different classes, which are categorized by the

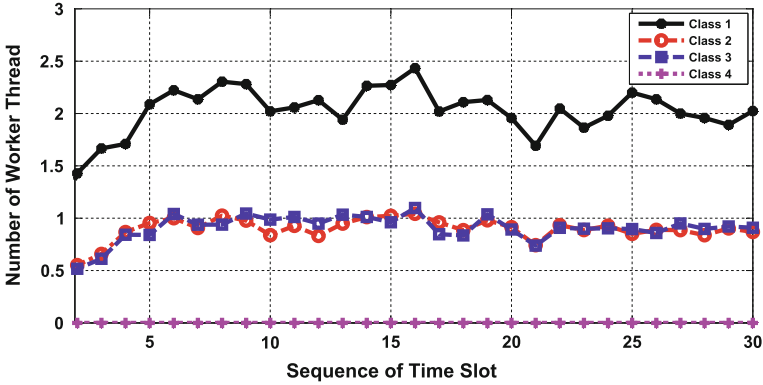


Fig. 9 Service differentiation under the resource distribution framework

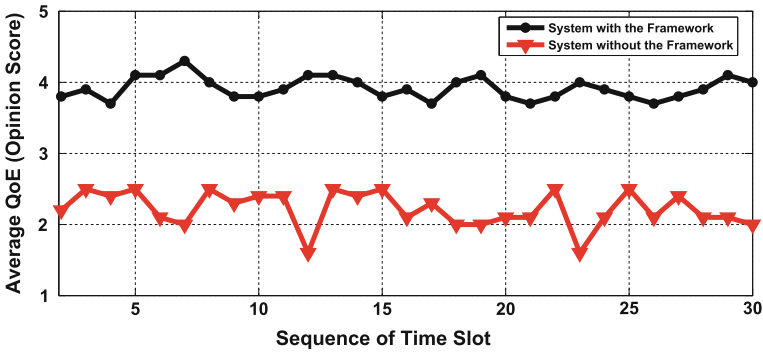


Fig. 10 A comparison of the average end-user’s QoE on Web browsing

designed WUA. Figure 10 further demonstrates the enhancement of average end-user’s QoE on Web browsing.

In the conventional Web system, which usually adopts FIFO and drop-tail queue management, 500 concurrent clients competing for 256 worker threads would cause at least half of clients simply blocked or waiting in the connection queue. Figure 9 shows that the Web clients, who are categorized into class C_1 , obtain around 2 worker threads on average from the Web server. Such clients are normally the honest clients and actively report their negative KCIs. The Web clients, who are categorized into other classes, obtain less resource and even nothing from the Web server.

The assigned worker threads would directly affect the Web page download time, which influences the end-user’s QoE. We adopt the existing studies on the quantitative relationship between QoE and download time by Shaikh [17]. It shows that the exponential relationship holds: $QoE = 4.836 * \exp(-0.15T)$, where T is the Web page download time and QoE is measured with the opinion score (5 = Excellent, 4 = Good, 3 = Average, 2 = Poor, 1 = Bad). For the purpose of comparison, we compare the results with a conventional Web system. Figure 10 shows the system

with the proposed framework having a much higher QoE than the conventional one. The main reason is the framework successfully provides service differentiation and thus allocates most of worker threads to the starving clients. On the other side, the traditional Web system simply treats all clients equally without any differentiation and prioritization.

In short, the experiment results show that a novel resource distribution framework can provide context-driven and QoE-aware service differentiation. Moreover, it incentivizes context-sharing and moderate competition by leveraging on the selfish but rational nature of Internet clients. For more experiment results, please refer to [4].

5 Related Work

5.1 Internet Protocol Stack Design

In order to partition the complicated data communication tasks, the five-layer Internet protocol stack [18] and the seven-layer Open Systems Interconnect (OSI) model [19] are designed. Many fundamental and respected principles have been gradually introduced and implemented in its layered architecture and communication protocols, such as packet switching for multiplexing [20], global addressing for routing datagrams [19] and end-to-end arguments for defining communication protocols [21]. Regulated by such established design principles, Internet designers do their job: they design, revise, deploy and configure the communication protocols.

Recently, there have been relevant research proposals relevant to extending the concept of Internet client, particularly the research on the identifier-locator split architecture. The identifier-locator split architecture uses independent name spaces to recognize the host and the host address separately. For example, MILSA (Mobility and Multihoming supporting Identifier Locator Split Architecture) [22] introduces a new Host-ID sub-layer into the traditional Network Layer to separate networked host from its locator. MILSA and other similar architectures, such as HIP [23] and LISP [24], attempt to enable Internet end-users, rather than the networked host, to be the destination of Internet services. Hence, to some extent, their studies incorporate Internet end-users into the architecture of the Internet protocol stack, although no context information is considered. More details can be found in [25] and the references therein.

5.2 Context-Aware Computing

The ubiquitous computing [26] has evolved to a paradigm known as context-aware computing. There have been several approaches for the context information acquisition, which typically includes the middleware based approach [27], the direct sensor

access approach and the context server based approach. The middleware based approach adopts the encapsulation to separate and hide low-level sensing to facilitate rapid prototyping and implementing of a context-aware system. This approach has been widely used in the context-aware systems, such as Gaia systems [28] and SOCAM [29]. Our User-Context Module architecture design also adopt the middleware based approach for acquisition of the context information of Internet clients. The existing context models can be categorized as the logic based model, ontology based model, object oriented model as well as the key-value model, which are well summarized in [30]. The models we designed for the User-Context Module can be classified as the logic based model.

In the existing context-aware systems, Internet protocol stack always serves as the long distance data communication carrier [31]. However, limited prior studies consider enabling the Internet to utilize the context information and be context-aware. The context-aware Web service [32] can be regarded as a good attempt in this direction on the application level. They mainly employ Web end-user's context information to support Web content adaptation [33], communication optimization [34] as well as security and privacy control [35]. Nevertheless, none of them introduce Internet client's context information directly into the underlying communication protocols.

5.3 *Quality of Experience (QoE)*

One of the main objectives of the User-Context Module application is to enhance the end-user's QoE. The ITU Telecommunication (ITU-T) Standardization defines QoE as "*the overall acceptability of an application or service, as perceived subjectively by the end user*" [11]. QoE [36, 37] can be simply interpreted as human's subjective perception on the performance of communication systems. Particular attention is given to measure QoE not only in terms of the Quality of Service (QoS) parameters [38], but a joint consequence of a communication context environment, the characteristics of the service and the network performance. Brooks and Hestnes [39] propose a structured assessment approach to describe end-user's QoE in a clearer and comprehensive way. The progress on enhancing and modeling QoE would directly impact Internet design and eventually benefit the Internet end-users.

6 Summary and Conclusion

The main contributions and novelty of this work can be summarized as below:

- It firstly reveals the fact that certain context information of Internet clients can be directly used in Internet protocol stack to help enhance improve protocol performance and end-user's QoE.

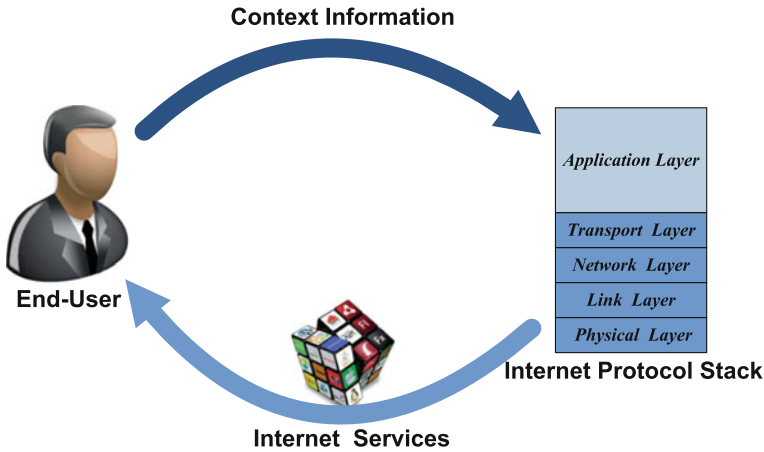


Fig. 11 New communication pathway and the closed communication loop

- The designed User-Context Module explicitly demonstrates how to capture, ascertain and utilize the context information in a systematic way.
- The designed resource allocation framework illustrates how to provide the context-driven service differentiation, incentivize actual context sharing and moderate competition among Internet clients.

Figure 11 illustrates that this work essentially builds a unique communication pathway for transmission of context information from the end-user side to the Internet protocol stack side. Considering the conventional pathway from the Internet protocol stack through Internet services to the client side, our work essentially establishes a closed communication loop.

Introducing the context information into the Internet communication protocols and de-conflating the traditional Internet client have a large exploration space, and we tentatively list some potential aspects:

- *Context Usage in Future Internet and IoT Architecture:* there are a number research projects on architectural design for the next generation Internet and IoT using the clean-slate approach. The context information can be explicitly incorporated into these new architectures. Meanwhile, the User-Context Module would expose a set of standard application programming interfaces (API) to facilitate the utilization by both the existing and new architectures.
- *Advanced End-User Models and KCIs:* human’s cognitive mechanism directly influences the Internet end-user’s interaction behaviors. The latest progress in sensor technology, brain-computer interface (BCI) as well as neuroscience hold a great promise for building more accurate end-user interaction models. With the advanced end-user models, new KCIs can be accordingly derived to describe the interactions among end-users, sensors, network devices and services.

- *New Applications of the User-Context Module*: the User-Context Module works with several communication protocols and layers under Internet architecture, e.g., the Application Layer's HTTP protocol and the Transport Layer's TCP protocol. It is expected to explore the interactions with other communication protocols, layers and architectures.

Finally, we hope this work can inspire network designers and open up a new realm for innovations on both the Internet and IoT design.

References

1. D. D. Clark, "The design philosophy of the DARPA Internet Protocols," *SIGCOMM Comput. Commun. Rev.*, vol. 25, no. 1, pp. 102–111, 1995.
2. R. Braden, "Requirements for Internet Hosts – Communication Layers," *IETF RFC 1122*, 1989.
3. A. K. Dey, "Understanding and Using Context," *Personal Ubiquitous Computing*, vol. 5, no. 1, pp. 4–7, 2001.
4. Y. Lu, M. Motani, and W. C. Wong, "A qoe-aware resource distribution framework incentivizing context sharing and moderate competition," *IEEE/ACM Transactions on Networking*, 2015.
5. Y. Lu, M. Mehul, and W. C. Wong, "When ambient intelligence meets the internet: User module framework and its applications," *Computer Networks*, vol. 56, no. 6, pp. 1763–1781, 2012.
6. V. Srivastava and M. Motani, "Cross-layer design: a survey and the road ahead," *IEEE Communication Magazine*, vol. 43, no. 12, pp. 112–119, 2005.
7. A. Sears and J. A. Jacko, *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, Second Edition (Human Factors and Ergonomics)*. Lawrence Erlbaum Associates, 2002.
8. S. K. Card, T. P. Moran, and A. Newell, *The psychology of human computer interaction*. L. Erlbaum Associates Inc., 1983.
9. R. Fielding *et al.*, "Hypertext Transfer Protocol – HTTP/1.1," *IETF RFC 2616*, 1999.
10. A. Sugiki, K. Kono, and H. Iwasaki, "Tuning mechanisms for two major parameters of Apache web servers," *Softw. Pract. Exper.*, vol. 38, no. 12, pp. 1215–1240, 2008.
11. ITU-T, Rec. P. 10/G. 100, Amendment 2: New Definitions for Inclusion in Recommendation ITU-T P.10/G.100, 2008.
12. J. Postel, "Transmission Control Protocol," *RFC 793*, 1981.
13. T. V. Laskshman and U. Madhow, "The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss," *IEEE/ACM Transactions on Networking*, vol. 5, no. 3, pp. 336–350, 1997.
14. P. Mehra, C. D. Vleeschouwer, and A. Zakhori, "Receiver-Driven Bandwidth sharing for TCP and its Applications to Video Streaming," *IEEE/ACM Transactions on Multimedia*, vol. 7, no. 4, pp. 740–752, 2005.
15. J. Wei and C. Z. Xu, "eQoS: Provisioning of Client-Perceived End-to-End QoS Guarantees in Web Servers," *IEEE Transactions on Computer*, vol. 55, no. 12, pp. 1543–1556, 2006.
16. P. Gill *et al.*, "Characterizing user sessions on YouTube," in *Proc. Annual Multimedia Computing and Networking Conference*, San Jose, CA, USA, Jan. 2008.
17. J. Shaikh, M. Fiedler, and D. Collange, "Quality of Experience from User and Network Perspectives," *Ann. Telecommun.*, vol. 65, pp. 47–57, 2010.
18. J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*. Pearson Education, Inc., 2008.
19. H. Zimmermann, "OSI Reference Model–The ISO Model of Architecture for Open Systems Interconnection," *IEEE Transactions on Communications*, vol. 28, no. 4, pp. 425–432, 1980.

20. D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden, "Tussle in cyberspace: defining tomorrow's internet," *IEEE/ACM Trans. Netw.*, vol. 13, no. 3, pp. 462–475, 2005.
21. J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-end arguments in system design," *ACM Trans. Comput. Syst.*, vol. 2, no. 4, pp. 277–288, 1984.
22. J. Pan, R. Jain, S. Paul, and C. So-in, "MILSA: A New Evolutionary Architecture for Scalability, Mobility, and Multihoming in the Future Internet," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 8, pp. 1344–1362, 2010.
23. R. Moskowitz and P. Nikander, "Host Identity Protocol (HIP) Architecture," *IETF RFC 4423*, 2006.
24. D. Farinacci, V. Fuller *et al.*, "Locator/ID Separation Protocol (LISP)," *IETF Internet Draft*, 2012.
25. S. Paul, J. Pan, and R. Jain, "Architectures for the future networks and the next generation Internet: A survey," *Computer Communications*, vol. 34, no. 1, pp. 2–42, 2011.
26. M. Weiser, "The Computer for the Twenty-First Century," *Scientific American*, September, 2002.
27. H. Chen, "An intelligent broker architecture for pervasive context-aware systems," Ph.D. dissertation, University of Maryland, 2004.
28. M. Roman *et al.*, "A Middleware Infrastructure for Active Spaces," *IEEE Pervasive Computing*, vol. 1, no. 4, pp. 74–83, 2002.
29. T. Gu, H. K. Pung, and D. Q. Zhang, "A Service-oriented middleware for building context-aware services," *Journal of Network and Computer Applications*, vol. 28, no. 1, pp. 1–18, 2005.
30. M. Baldauf, S. Dustdar, and F. Rosenberg, "A Survey on Context-aware Systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007.
31. J. M. Batalla, G. Mastorakis, C. X. Mavromoustakis, and J. urek, "On cohabitating networking technologies with common wireless access for home automation systems purposes," *IEEE Wireless Communication Magazine*, 2016.
32. H. L. Truong and S. Dustdar, "A Survey on Context-aware Web Service Systems," *International Journal of Web Information Systems*, vol. 5, no. 1, pp. 5–31, 2009.
33. B. Han, W. Jia, J. Shen, and M. C. Yuen, "Context-awareness in Mobile Web Services," *Parallel and Distributed Processing and Applications*, vol. 3358, pp. 519–528, 2008.
34. I. Matsumura *et al.*, "Situated Web Service: Context-aware Approach to High-Speed Web Service Communication," in *Proc. IEEE Conf. on Web Services*, Chicago, IL, Sept. 2006.
35. C. D. Wang, T. Li, and L. C. Feng, "Context-aware Environment-Role-Based Access Control Model for Web Services," in *Proc. IEEE Conf. on Multimedia and Ubiquitous Engineering*, Busan, Korea, April 2008.
36. R. Jain, "Quality of Experience," *IEEE Multimedia*, vol. 11, no. 1, pp. 95–96, 2004.
37. Nokia, "Quality of Experience (QoE) of mobile services: Can it be measured and improved," White Paper, Finland, 2004.
38. M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *IEEE Network*, vol. 24, no. 2, pp. 36–41, 2010.
39. P. Brooks and B. Hestnes, "User measures of quality of experience: why being objective and quantitative is important," *IEEE Network*, vol. 24, no. 2, pp. 8–13, 2010.

Security Challenges of the Internet of Things

Musa G. Samaila, Miguel Neto, Diogo A.B. Fernandes,
Mário M. Freire and Pedro R.M. Inácio

Abstract The Internet of Things (IoT) is an environment in which ordinary and complex consumer products, buildings, bridges, animals or even people, etc. are embedded with sensors, equipped with a variety of communication technologies and given unique identifiers that can enable them connect to the Internet. This allows them to talk to each other, collect data and transfer data over the Internet. IoT has the potential to enhance the way we do things by increasing productivity and efficiency. It also has the prospects of delivering significant business benefits. Nonetheless, implementing secure communication in the IoT and integrating security mechanisms into some of its devices have been a major impediment to its progress, resulting in many privacy concerns. Although IoT is a hybrid network of the Internet, many security solutions for the Internet cannot be directly used on the resource-constrained devices of the IoT, hence the need for new security solutions. In this chapter, we discuss the security challenges of the IoT. First, we discuss some basic concepts of security and security requirements in the context of IoT. We then consider fundamental security issues in the IoT and thereafter highlight the security issues that need immediate attention.

M.G. Samaila (✉) · M. Neto · D.A.B. Fernandes · M.M. Freire · P.R.M. Inácio
Department of Computer Science, Instituto de Telecomunicações, University of Beira Interior, Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal
e-mail: mgsamaila@it.ubi.pt

M. Neto
e-mail: migasn@gmail.com

D.A.B. Fernandes
e-mail: dfernandes@penhas.di.ubi.pt

M.M. Freire
e-mail: mario@di.ubi.pt

P.R.M. Inácio
e-mail: inacio@di.ubi.pt

M.G. Samaila
Centre for Geodesy and Geodynamics, National Space Research and Development Agency,
P.M.B. 11, Toro, Bauchi State, Nigeria

1 Introduction

Internet of Things (IoT) is a paradigm and concept of ubiquitous connectivity, where nearly all virtual and physical electrical *things/objects* (electrical appliances) are expected to be embedded with Internet Protocol (IP) suite to enable them to connect to each other via the Internet. Using unique identifiers, these innumerable connected smart devices can communicate over multiple networks, forming a larger IP based network of interconnected *things*, or an ecosystem of connected *devices* [1]. The devices include common *things* such as lamps, bread toasters, wearable devices, toothbrushes and virtually every useful *thing* one can imagine. Other devices that participate in this communication include embedded sensor devices used in Machine-to-Machine (M2M) communications, implantable and other medical devices, smart city, smart energy grids, Vehicle-to-Vehicle (V2V) Communications, etc. Another essential component of the IoT phenomenon is the Machine-to-Human (M2H) communication. Constrained devices with limited computational power, memory and energy resources, and having lossy network channels like Wireless Sensor Networks (WSNs) responsible for information gathering also constitute an integral part of the IoT ecosystem.

The enabling technologies that will drive the future of IoT include sensor technology, Radio Frequency Identification (RFID) [2], Micro-Electro-Mechanical Systems (MEMS) technology, nanotechnology and smart things, energy harvesting, cloud computing along with the ongoing evolution of wireless connectivity and the transition towards Internet Protocol Version 6 (IPv6). The interaction between these technologies is breeding a new form of communication that is enabling seamless exchange of information between systems, devices and humans, thereby serving as an enabling platform for the IoT. Consequently, ordinary electronic devices and many other *things* are now manufactured with communications, computing and digital sensing capabilities. Such functionalities enable these devices to sense their environment and participate in data exchange. Thus, *things* are said to have *digital voice*.

Just like the Internet, this ecosystem of connected *objects* has the potential to change virtually everything in the world [3]. Today, IoT is rapidly becoming one of the most hyped technologies in both academia and industry. Moreover, the transition from the use of traditional devices to the use of Internet connected smart devices is accelerating at an alarming pace. As a growing network of everyday *objects*, IoT represents the next major economic revolution that will be enabled by the Internet [4]. Experts believe that we are entering into a new era, one in which the IoT will replace the traditional Internet that we know today. In this new era of connectedness, business playing ground is swiftly changing as a result of the impact of IoT, which is now creating new business opportunities, new sources of revenue and improved processes. According to Gartner, the global economic benefits of IoT will be close to \$2 trillion [5]. Gartner also says that 4.9 Billion *things* will be connected before the end of 2015 [6], 6.4 billion in 2016 [7], and the number is expected to reach 25 Billion in 2020 [6] as more light bulbs, wearable devices like

watches, and cars connect to the Internet. Additionally, in order to consolidate business operations around the IoT, companies and researchers are coming up with more innovative solutions for the IoT [8, 9]. This is good news for operators, investors and every player with a stake in the IoT.

In spite of the potential advantages and benefits of IoT, and the fact that it is affecting increasing number of businesses and creating new exciting opportunities, however, there are still many security and privacy challenges that need to be addressed if IoT is to be willingly embraced by the business community, policy makers and the society at large. The idea of interconnecting incalculable number of remotely controlled smart devices (some of which are resource-constrained) via the Internet is raising alarms about security and privacy of users. This chapter, therefore, is focused on examining and describing the relevant security and privacy challenges posed by IoT connected devices that have been identified in the literature, and the difficulties of dealing with them. We focus specifically on connected devices that have limited resources in terms of memory, computing power and energy.

This chapter is structured as follows. Section 1 presents a brief overview of IoT, where we briefly consider the enabling technologies, the prospects and the need for securing the IoT. Section 2 looks at the concepts of security in the IoT. Section 3 examines the fundamental IoT security issues. Section 4 highlights some IoT security issues that need immediate attention. Finally, Sect. 5 presents our conclusions.

2 Concepts of Security in the Internet of Things

In this section we discuss some basic concepts of IoT security. Specifically, we present a general overview of IoT security, security goals for IoT, where the IoT security becomes delicate, size and heterogeneity of *things*, and the privacy concerns in the IoT.

2.1 A General Overview of Internet of Things Security

As the economic, social and technical significance of IoT is growing day by day, IoT is not only attracting the interest of investors, users and researchers but also nefarious users who are working hard to turn connected devices into weapons for cyber attacks or stealing data. Although the general idea behind the creation of IoT is to make life easier, it is obvious that interconnecting a large amount of non-traditional computing devices, such as smart water sprinklers and smart refrigerators, over the Internet, can be dangerous, and will bring a major change to individual and enterprise security and privacy. This is because a couple of security issues on one device can affect a number of devices on the same or different

network. An attacker can, for example, use compromised connected devices like smart TVs to launch Denial-of-Service (DoS) or Man-in-the-Middle (MitM) attacks on other networks and services. For instance, security researchers from Proofpoint [10] revealed cyber attacks that may probably be the first IoT botnet attacks to be using about 100,000 ordinary smart household gadgets including smart TVs, home-networking routers, connected multi-media centres and a refrigerator. The attacks were observed between December 23, 2013 and January 6, 2014, in which hackers used the compromised appliances and sent out about 750,000 malicious spam emails, targeting a number of enterprises and individuals all over the world.

While security and privacy concerns in the Internet are not new and still present challenges, security and privacy issues in the IoT pose additional and unique challenges. Due to the increasing popularity of the IoT, it is certain that more and more varieties of devices that gather all sorts of data will be deployed or embedded into different systems, organizations, homes, etc. As this massive data capturing activities constantly increase, concerns for corporate and individuals security and privacy will also increase. Most of these devices pose a number of potential security risks that could be exploited by malicious entities to harm legitimate users. Given the capabilities of some of the smart devices, some users may not even know that they are being recorded or tracked.

As IoT devices generate more data, protection of data will continue to be an issue. As a consequence, IoT is hardly safe for processing sensitive information, especially as the technology become more pervasive and integrated more into our everyday lives. The staggering variety of devices comprising the IoT along with the different constraints associated with such devices further compound the problem. One motivating factor that will make the IoT data a more attractive target for malicious users is the fact that data is directly accessible over the Internet, and can potentially be accessed through poorly secured smart devices that have little or no security measures. Hence the need to ensure that each device is properly secured is of paramount importance. Addressing such challenges and ensuring security in the IoT would prevent intruders from exploiting vulnerabilities that could be used to compromise personal information.

2.2 Security Goals for the Internet of Things

As a fundamental component of every network design, security is among the biggest obstacles standing in the way of full deployment of the IoT. Recently, there have been numerous successful attacks on the IoT. Some of the attacks come from white hat hackers who want to examine the performance of these IoT devices and find out how vulnerable they are to intrusion. Other attacks come from malicious entities who exploit known vulnerabilities in such devices in order to discover sensitive information for personal gain. Considering the rising incidence of cyber attacks targeting the IoT [11], there is need to plan and implement a good security strategy for the IoT. This can only be achieved if new security goals are identified

and implemented in the design process of IoT devices and systems. Introducing security goals or requirements early in the design process can help fortify information security, reduce risks and improve risk management.

Security goals (or requirements) are basically the fundamental principles that describe functional and non-functional objectives needed to be satisfied so as to achieve the security features of a system. Explicitly outlining security goals is key to baking security into a design process. In information security, there are three fundamental principles of security, namely confidentiality, integrity and availability, which are often referred to as the CIA triad. These fundamental principles have been broadened over time to include other security requirements such as authentication, access control, non-repudiation and privacy preservation. Essentially, these fundamental principles constitute the central objective of any security program of an Information Technology (IT) system. Usually, every security mechanism, safeguard and control is put in place in order to provide one or more of these requirements. Similarly, every threat, risk and vulnerability is evaluated for its potential ability to compromise one or more of these principles. But since every system has unique security goals, the level of security needed to achieve these requirements differ from one system to another. The same also applies to all IoT systems.

Considering the diversity in IoT system functions, device types and deployment locations, there is no *one size fits all* set of security requirements that can be effectively used for every application area within the IoT ecosystem. Although IoT systems vary considerably from one application to another, some applications may share common security goals [12]. The principal IoT security requirements to counter most security threats can be characterized according to the following fundamental security attributes [13, 14], namely confidentiality, integrity, availability, authentication, access control, non-repudiation, secure booting and device tampering detection. It is important at this juncture to state that, depending on its area of application, an IoT system or device may require some, all, or more of the above security requirements. Similarly, the degree of implementation of a particular requirement depends on the application scenario, for example, some scenarios may require low, medium or high degree of implementation. Below is a brief description of the given security requirements.

2.2.1 Confidentiality

Connecting unsecured, small and inexpensive smart devices to the Internet exposes users to massive security risks. Vulnerabilities in such devices serve as entry points through which cyber-criminals can access private or corporate assets. No doubt, IoT presents potential for greater comfort and efficiency at home and in workplace, thereby improving living conditions. However, without proper security measures, such benefits may turn into nightmare.

Data confidentiality is an important security requirement that ensures that only those authorized to view the information are given access to the data and that no sensitive information gets to unauthorized persons. Lately [15, 16], there are growing concerns over data confidentiality in the IoT, since connected devices are sometimes used for transmitting confidential data. The level of confidentiality needed depends on application scenario and implementation. For instance, the level of confidentiality required for securing smart water sprinklers will definitely not be the same as that required to secure devices in critical sectors like the healthcare industry. The measures needed to be undertaken in order to provide End-to-End (E2E) message secrecy in the case of patient confidentiality should be strong enough such that access is restricted to only those authorized to view the message.

2.2.2 Integrity

IoT devices often store sensitive user data locally. For example, a user can store bank details, social security number, contacts, travel preferences and favourite music play lists. A number of people have concerns that their sensitive data could be accessed or manipulated over the IoT. Apart from personal user information, network service providers and manufacturers of smart devices can, as well, store important data on a device. The data can contain sensitive information, such as billing and payment records, usage statistics, business secrets and decryption keys. Intentional or unintentional alteration or deletion of such data can be traumatic, hence the need for information integrity protection.

In the context of IoT, integrity is maintaining the accuracy, consistency and trustworthiness of data in transit or stored on any IoT device. There should be assurance that information cannot in anyway be modified by unauthorized entities.

2.2.3 Availability

Availability in the IoT ensures that a system or a service is available to authorized users, and that authorized users have access to every data they are authorized to access. Thus the connectivity of an IoT device or service must persist even if there is a link failure, which necessitates the need for link handover whenever a link fails.

The best way to ensure availability is by performing hardware repairs as soon as a problem occurs, carrying out a regular preventive maintenance of all hardware and ensuring that the Operating System (OS) is free of any software conflicts and is up to date. In addition, other software on the system must remain updated. It is also important to maintain an immediate and adaptive recovery in worst case scenarios. Backup copies of data should be stored in safe locations to safeguard against data loss in case of a natural disaster, such as fire or flood.

2.2.4 Authentication

Authentication is a property that ensures that a transaction or any other exchange of information is from the source it claims to be from. This implies that all IoT devices must be able to prove their identity in order to maintain authenticity. Device authenticity can be verified through authentication, which involves proof of identity. This happens whenever a device is connected to a network. Device authentication enables a device to access a network based on identical credentials stored in a safe location. The device authenticates itself prior to receiving or transmitting any information. In most cases the process of authentication involves more than one proof of identity.

Parts authentication is also an essential requirement that should be considered in IoT system design. It will ensure that no third party components with potential security risks are connected to the system. Additionally, it has the potential to allow secure in-service upgrades as a result of the code-signing confirmation that authenticates the identity of every firmware source.

2.2.5 Access Control

Access control refers to the security attribute that allows only authorized entities to access a resource, such as sensor data, file or website. In the context of IoT, access control enables secure interaction between devices. It determines who is allowed to access some resources, and limits the privileges of device Applications (apps) and components so that they can only access resources necessary for their normal operations. Access control is needed in an IoT system to ensure that only trusted entities can update device software, command actuators, perform device configuration operations or access sensor data. An efficient access control also enables the creation of new business services that allow customers to access some information like sensor data after payment.

Authentication is an essential ingredient for access control, since in order to control access both users and devices must be identified. IoT, however, poses a distinctive set of challenges due to highly constrained computational power, memory and low power requirement of many IoT devices along with the nature of deployment of some devices. As such, some standard authorization models like Access Control List (ACL) and Role Based Access Control (RBAC) may not apply directly [17].

2.2.6 Non-repudiation

In the IoT context, non-repudiation is a security property that provides available proofs that will prevent any user or device from denying an action, such as message exchange. It ensures the availability of evidence, usually through a Trusted Third Party (TTP). The evidence should make the transfer of credentials between entities

undeniable. It is also possible to use the process of data monitoring in non-repudiation to identify potentially compromised *things*.

Non-repudiation is usually not considered as an important security requirement for many IoT application scenarios. But for business applications that involve payment for services [18], non-repudiation is an indispensable security property that will prevent users and service providers from denying payment action. The greatest bottleneck, however, for implementing non-repudiation in some IoT devices is the challenge that using attestation on resource-constrained devices poses.

2.2.7 Secure Booting

As one of the foundations for security in a device, secure booting blocks any attempt to run different software on an IoT device when switching it on. It is a technique that asserts and verifies the integrity of an executable image before control is passed to it. This is the security property which ensures that device firmware has not been tampered with. As soon as power is introduced to an IoT device, the integrity and authenticity of the software running on the device is verified using digital signatures that are generated cryptographically. The digital signature on the software image that is verified by the device ensures that only authorized software that has been signed by the entity that authorized it is allowed to run on the device.

Secure booting requires specific hardware capabilities, since it is implemented using cryptographically signed code provided by the manufacturer along with the hardware support that will verify the authenticity of the code. This further highlights the need for baking security in the device itself [19].

2.2.8 Device Tampering Detection

Device tampering detection is a security requirement that ensures that any attempt to tamper with an IoT device, whether logically or physically, is detected [20]. Although some new Micro-Controller Units (MCUs) have some advanced memory and code protection capabilities that protects against unauthorized access, the use of these tamper-resistant protections may not always provide the required protection, or may not be available.

A large number of IoT devices like sensors are deployed in open environments, allowing attackers to have direct contact with them. In addition, some skilled attackers can even take them to their lab for analysis. Example of IoT devices that are likely to be targets for hardware tampering include sensor nodes and IoT wearable devices.

2.3 *Where Internet of Things Security Becomes Delicate*

While security considerations are not new in IT networks, IoT poses unique security challenges for both individual users and companies. Given the potential pervasive nature of the IoT devices and services, it will not be inappropriate to say that computers have now spread from our desktops to almost every aspect of our lives. They are now found as tiny devices in our pockets, on our wrists, implanted in the body of some animals and humans, and embedded in cars and almost all the everyday gadgets we use [21]. While we may not think of some of these small devices as computers, they can collect, process, store and share vast amounts of data.

In contrast to the paradigm of traditional computer systems that are usually protected inside the secure perimeter of enterprise networks, most devices comprising the IoT are poorly secured, some are poorly designed, a large proportion of them are deployed outside of the standard enterprise security perimeter and quite a number of them use lightweight specialized OSes like VxWorks, MQX and INTEGRITY [20]. As a result, standard computer security solutions may not even run on most of these devices, which makes the task of securing IoT devices and services tricky. The next section provides a discussion on the challenges of implementing security for IoT and some related hardware issues.

2.3.1 **Challenges of Implementing Security for IoT**

The following lists some unique security challenges and design considerations for the IoT devices and systems, which differ considerably from the traditional computers and computing devices [22–24]:

1. Some manufacturers of IoT devices that are new in the business lack security expertise, and can not afford to hire the services of security experts. Therefore they rely on the rudimentary security mechanism on the hardware and software components they acquire. Consequently, they end up producing devices with many security vulnerabilities.
2. Many companies that produce IoT devices spend less or nothing on research and development that can potentially improve the security of their products in quest of competition for inexpensive devices.
3. Virtually all devices that can be connected to the Internet have embedded OSes in their firmware. However, in view of the fact that such devices are designed to be very small and inexpensive, their OSes are usually not designed with security in mind. As a result, most of them have vulnerabilities.
4. The homogeneous nature of IoT deployments, for example, in WSNs in which almost all sensor nodes, apart from the sink node, are very identical represents a potential security risk. Because if attackers can identify some vulnerability on a single device, they can use it to compromised the remaining devices and even others that are similar in design or use the same protocols.

5. As the number of connected devices increases, the techniques used in exploiting potential entry points or vulnerabilities also increase. Now one does not have to be a professional hacker to be able to hack some IoT devices because of the availability of tools and tricks on-line coupled with simplicity in design of some of the devices. Cyber criminals can reprogram poorly designed devices and cause them to malfunction in order to steal sensitive information.
6. A vast number of IoT devices deployed in difficult terrains are expected to be placed there unattended for years, and due to the nature of the terrains, it may be difficult to perform some upgrade or configuration on them. For some applications that involve a very large number of devices, IoT devices are designed without provision for upgrade or update, probably due to the complications that will be involved because of the large number. On the other hand, there are some upgradeable *things* that are only replaced every few years, such as smart refrigerators and smart cars, and may have long life cycle. Some of these *things* may even outlive the companies, thereby leaving them without further support. It is obvious that the security mechanisms on the devices in the above scenarios will be outdated, thereby raising serious security concerns.
7. A number of applications may require the deployment of IoT devices in locations where it will be very difficult to provide physical security. In such instances, malicious entities may physically capture some devices in order to reverse engineer them, and possibly access sensitive data.
8. IoT is designed to provide seamless connections among diverse devices in different systems and subsystems using the Internet. As such, a compromised washing machine in a given country can be used to send thousands of risky spam emails across the globe using its Wi-Fi connection.

2.3.2 Hardware Issues

As the IoT technology grows, attention is mostly focused on applications, such as sensing, wireless transmission, smartness and other aspects of the IoT, and forgetting about the underlying hardware that enables such functionalities [25]. In recent years, there are significant technological advancements in hardware manufacturing processes, such as miniaturization of chips, which has inherent advantages, including, but not limited to, smaller size, lower cost and higher speed. Investment in this industry is increasing considerably, and hardware giants like Intel and ARM along with other companies are making significant improvements in their hardware. Nevertheless, in the race for smaller, more energy efficient, lighter and lower cost IoT hardware, the hardware community is facing a number of challenges.

First and foremost, in order for the miniature chips to consume less battery power, outdated architecture is used on at least the first-generation of Intel Edison platform, which is based on Quark processors. Fortunately, the processor speed is improving, because the next-generation of Edison that followed is based on Atom Silvermont cores that is on some Android and Windows tablets [26]. This processor

is significantly faster. Presently, the Edison platform for IoT applications has a *Tangier* System-on-a-Chip (SoC) that mixes a dual-core Atom running Linux with a Quark chip [27]. Eventually, the modern 64-bit x86 CPU cores may end up being used in the next generation of Edison microcomputers for IoT applications. But if the modern 64-bit x86 CPU cores are used in wearable devices, it is not likely that they will be cheap again, and considering their complexity, their power requirement will definitely be more than what a disposable IoT device can withstand.

Another issue is that improving the processing power of the microcomputers will make them to dissipate more heat, which will result in bigger packaging, and hence bigger size. Additionally, processors with hardware-assisted security will consume more power, which implies bigger and more expensive batteries [26].

2.4 Size and Heterogeneity of Things

IoT is characterized by large number of heterogeneous devices. It is expected that this heterogeneity will allow seamless connection of combinations of smart *things* via highly constrained and non-constrained network environments. Connected devices range from very simple and lightweight devices powered by 8-bit MCUs to very sophisticated devices with powerful processing capabilities and extremely large memories. Furthermore, in order to make the IoT vision more realizable, it is expected that in the coming years there will be a growing need for more services that will interconnect multiple IoT application domains. Be that as it may, the sheer size and the growing heterogeneity in terms of device types, device resources, topology and security/communication protocols constitute a challenge to security and privacy in the IoT. The following sections elaborate on the size and heterogeneity of the IoT with more details.

2.4.1 Size of IoT

Considering that the market demand for IoT is expanding daily, the size of IoT can be described in different perspectives, such as IoT market size and IoT revenue size. For example, as IoT is fast becoming a powerful force that is transforming businesses across all industries, it is expected to generate incremental revenue that is estimated to be in billions of dollars. Nonetheless, for the purpose of this chapter, we focus only on the size of IoT in terms of number of connected devices and its impact on security. As the IoT matures, the number of connected *things* continues to grow and applications with extremely large number of devices are becoming commonplace. This number represents a security risk [22], especially if there is a security breach in one or more of the devices. Updating a large number of devices that are already deployed (which may be in millions) will be a very big challenge for manufacturers, and if an attacker successfully exploits vulnerability in a single

device he can compromise other devices on the same network, or extend the effects to other networks, until he eventually reaches his final target.

Moreover, the large number of interconnections of devices in some applications, such as WSNs, which is far more than the number of interconnections in the traditional Internet, constitutes a security concern. Considering the massive number of wireless links between devices in the IoT, malicious entities can exploit any available vulnerability and try to compromise a network.

2.4.2 Heterogeneity of the IoT

The IoT encompasses heterogeneous networks of intelligent devices with diverse network protocols and communication technologies. In the concept of IoT, these diverse *things* are expected to exchange information and data seamlessly without human intervention [28], but most smart *things* use network solutions that are proprietary to specific vendors, leading to undesirable co-location of networking technologies. Furthermore, many intelligent consumer items do not communicate with devices that use only propriety network solutions [29]. Consequently, they have to communicate via gateways, resulting in lossy connections. In addition, since the Industrial, Scientific and Medical (ISM) bands in urban environments are often congested, interference from other wireless networks may hamper the performance of IoT devices, especially where large number of devices are deployed. In the above scenarios, if, for example, there is an attempt to compromise a smart device equipped with device tampering detection mechanism, and that device raised an alarm, the alarm may not be communicated to the relevant authorities due to poor network connection. Hence the device may eventually be compromised. As such, enabling across-the-board network protocols and communication technologies is essential. This highlights the need for seamless interoperability between IoT systems [30].

Considering the level of mobility in the IoT and the fact that most of the connections are wireless, every single hand-off represents a tremendous opportunity for attackers to infiltrate IoT systems. The risk of such security breach can be exacerbated by inadequate interoperability among IoT systems and subsystems. This emphasizes the need for sustainable interoperability to be adopted across a wide range of application domains, which will provide common interface solutions. However, creating a framework to achieve the requisite degree of interoperability is not an easy task.

2.5 Privacy Concerns in the Internet of Things

Since the inception of the World Wide Web in 1989 [31], preservation of privacy has been a concern. Concerns about privacy on the Internet are exacerbated by the coming of IoT, and are expected to grow even more than was previously thought as

the IoT is beginning to take shape and gain popularity with new application domains being created daily. This can be attributed to the fact that securing IoT devices and systems presents additional challenges to security administrators than securing the traditional computers and related systems. For example, a large number of connected devices, such as sensors and smart *things* will be deployed all over the place. Such devices may operate autonomously or be controlled remotely from somewhere. When these devices interact with each other and with humans they will collect, process and transmit data about the environment, objects and of course, about humans [32].

Considering what happens with smart phone technology that sometime captures information without the consent or knowledge of the user [33], there is every reason for users of IoT devices to worry about their privacy. For instance, in some applications like smart healthcare, smart devices leave traceable signatures of behaviours and locations of users that some malicious attackers can exploit.

It is very exciting to see IoT being integrated into different aspects of our live, since it has the potential to bring positive changes to the way we do things. This, however, represents a significant privacy risk that can enable an attacker to carry out an unwanted surveillance on users. Governments can also use IoT devices to carry out unnecessary surveillance on their citizens in the name of counter-terrorism, also known as the *big brother effect* [34, 35]. Without adequate security measures, today it is possible for attackers to hack and take control of any device that has audio or visual capability and use it for malicious surveillance purposes. For example, attackers were able to hack baby monitors in April 2015 [36]. The reality is that even smart TVs can be used by malicious attackers to spy on users [37].

Additionally, as the IoT gains more momentum and acceptance, many companies and businesses have so far collected huge amounts of data about their customers and visitors. This information is being collected from web cookies, social networks, mobile apps, video cameras, RFID bracelets, etc. The stored data may contain a considerable amount of personal and sensitive information. The disturbing reality is that, in most cases, customers are not given the choice to opt-out of data collection. While the motive behind the data collection may be to improve services and customer experience, as well as enable companies identify new business opportunities based on the data, the use of such personal data may amount to infringing on the privacy of users.

3 Fundamental Internet of Things Security Issues

In this section, we examine the fundamental IoT security issues. Essentially, we consider some root causes of IoT security issues, such as *things* are not designed with security in mind; open debugging interfaces; inappropriate network configuration and use of default passwords by users; and lack of encryption of critical information before storage. We also look at current and emerging IoT cyber threat landscape and finally, overview of the IoT threat agents.

3.1 Things Are not Designed with Security in Mind

With the advent of IoT and an ever more connected public, more companies are embedding computers and sensors into their products and enabling them to connect to the Internet. Computers and sensors are now being embedded into all sorts of consumer devices, including tea kettles and clothing. Many houses, offices and factories have a number of computers and sensors embedded all over. While companies are desperately competing and rushing to be the first to launch a particular product into the emerging IoT market, they sometimes forget device security. In the quest to react to various business opportunities, many manufacturers leave a lot of devices open to vulnerabilities. Today, IoT has introduced new exploitable attack surfaces that expand beyond the web into the cloud, diverse OSes, different protocols and many more. Consequently, every day we hear news about new devices that are being compromised by hackers as a result of security vulnerabilities.

The story of data breaches in the IoT devices or networks is becoming commonplace, and if the situation is not carefully handled, something similar to what happened in the mid 1990s (i.e., when the level of insecurity in personal computers reached an alarming stage) [38] may eventually happen with the IoT. This can be attributed greatly to the fact that many vendors do not make security their top priority; they only consider it as an afterthought when their products have security issues. This makes a number of IoT devices and networks vulnerable to all sorts of attacks. As such, many IoT devices can be hacked directly over the network or indirectly using some apps.

3.1.1 Testing the Security of IoT Devices

Several organizations and security firms, including but not limited to Hewlett Packard (HP) [39], Veracode [40] and Symantec [41] have conducted research on security in the IoT. The results of these studies revealed some disturbing facts about the security and privacy status in the IoT. For example, HP carried out the study on 10 of the most popular IoT devices in use. Highlights of the results contained in an official report released by the company in 2015 are presented below:

- Six out of ten of the devices with user interfaces were found vulnerable to some issues like persistent XSS and weak credentials.
- 80 % of the devices captured one or more user information directly or using device mobile apps, or the cloud.
- 90 % of the devices have used unencrypted network service.
- 70 % of devices along with their cloud and mobile apps enable attackers to identify valid user accounts via account enumeration.
- 70 % of devices along with their cloud and mobile apps components failed to require complex and lengthy passwords.

To emphasize the fact that IoT devices, their mobile apps and associated cloud services are mostly designed without security in mind, we now consider the results of the study conducted by Veracode. Six new always-on common IoT devices with up-to-date firmware were examined, which include the following [42]:

1. Chamberlain MyQ Internet Gateway—an Internet-based remote control for garage doors.
2. Chamberlain MyQ Garage—an Internet-based remote control for controlling garage doors, interior switches and electrical outlets.
3. SmartThings Hub—a central control device for controlling home automaton sensors, switches and door locks.
4. Unified Computer Intelligence Corporation (Ubi)—an always-on voice-controlled device for answering questions, controlling home automaton and performing tasks such as sending emails and SMS messages.
5. Wink Hub—a device used as a central control for home automation products.
6. Wink Relay—a combination hub and control device for home automation sensors and products.

Within the scope of their study, the Veracode team of researchers uncovered security vulnerabilities in these devices that could negatively impact user experience. The results of the study are summarized below:

- Cyber criminals can leverage data obtained from the Ubi device to know exactly when a user is at home based on the level of noise or brightness of light in the room, which could facilitate burglary.
- Security vulnerabilities in Wink Relay or Ubi can enable malicious attackers to turn on microphones on IoT devices with such capabilities and eavesdrop on users.
- Vulnerabilities in Chamberlain MyQ system can allow an attacker to know when a garage door is closed or opened. Such information could be used to rob the house.
- Some of the vulnerabilities provide attackers with the opportunity to remotely run arbitrary code on a device.

To scrutinize the security of IoT devices, a Symantec team of researchers examined 50 smart home devices that are commonly used today. The team discovered that: none of the tested devices allow the use of strong passwords, none used mutual authentication and in addition, user accounts on those devices are not protected against brute-force attacks. They also found that about 2 out of 10 of the mobile apps for controlling those devices did not use Secure Socket Layer (SSL) for encrypting information exchange between the devices and the cloud.

The team further highlighted a number of attacks surfaces of smart home devices, including physical access, local attacks over Wi-Fi/Ethernet, cloud polling, direct connection, cloud infrastructure attacks and malware. They went further to describe several possible attack scenarios based on each attack surface.

3.2 *Open Debugging Interfaces*

The importance of building security into IoT devices at the outset, rather than considering it as an afterthought was mentioned earlier. One of the important approaches to implementing security by design is to make sure that the attack surface is minimized as much as possible [43]. As such, there is need for manufacturers of IoT gateways to implement only the necessary interfaces and protocols that will enable an IoT device to perform its intended functionalities. Manufacturers should put a limit on the services of all interfaces on the device for debugging purposes, which, in most cases, may not be even needed by the user, and can allow hackers to have direct channel into the local area network using the Wi-Fi.

Although leaving such interfaces may be indispensable for the manufacturer and researchers for development and testing purposes, a user may not even know that such interfaces exist throughout the lifespan of the device, hence he does not need them. In addition, these open debugging interfaces are potentially dangerous and present opportunities for malicious entities to hack the device or access important information. Moreover, through them, malicious attackers can remotely run some harmful code, such as virus and spyware on the device [43].

For instance, in the research study conducted by Verocode [40], reported earlier, the research team discovered that some debugging interfaces were left open and unsecured on two of the devices they examined, through which a malicious attacker can run arbitrary code and harm the system. Even though some of these interfaces may be hidden, a malicious person can go to any length to discover them. Once discovered, all one needs to do is to go to Android developer site and download a simple toolkit that can enable him have an access into the gateway within few minutes.

In order to implement a workable security in the IoT, the concept of security by design should be given the priority it deserves so that unnecessary security flaws can be avoided. For example, it is important for manufacturers to include some mechanism in their design that will prevent even a legitimate user from running malicious code that can be dangerous to the gateway device or a smart device.

3.3 *Inappropriate Network Configuration and Use of Default Passwords by Users*

With the continued rise in production of diversity of IoT devices, more smart *things* hit the market by the day, and now the connected versions of almost every household appliance is available in the market. Additionally, as manufacturers are seriously competing to grab their share of the predicted \$1.7 trillion revenues in the IoT market by 2019 [44], the prices of these *things* are getting cheaper by the day. For these reasons, people are now using connected devices more than ever before. A recent survey carried out by TRUSTe [45] showed that 35 % of U.S. and 41 % of

British domestic online consumers own at least one smart *thing* apart from their phone. The survey further revealed the most popular devices in use, which include: smart TVs (20 %), in-car navigation systems (12 %), followed by fitness bands (5 %) and home alarm systems (4 %). Even though percentage and popularity of devices may differ, the results of this survey are likely to be true for other European Countries, China and many other developing Countries.

Unfortunately, it seems that many consumers are not aware of IoT security concerns and their associated implications on their privacy and security. This is evident from the manner in which some consumers install, configure and use their smart devices. A significant portion of the security concerns revolves around appliances for smart homes. The *do-it-yourself* syndrome, where some consumers use default passwords and settings on their smart devices, as in the case of the security cameras reported in [46–48], has resulted in inappropriate network design and implementation and consequently leaving their Internet routers and smart devices open for hackers to access [49]. This is one of the reasons why many smart home internal networks are badly configured [50].

Another issue is the use of weak passwords. In most cases when some users are smart enough to change default passwords, they, however, use simple passwords that are easy to guess. Sometimes, a user that is enlighten about security may want to use a long and complex password that would be difficult for an attacker to guess, but the restrictions on some devices may not allow such setting. In addition, many of the devices do not have keyboards, and since all configurations must be done remotely, some users become reluctant about security settings.

It is possible that some users are unaware that attackers usually look for poorly configured networks and devices to exploit. Therefore, it is important for vendors to find a way of educating consumers on current security best practice, which include changing passwords regularly and proper network configuration.

3.4 Lack of Encryption of Critical Information Before Storage

It is an obvious known fact that many IoT devices, with or without the consent of their owners, do collect some kind of personal information. Such information may include name, date of birth, address, zip code, email address, location, health information, social security number, and in some cases even credit card number. A leading global data privacy management company, TRUSTe [51], conducted an online survey in the U.S. on 2,000 Internet users aged between 18 and 75, and found that 59 % of users are aware that smart *things* can capture sensitive information about their personal activities. 22 % believed that the benefits and conveniences that come with the IoT innovation are worth sacrificing their privacy for and, surprisingly, 14 % were comfortable with such companies collecting their personal information. The question now is not, if these devices are really collecting

personal data from users, but, as rightly posed by HP [39]: “Do these devices really need to collect this personal information to function properly?” One of the obvious reasons that most companies would give for capturing personal user data is that they need such data in order to improve their products. Another reason could be to know the habits of their valuable customers so they can better serve them by creating new services that would meet their needs as individuals.

Now that it is established that some IoT devices collect sensitive personal information from their users, limiting the amount of data collected by these devices is crucial. Of course, this is a difficult thing to do because of the business opportunities such data collections have created for these companies [52, 53]. However, data stored on IoT devices constitutes a tempting target for attackers, either due to the value of information it refers to, or because of the simplicity involved in accessing it. As these devices store data locally on their memories or transmit it over a variety of networks, such as home and hospital networks (which on many occasions are insecure) [54], there are concerns that the data could be accessed or manipulated over the Internet. Hence, the most appropriate thing to do irrespective of the amount of data that a device collects is to ensure that this data is well protected, whether stored on the device internal memory or on transit. So far, encryption is the best way to protect data against unauthorized access [55, 56], and hence protecting its integrity and confidentiality. Unfortunately, many IoT devices store data on their memories in an unencrypted form, making it easy for hackers to lay hands on it [14].

While encryption is widely believed to be the best approach to securing data, implementing it on many IoT devices presents some unique challenges for security experts. For example, the use of SSL for securing communication in some IoT devices is not an option, since it requires more processing power and memory, which are very scarce resources on the resource-constrained devices. One other option is to think of using Virtual Private Network (VPN) tunnel, which requires a fully featured OSes. However, smart devices run very light versions of OSes [57].

3.5 Current and Emerging IoT Cyber Threat Landscape

As IoT enters deep into our daily lives, and more *things* are getting connected and tapping into the Internet than people, security risks associated with the IoT continue to grow in both number and sophistication. Today, an increasingly massive amount of personal information and business data captured by numerous connected *things* and sensors (that may have exploitable vulnerabilities) are being transferred among smart *things* and between *things* and the cloud, thereby expanding the threat landscape. On top of that, a single weak link on any of such devices is enough to provide hackers with unlimited doorways through which they can gain unauthorized and unlawful access to an IoT system.

In the context of computer security, a threat is a potential to exploit a vulnerability or a potential to violate a security policy, and hence expose a computer

system to harm [58]. Thus, a threat may or may not happen, but if it happens, it is capable of inflicting damage on a system. The Proliferation of highly vulnerable smart devices in the enterprise, homes and in everyday life is creating an unprecedented increase in the number and sophistication of cyber threats in the IoT. Since every *thing* that connects with the Internet creates potential entry point for cyber criminals, and considering the large number of smart devices involved in the connections, effective cyber security in general is becoming increasingly complex to provide.

Given the complex protection model of resource-constrained devices of the IoT, point solutions like security suite or antivirus software that can be easily installed on laptops, smart phones and tablets cannot be used to protect these devices against threats. They lack advanced OSES that can handle the requirements of such functionalities. Furthermore, in view of the diversity of threats, delivering perimeter security in the context of IoT will require much skills and efforts. Hence, understanding threats is crucial, as it often allows security administrators to provide proper countermeasures against these threats. The most common threats that are known across different ecosystems include, but are not limited to, DoS, MitM, attacks on privacy and physical attacks. The effects of threats vary considerably depending on their nature. For instance, some may affect the confidentiality, integrity or availability of data, while others may damage a system completely. To fully understand the growing trend of cyber threats in the IoT, we need to identify the valuable assets and the diverse vulnerabilities in the IoT.

3.5.1 Internet of Things Cyber Assets

In the context of the IoT, an asset can be defined as any tangible or intangible item that is considered valuable to a person or an enterprise, which is also subject to threats. Assets in any IoT ecosystem can be hardware, software, or a piece of data (belonging to individuals or cooperate organizations), services, or the data therein [59]. As IoT is revolutionizing many industries, the manufacturing industry is also changing. Currently, cyber assets for industrial IoT do not only refer to computing devices, but include different machines, such as boilers, conveyors, compressors and some other equipment [60]. These factory devices may not necessarily be in the same geographical location, but are being connected via the Internet in order to optimize performance and efficiency. For a company, assets may also refer to things that are related to the business, but are not really electronic or infrastructure embodiment, such as the reputation of the company [61]. In Fig. 1 we provide an overview of typical IoT assets for five different application domains.

3.5.2 Vulnerabilities Associated with IoT Devices

Vulnerabilities represent weaknesses or mistakes in a device or system that allow an unauthorized entity to locally or remotely execute commands, access or modify unauthorized data, interrupt normal operation of a system, and/or damage a system

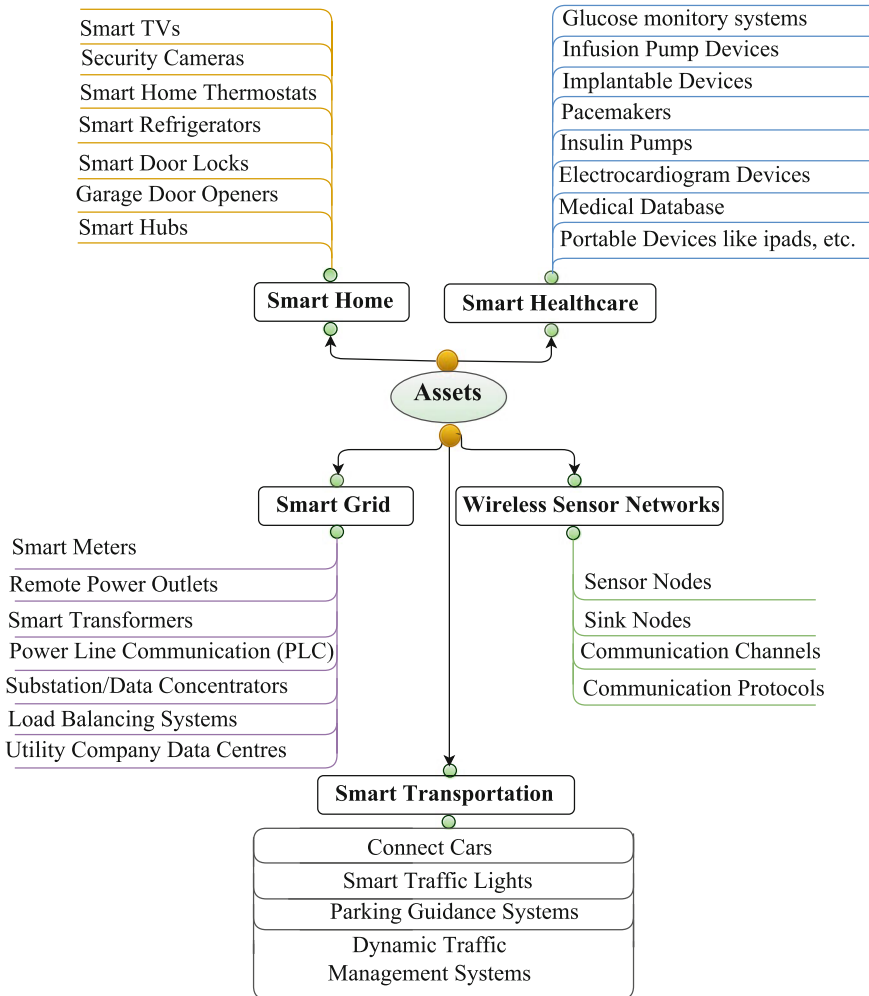


Fig. 1 Overview of typical IoT assets for five application domains

[61]. In computer security, vulnerability is popularly defined as “the intersection of three elements: a system susceptibility or flaw, attacker access to the flaw, and attacker capability to exploit the flaw” [62, 63]. Security flaws can be found on any or both of the two major components of an IoT system, namely software and hardware. Usually, hardware vulnerabilities are not easy to figure out, and fixing such vulnerabilities is typically much harder. The hardware of most IoT devices have a considerable number of embedded micro programs in them, and fixing vulnerabilities in these micro programs is no trivial task for different reasons, which include cost, lack of expertise, incompatibility and interoperability with bigger hardware, and it requires a lot of effort and time. Software vulnerabilities can exist

in the device OSes, communication protocols and other apps software. A number of factors are responsible for software vulnerabilities, which include design flaws and software complexity [59].

The Open Web Application Security Project (OWASP), under the IoT Top 10 project has highlighted top ten security vulnerabilities that are associated with many IoT devices [64]. The top ten security vulnerabilities presented below serve as guide for manufactures to take into consideration to secure IoT devices:

1. Insecure web interface
2. Insufficient authentication/authorization
3. Insecure network services
4. Lack of transport encryption
5. Privacy concerns
6. Insecure cloud interface
7. Insecure mobile interface
8. Insufficient security configurability
9. Insecure software/firmware
10. Poor physical security.

3.6 Overview of Internet of Things Threat Agents

Threats are normally manifested through threat agents (which can be individuals or groups), also known as threat actors, who maliciously break through systems using a variety of techniques, causing different levels of damages. There exist multitudes of threat actors with diverse motives targeting the IoT, including business competitors seeking advantage, criminals seeking financial gain, foreign governments perpetrating espionage for economic or military purpose, just to mention a few [59]. Over the years these potential adversaries have proven remarkably innovative, resourceful and very determined in exploiting vulnerabilities [65]. The OWASP IoT project, under the threat agent category [66], has formulated a mathematical expression that describes threat agent as:

$$\text{Threat Agent} = \text{Capabilities} + \text{Intentions} + \text{Past Activities} \quad (1)$$

The expertise of threat agents pose a greater challenge than the monitoring of malicious code scheme, and much bigger challenge than monitoring changes to system configuration using intrusion detection techniques. Therefore, identifying these entities who can potentially exploit the assets of individuals or companies in the IoT is very fundamental. The OWASP [66] IoT project has also broadly classified threat agents, which we have summarize in Table 1 below. Seven threat agents are shown in Table 1 with each having its class and typical examples.

Table 1 Classification of IoT threat agents

Threat agent	Class	Typical examples
Non-targeted specific	Software	Computer viruses, worms, trojans, logic bombs
Employees	Insider	Disgruntled staff, contractors, security guards
Organized crime and criminals	Outsider	Criminals that target valuable information, such as credit card numbers and bank accounts
Corporations	Outsider	Corporations, government agencies, partners, competitors
Human	Unintentional	Accidents, carelessness
Human	Intentional	Insider, outsider
Natural disasters	Non-human factors	Flood, fire, lightning, meteor, earthquakes

4 IoT Security Issues that Need Immediate Attention

In this section we highlight IoT security issues that need immediate attention. Particularly, we discuss efficient lightweight authentication schemes for IoT, robust and flexible lightweight cryptography, need for efficient key revocation schemes and need for standardization of security solutions.

4.1 Efficient Lightweight Authentication Schemes for IoT

As the IoT leads us to the Internet of Everything (IoE) with people, *things*, data and processes as its core components, authentication is among the most critical functionalities that will enable secure communication between these entities. Authentication, in the context of the IoT, simply refers to the process of identifying and validating users and connected *things* like smart devices, computers and machines. It allows authorized users or devices to access resources as well as denies malicious entities access to such resources [67]. It can also restrict authorized users or devices from accessing compromised devices. Furthermore, authentication reduces the chances for an intruder to establish multiple connections with the gateway, thereby reducing the risks of DoS attacks.

In a secure IoT communication, prior to any communication between two or more entities that will involve accessing a resource, each participating entity must be validated and authenticated so as to establish its real identity in the network. It implies that each legitimate node or entity must have a valid identity in order to participate in the communication. In spite of the importance of secure identity in IoT communications, however, many of the IoT devices in the market today lack security identities and have fallen victim to a number of security violations [68].

An authentication process typically relies on the use of usernames and passwords. For example, on the traditional Internet, websites authenticate users by

requiring usernames and passwords, and browsers authenticate websites using SSL protocol. But one contending issue in the IoT is that the devices usually deployed at the core of the communication system and the terminal nodes in most ecosystems are made up of sensors, and in some cases RFID tags. These terminal devices are used for gathering information and transmitting the gathered information to the various platforms. Consequently, in absence of identity validation and authentication, an adversary can connect to these sensors and also access the data, or carry out a wide range of malicious activities. Considering that most of them are energy-starved nodes and have limited computation and memory resources [69], traditional secure authentication schemes, most of which are based on public key cryptography that need a lot of computation and memory space [70], cannot be used directly on them. Hence the need for secure and efficient lightweight authentication schemes for IoT ecosystems cannot be overemphasized.

4.2 *Robust and Flexible Lightweight Cryptography*

Lightweight Cryptography (LWC) is an emerging field for developing cryptographic algorithms or protocols for implementation in constrained environments, such as WSNs, RFID tags, smart health-care devices, and embedded systems among many others [71]. LWC is expected to play a vital role in securing the IoT and ubiquitous computing in general [72]. The term *lightweight* can be considered from two perspectives, namely hardware and software. However, lightwightness in hardware does not necessarily imply lightwightness in software and vice versa. Besides, there are even design trade-offs between them [73]. For a more thorough discussion on this subject, we refer the reader to [73, 74].

In the last few years, a number of lightweight ciphers have been developed tailored for small scale embedded security, including KLEIN, PRESENT, XTEA, CLEFIA, Hummingbird 2, just to mention a few. But the reality is that most of these primitives guarantee only a low level of security, which restricts their deployment [75]. Since trade-offs invariably exist between security and performance, balancing the trade-offs between security and efficiency for LWC will continue to be a challenge. Similarly, the power consumption of resource-constrained devices and the issues associated with the hardware weight and the software weight for LWC need to be addressed in order to develop more robust and flexible LWC for IoT applications.

4.3 *Need for Efficient Key Revocation Schemes*

Many resource-constrained devices of the IoT are often deployed in open and hash environments where disruption of connectivity and device failures are not rare phenomena. A good example is WSNs, a key technology for collecting a variety of

data from different environments in the IoT. As these highly constrained devices are deployed in such hostile environments [76], they are susceptible to all manner of attacks. Hence it is important to devise a mechanism that can effectively and efficiently revoke secret-keys (or private-keys) as soon as a sensor node or any other smart device is compromised.

In the Internet, secure communication between two or more entities relies on trust of digital certificates. Clients can present certificates to servers and vice versa. In cryptography, a digital certificate is simply an electronic document that ties up an attribute such as a public-key with an identity. Public Key Infrastructure (PKI) is a system that manages the creation, distribution and revocation of digital certificates and private-keys. Digital certificates and secret-keys have an expiration date. They can also be revoked prior to expiration for a number of reasons, such as compromise of private-keys or change in the status of an entity that holds the key. PKI allows users to revoke a public-key at the Certificate Authority (CA) if the corresponding private-key is compromised. Revoking any certificate associated with a compromised key is very critical so as to mitigate the possible damages that a compromised key can cause. When a certificate is revoked, certificate revocation information must be made available to participating entities through Certificate Revocation List (CRL), On-line Certificate Status Protocol (OCSP), or some other means. For more details on this topic, we refer the reader to [77].

Like in the traditional Internet, trust within the IoT is also a fundamental requirement. There is need for entities to trust that smart devices and their associated data and services are secure from every form of manipulation. However, implementation of key revocation is more challenging in the IoT than in the traditional Internet. There are many reasons for the complications in the implementation, including the size of network, diversity of devices, and constrained nature of many devices. For instance, connecting low cost devices with limited resources makes it difficult for cryptographic algorithms based on Public Key Cryptography (PKC) to function without impacting on the quality or level of security to be provided. Moreover, many smart devices completely ignore CRL, thereby giving opportunity to malicious entities to use keys that were obtained through a data breach to perform malicious activities [41]. Developing efficient and reliable key revocation schemes that will cope with the diverse issues in the IoT is therefore crucial.

4.4 Need for Standardization of Security Solutions

The variety of devices comprising the IoT is stunning; as a result, IoT is crowded with a number of wireless communication technologies like Wi-Fi, Bluetooth, IEEE 802.15.4, Zigbee, Long-Term Evolution (LTE), etc. This mixture of different physical layers makes interoperability among connected devices very difficult. While devices using different communication technologies can still communicate through IP routers, a gateway must be used when the incompatibility issues in the

protocol stack go beyond the physical and link layers [78], which increases the cost of deployment. It also complicates the use of security solutions across devices with diverse wireless communication technologies. As such, standardized solutions that can be used across multiple domains are required.

Recognizing the above challenges, some standardization initiatives for the IoT are already underway. Discussions about the establishment of standards for the IoT began in 2013. By 2014, few of the standards started taking shape, and quite a handful of them have already started certifying a few products on a preliminary basis as at September 2015 [79]. The standards include Thread Group, AllSeen Alliance/AllJoyn, Open Interconnect Consortium/IoTivity, Industrial Internet Consortium, ITU-T SG20, IEEE P2413 and Apple HomeKit [79]. While some efforts have gone into development of some standards, real standards supporting the IoT are yet to be fully operational. As a consequence, the market is left open for manufacturers to compete without specific guiding rules using different approaches. This gives rise to the development of different protocols and platforms, which in turn results in products with so many vulnerabilities.

A major obstacle in the development of standards for IoT is lack of uniform definition for the smart devices. For example, harmonizing a standard for light bulbs and a standard for pacemakers will definitely be an issue. Again, considering that IoT requires many technologies to work, a single standard cannot cover them all. This is completely different from the desktop and laptop computers where a single standard covers the way they work [79]. Furthermore, due to the number of the standardizing bodies, there may be overlap in their functions, or even conflicts in their strategies. Thus, there is need for the various standards to harmonize their work in order to realize standardized security solutions for the IoT.

5 Conclusions

Through the IoT, an increasing number of people and all types of devices like consumer products communicate and collaborate over the Internet via various accessing methods. This is a clear indicator that the world is moving so fast towards ubiquitous connectivity that is already revolutionizing interactions in almost every human endeavour, including transportation, health-care, economics, work, education, entertainment, among many others. IoT also encompasses the M2H and M2M communication technologies that are creating many business opportunities worldwide. Notwithstanding, such big promises, benefits and opportunities usually come with some risks, and for the IoT, the risks are just as important as the benefits. Security and privacy concerns are among the risks likely to hamper its growth. Presently, there are a number of security and privacy challenges that need to be addressed in order for the IoT to reach its full potential and be fully applicable in an efficiently utilized manner.

In this chapter, we have discussed security and privacy challenges of the IoT, and showed that protecting enterprise intellectual property, customer information

and operational infrastructures is needed now more than ever before. We also pointed out some issues that emphasized the need for the IoT to be secured so that we can fully realize its value and benefits. Furthermore, we highlighted some challenges that are standing as obstacles to the realization of secure communications in the IoT. Some of these issues are the cause of the numerous vulnerabilities that were uncovered in so many smart devices lately. Finally, we discussed some issues that need immediate attention if IoT security and privacy challenges must be addressed.

Acknowledgments The authors wish to thank the Centre for Geodesy and Geodynamics, National Space Research and Development Agency, Toro, Bauchi State, Nigeria for supporting this work. This work was partially supported by the UID/EEA/50008/2013 Project.

References

1. J. M. Batalla, G. Mastorakis, C. X. Mavromoustakis, and J. Zurek. On Cohabiting Networking Technologies with Common Wireless Access for Home Automation Systems Purposes. *IEEE Wireless Communication Magazine*, 2016.
2. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. *IEEE Commun. Surveys & Tuts.*, 2015. ISSN 1553-877X. doi:[10.1109/COMST.2015.2444095](https://doi.org/10.1109/COMST.2015.2444095).
3. C. Prasse, A. Nettstraeter, and M. T. Hompel. How IoT will Change the Design and Operation of Logistics Systems. In *IEEE Int Conf IoT*, Oct 2014. doi:[10.1109/IOT.2014.7030115](https://doi.org/10.1109/IOT.2014.7030115).
4. World Economic Forum. Industrial Internet of Things: Unleashing the Potential of Connected Products and Services. Available via World Economic Forum, Jan 2015. URL http://www3.weforum.org/docs/WEFUSA_IndustrialInternet_Report2015.pdf. Cited 12 Dec 2015.
5. C. Pettey. The Internet of Things Is a Revolution Waiting to Happen. Available via Gartner, Inc., April 2015. URL <http://www.gartner.com/smarterwithgartner/the-internet-of-things-is-a-revolution-waiting-to-happen/>. Cited 20 Nov 2015.
6. Gartner. Gartner Says 4.9 Billion Connected “Things” Will Be in Use in 2015. Available via Gartner, Inc., Nov 2014. URL <http://www.gartner.com/newsroom/id/2905717>. Cited 20 Nov 2015.
7. P. Gutierrez. Gartner Predicts 6.4 Billion ‘Things’ in 2016. Available via IoTHUB, Nov 2015. URL <http://www.iothub.com.au/news/gartner-predicts-64-billion-things-in-2016-411686>. Cited 21 Nov 2015.
8. J. M. Batalla, M. Gajewski, W. Latoszek, P. Krawiec, C. X. Mavromoustakis, and G. Mastorakis. ID-based Service Oriented Communications for Unified Access to IoT. *Computers & Electrical Engineering*, pages –, 2016. ISSN 0045-7906. doi:[10.1016/j.compeleceng.2016.02.020](https://doi.org/10.1016/j.compeleceng.2016.02.020).
9. M. A. Al Faruque and K. Vatanparvar. Energy Management-as-a-Service Over Fog Computing Platform. *IEEE Internet of Things Journal*, 3(2):161–169, April 2016. ISSN 2327-4662. doi:[10.1109/JIOT.2015.2471260](https://doi.org/10.1109/JIOT.2015.2471260).
10. Proofpoint. Proofpoint Uncovers IoT Cyberattack. Available via Proof-point, Inc., Jan 2014. URL <http://investors.proofpoint.com/releasedetail.cfm?ReleaseID=819799>. Cited 26 Nov 2015.
11. A. Ukil, J. Sen, and S. Koilakonda. Embedded Security for Internet of Things. In *IEEE 2nd National Conf Emerging Trends & Appls Comput Sci*, March 2011. doi:[10.1109/NCETACS.2011.5751382](https://doi.org/10.1109/NCETACS.2011.5751382).

12. Symantec. Securing IoT Devices and System. Available via Symantec Corporation, 2015. URL <https://www.symantec.com/iot/>. Cited 12 Dec 2015.
13. WIND. Security in the Internet of Things. Available via Wind River Systems, Inc., January 2015. URL http://www.windriver.com/whitepapers/security-in-the-internet-of-things/wr_security-in-the-internet-of-things.pdf. Cited 1 Nov 2015.
14. S. Raza. Lightweight Security Solutions for the Internet of Things. Dissertation, Malardalen University Sweden, Jun. 2013.
15. A.M. Gamundani. An Impact Review on Internet of Things Attacks. In IEEE Int. Conf. Emerging Trends Netws. & Comput. Commun, May 2015. doi:10.1109/ETNCC.2015.7184819.
16. M. Abomhara and G.M. Koiem. Security and Privacy in the Internet of Things: Current Status and Open Issues. In IEEE Int. Conf. Privacy Secur. Mobile Syst., May 2014. doi:10.1109/PRISMS.2014.6970594.
17. S. Sicari, A. Rizzardi, L.A. Grieco, and A. Coen-Porisini. Security, Privacy and Trust in Internet of Things: The Road Ahead. Comput Netws., 2015. ISSN 1389-1286. doi:10.1016/j.connect.2014.11.008.
18. I. Alqassem and D. Svetinovic. A Taxonomy of Security and Privacy Requirements for the Internet of Things (IoT). In IEEE Int. Conf. Ind. Eng. Eng. Manag., Dec 2014. doi:10.1109/IEEM.2014.7058837.
19. T. Xu, J.B. Wendt, and M. Potkonjak. Security of IoT Systems: Design Challenges and Opportunities. In IEEE/ACM Int Conf Comput-Aided Design (ICCAD), Nov 2014. doi:10.1109/ICCAD.2014.7001385.
20. A. Grau. The Internet of Secure Things - What is Really Needed to Secure the Internet of Things? Available via ICON LABS, 2015. URL <http://www.iconlabs.com/prod/internet-secure-things-%E2%80%93-what-really-needed-secure-internet-things>. Cited 16 Dec 2015.
21. The Economist. Embedded Computers: Hacking the Planet. Available via The Economist Newspaper, Jul 2015. URL <http://www.economist.com/news/leaders/21657811-internet-things-coming-now-time-deal-its-security-flaws-hacking>. Cited 27 Nov 2015.
22. J. Dixon. Who Will Step Up To Secure The Internet Of Things? Available via Crunch Network, Oct 2015. URL <http://techcrunch.com/2015/10/02/who-will-step-up-to-secure-the-internet-of-things/>. Cited 4 Dec 2015.
23. K. Rose, Scott Eldridge, and Lyman Chapin. The Internet of Things: An Overview-Understanding the Issues and Challenges of a More Connected World. The Internet Society, pages 1–50, Oct 2015.
24. A. Kumar. Internet of Things (IOT): Seven Enterprise Risks to Consider. Available via TechTarget, 2015. URL <http://searchsecurity.techtarget.com/tip/Internet-of-Things-IOT-Seven-enterprise-risks-to-consider>. Cited 4 Nov 2015.
25. T. Lee. The Hardware Enablers for the Internet of Things - Part I. IEEE Internet of Things Newsletter, Jan 2015.
26. N. Hajdarbegovic. Are We Creating An Insecure Internet of Things (IoT)? Security Challenges and Concerns. Available via Toptal, 2015. URL <http://www.toptal.com/it/are-we-creating-an-insecure-internet-of-things>. Cited 23 Nov 2015.
27. E. Brown. Edison IoT Module Ships with Atom/Quark Combo SoC. Available via LinuxGizmos.com, Sep 2014. URL <http://linuxgizmos.com/edison-iot-module-ships-with-atom-plus-quark-combo-soc/>. Cited 7 Dec 2015.
28. E. Baccelli, O. Hahm, M. Gunes, M. Wahlisch, and T.C. Schmidt. RIOT OS: Towards an OS for the Internet of Things. In IEEE Conf. Comput. Commun. Workshops, April 2013. doi:10.1109/INFCOMW.2013.6970748.
29. E. D. Poorter, I. Moerman, and P. Demeester. Enabling Direct Connectivity Between Heterogeneous Objects in the Internet of Things through a Network-Service-Oriented Architecture. EURASIP J Wireless Commun & Netw., 2011. doi:10.1186/1687-1499-2011-61.

30. V. L. Shivraj, M. A. Rajan, M. Singh, and P. Balamuralidhar. One Time Password Authentication Scheme Based on Elliptic Curves for Internet of Things (IoT). In IEEE 5th National Symp. Info. Technol.: Towards New Smart World, Feb 2015. doi:[10.1109/NSITNSW.2015.7176384](https://doi.org/10.1109/NSITNSW.2015.7176384).
31. W. Coomans, R. B. Moraes, K. Hooghe, and J. Maes. The 5th Generation Broadband Copper Access. In Proceedings of IEEE 9th ITG Symp. Broadband Coverage in Germany, pages 1–5, April 2015.
32. M.J. Covington and R. Carskadden. Threat Implications of the Internet of Things. In M. Maybaum K. Podins, J. Stinissen, editor, IEEE 5th Int. Conf. Cyber Conflict, pages 1–12, June 2013.
33. B. Contos. Security and the Internet of Things - are we Repeating History? Available via CSO, Jul 2015. URL <http://www.csoonline.com/article/2947477/network-security/security-and-the-internet-of-things-are-we-repeating-history.html>. Cited 11 Nov 2015.
34. D. Bradbury. How can Privacy Survive in the Era of the Internet of Things? Available via The Guardian, Apr 2015. URL <http://www.theguardian.com/technology/2015/arp/07/how-can-privacy-survive-the-internet-of-things>. Cited 12 Nov 2015.
35. R. Benest. The Internet of Things: Big Progress or Big Brother? Prospect J Int. Affairs UCSD,, Jun 2015. URL <http://prospectjournal.org/2015/06/04/the-internet-of-things-big-progress-or-big-brother/>. Cited 18 Dec 2015.
36. D. Storm. 2 More Wireless Baby Monitors Hacked: Hackers Remotely Spied on Babies and Parents. Available via ComputerWorld, Apr 2015. URL <http://www.computerworld.com/article/2913356/cybercrime-hacking/2-more-wireless-baby-monitorshacked-hackers-remotely-spied-on-babies-and-parents.html>. Cited 28 Nov 2015.
37. G. Walters. It's Not Just Smart TVs. Your Home is Full of Gadgets that Spy on You: How Internet Giants are Collecting Your Personal Data Through their High-tech Devices. Available via MailOnline, Feb 2015. Security Challenges of the Internet of Things 29 URL <http://www.dailymail.co.uk/sciencetech/article-2950081/It-s-not-just-smart-TVs-home-gadgets-spy-internet-giants-collecting-personal-data-high-tech-devices.html>. Cited 10 Dec 2015.
38. B. Schneider. The Internet of Things is Wildly Insecure - And often Unpatchable. Available via WIRED, Jun 2014. URL <http://www.wired.com/2014/01/theres-no-good-way-to-patch-the-internet-of-things-and-thats-a-huge-problem/>. Cited 13 Dec 2015.
39. Hewlett Packard. Internet of things Research Study. Available via HP Enterprise, 2015. URL <http://www8.hp.com/h20195/V2/GetPDF.aspx/4AA5-4759ENW.pdf>. Cited 9 Dec 2015.
40. Veracode. Veracode Study Reveals the Internet of Things Poses Cybersecurity Risk. Available via VERACODE, Apr 2015. URL <https://www.veracode.com/veracode-study-reveals-internet-things-poses-cybersecurity-risk>. Cited 15 Nov 2015.
41. M. B. Barcena and C. Wueest. Insecurity in the Internet of Things. Available via Symantec, Mar 2015. URL https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/insecurity-in-the-internet-of-things.pdf. Cited 17 Dec 2015.
42. C. Osborne. Internet of Things Devices Lack Fundamental Security, Study Finds. Available via ZDNet, Apr 2015. URL <http://www.zdnet.com/article/internet-of-things-devices-lack-fundamental-security-study-finds/>. Cited 2 Dec 2015.
43. C. Fife. Securing the IoT Gateway. Available via CITRIX, Jul 2015. URL <https://www.citrix.com/blogs/2015/07/24/securing-the-iot-gateway/>. Cited 19 Nov 2015.
44. J. Greenough. The 'Internet of Things' will be the World's most Massive Device Market and Save Companies Billions of Dollars . Available via Business Insider, Apr 2015. URL <http://www.businessinsider.com/how-the-internet-of-things-market-will-grow-2014-10>. Cited 24 Nov 2015.
45. TRUSTe. Majority of Consumers Want to Own the Personal Data Collected from their Smart Devices: Survey. Available via TRUSTe, Jan 2015. URL <http://www.truste.com/blog/2015/01/05/majority-consumers-want-own-personal-data-survey/>. Cited 4 Dec 2015.

46. M. Smith. Peeping into 73,000 Unsecured Security Cameras Thanks to Default Passwords. Available via NetworkWorld, November 2014. URL <http://www.networkworld.com/article/2844283/microsoft-subnet/peeping-into-73-000-unsecured-security-cam-Eras-thanks-to-default-passwords.html>. Cited 7 Nov 2015.
47. D. Bisson. 73,000 Security Cameras Viewable Online Due to Use of Default Passwords. Available via Tripwire, November 2014. URL <http://www.tripwire.com/state-of-security/latest-security-news/73000-security-cameras-viewable-onlin-due-to-use-of-default-passwords/>. Cited 7 Nov 2015.
48. K. Zetter. Popular Surveillance Cameras Open to Hackers, Researcher Says. Available via WIRED, May 2012. URL <http://www.wired.com/2012/05/cctv-hack/>. Cited 7 Nov 2015.
49. M. Kostyukov. Smart Home Security. Available via Home Toys, January 2006. URL <http://www.hometoys.com/content.php?post-type=763>. Cited 27 Nov 2015.
50. J. T. Ho, D. Dearman, and Khai N. Truong. Improving Users' Security Choices on Home Wireless Networks. In Proc. 6th ACM Symp. Usable Privacy Secur., 2010. ISBN 978-1-4503-0264-7. doi:10.1145/1837110.1837126.
51. TRUSTe. 59 % of U.S. Internet Users Know Smart Devices Can Collect Information About Their Personal Activities. Available via TRUSTe, May 2014. URL <http://www.truste.com/events/iot/2014/05/59-of-u-s-internet-users-know-smart-devices-can-collect-information-about-their-personal-activities/>. Cited 8 Dec 2015.
52. M. Rozenfeld. The Value of Privacy: Safeguarding your Information in the Age of the Internet of Everything. The Institute - IEEE News Source, Mar 2014.
53. T. Olavsrud. Internet of Things Connections to Quadruple by 2020. Available via CIO, Mar 2015. URL <http://www.cio.com/article/2899643/data-analytics/internet-of-things-connections-to-quadruple-by-2020.html>. Cited 19 Nov 2015.
54. A. Grau. Hackers Invade Hospital Networks Through Insecure Medical Equipment. IEEE Spectrum, Jun 2015.
55. G. Singh and Supriya. Modified Vigenere Encryption Algorithm and Its Hybrid Implementation with Base64 and AES. In IEEE 2nd Int. Conf. Adv. Comput. Netw. & Secur., Dec 2013. doi:10.1109/ADCONS.2013.33.
56. A. Fragkiadakis, P. Charalampidis, S. Papadakis, and E. Tragos. Experiences with Deploying Compressive Sensing and Matrix Completion Techniques in IoT Devices. In IEEE 19th Int. Workshop Comput. Aided Modeling & Design Commun. Links & Netw., Dec 2014. doi:10.1109/CAMAD.2014.7033237.
57. J. Horn. 5 Security Questions for Your Next IoT Deployment. Available via RacoWireless, Dec 2014. URL <http://embedded-computing.com/guest-blogs/5-security-questions-for-your-next-iot-deployment/#>. Cited 1 Dec 2015.
58. M. Alhabeeb, A. Almuhaideb, P. D. Le, and B. Srinivasan. Information Security Threats Classification Pyramid. In IEEE 24th Int. Conf. Adv. Inf. Netw. Appl. Workshops, Apr 2010. doi:10.1109/WAINA.2010.
59. M. Abomhara and G. M. Kien. Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks. J Cyber Secur., 2014. doi:10.13052/jcsm2245-1439.414.
60. P. Reynolds. The Industrial Internet of Things Will Flip Cyber Security on Its Head. Available via Industrial IoT/Industrie 4.0 Viewpoints, Jun Security Challenges of the Internet of Things 31 2015. URL <http://industrial-iot.com/2015/06/the-industrial-internet-of-things-will-flip-cyber-security-on-its-head/>. Cited 12 Nov 2015.
61. E. Bertino, L. D. Martino, F. Paci, and A. C. Squicciarini. Security for Web Services and Service-Oriented Architectures. Springer, 2010. ISBN 978-3-540-87741-7. doi:10.1007/978-3-540-87742-4.
62. Y. Ilyin. Can we Beat Software Vulnerabilities? Available via Kaspersky Lab, Aug 2014. URL <https://business.kaspersky.com/can-we-beat-software-vulnerabilities/2425/>. Cited 3 Dec 2015.
63. ExpressVPN. What is a Security Hole and How can it Get you Hacked? Available via ExpressVPN, 2015. URL <https://www.expressvpn.com/internet-privacy/guides/what-is-a-security-hole-how-can-it-get-you-hacked/>. Cited 19 Nov 2015.

64. OWASP. OWASP Internet of Things Top Ten Project. Available via OWASP IoT Project, 2014. URL https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project. Cited 5 Dec 2015.
65. S. Pramanik. Threat Motivation. In IEEE 10th Int. Conf. & Expo. Emerging Technol. Smarter World, Oct 2013. doi:10.1109/CEWIT.2013.6851346.
66. OWASP-Threat Agent Category. What is a Threat Agent? Available via OWASP IoT Project, May 2012. URL https://www.owasp.org/index.php/Category:Threat_Agent. Cited 9 Dec 2015.
67. O. O. Bamasag and K. Youcef-Toumi. Towards Continuous Authentication in Internet of Things Based on Secret Sharing Scheme. In Proceedings of the WESS'15: Workshop Embedded Syst. Secur. ACM, 2015. ISBN 978-1-4503-3667-3. doi:10.1145/2818362.2818363.
68. M.A. Jan, P. Nanda, Xiangjian He, Zhiyuan Tan, and Ren Ping Liu. A Robust Authentication Scheme for Observing Resources in the Internet of Things Environment. In IEEE 13th Int. Conf. Trust, Secur. & Privacy Comput. Commun., Sept 2014. doi:10.1109/TrustCom.2014.31.
69. K. Fan, J. Li, H. Li, X. Liang, X. Shen, and Y. Yang. ESLRAS: A Lightweight RFID Authentication Scheme with High Efficiency and Strong Security for Internet of Things. In IEEE 4th Int. Conf. Intell. Netw. Collab. Syst., Sept 2012. doi:10.1109/iNCoS.2012.48.
70. G. Zhao, X. Si, J. Wang, X. Long, and T. Hu. A Novel Mutual Authentication Scheme for Internet of Things. In IEEE Proceedings of Int. Conf. Modeling, Identification & Control, June 2011. doi:10.1109/ICMIC.2011.5973767.
71. G. Bansod, N. Raval, N. Pisharoty, and A. Patil. Modified SIMON and SPECK: Lightweight Hybrid Design for Embedded Security. Cryptology ePrint Archive: Report 2014/1016,, Dec 2014. URL <https://eprint.iacr.org/2014/1016>. Cited 12 Nov 2015.
72. M. Katagi and S. Moriai. Lightweight Cryptography for the Internet of Things. Sony Corporation, pages 1–4, 2011. URL <https://www.iab.org/wp-content/IAB-uploads/2011/03/Kaftan.pdf>. Cited 3 Dec 2015.
73. Universite du Luxembourg. On Lightweightness, Mar 2015. URL https://www.cryptolux.org/index.php/On_Lightweightness. Cited 11 Nov 2015.
74. J. Woods and P. Muoio. Practical Applications of Lightweight Block Ciphers to Secure EtherNet/IP Networks. ODVA Industry Conf & 17th Annual Meeting, pages 1–15, Oct 2015. URL https://www.odva.org/Portals/0/Library/Conference/2015_ODVAConference_Woods_Practical-applications-of-Lightweight-Block-Ciphers.pdf.
75. I. Mansour, G. Chalhoub, and P. Lafourcade. Key Management in Wireless Sensor Networks. J Sensor & Actuator Netw., 2015. doi:10.3390/jsan4030251.
76. M. Ge and K. R. Choo. A Novel Hybrid Key Revocation Scheme for Wireless Sensor Networks. Springer International Publishing Switzerland, 2014. doi:10.1007/978-3-319-11698-3_35.
77. Cisco Systems. Public Key Infrastructure Certificate Revocation List Versus Online Certificate Status Protocol. White Paper, pages 1–6, 2004.
78. S. L. Keoh, S. S. Kumar, and H. Tschofenig. Securing the Internet of Things: A Standardization Perspective. IEEE Internet of Things J, June 2014. ISSN 2327-4662. doi:10.1109/JIOT.2014.2323395.
79. C. Null. The State of IoT Standards: Stand by for the Big Shakeout. Available via TechBeacon, Sep 2015. URL <http://techbeacon.com/state-iot-standards-stand-big-shakeout>. Cited 24 Nov 2015.

Part II
Technologies for Connecting Everything

A Novel Machine to Machine Communication Strategy Using Rateless Coding for the Internet of Things

Boulos Wadiah Khoueiry and M. Reza Soleymani

Abstract Internet of things is a shifting paradigm where almost every physical object will be furnished with sensing, communication and processing capabilities that allow them to communicate with other devices. On the other hand, machine to machine communication, which is a key enabling technology for the internet of things, enables billions of multipurpose networked devices to exchange information among themselves with minor or no human involvement. This chapter aims to investigate a novel communication strategy that considerably increases the efficiency of the channel in the multicast setting. Specifically, we consider the scenario where three devices that are close by and want to exchange their messages via a low cost relay or another device in proximity. The main advantage of the proposed scheme is twofold: (1) use of joint channel and physical layer network coding where devices simultaneously transmit their messages, (2) no decoding at the relay where relaying can be as simple as amplify and forward or de-noise and forward. Furthermore, an efficiently scalable technique for disseminating information among a large number of devices is proposed. Simulation results using practical Raptor codes show that the proposed scheme achieves a near optimal sum rate performance. Additionally, the performance of the proposed scheme is compared with traditional single user communication scheme and functional decode and forward relaying strategy.

1 Introduction

Introduced by Ashton in 1999 [1], the internet of things (IoT) is an emerging and promising subject from social, economic and engineering perspectives. IoT is the global biggest network where consumer electronics, vehicles, wearable micro

B.W. Khoueiry (✉) · M. Reza Soleymani
Wireless and Satellite Communication Laboratory, Department of Electrical
and Computer Engineering, Concordia University, Montreal, QC, Canada
e-mail: b_khouei@ece.concordia.ca

M. Reza Soleymani
e-mail: msoleyma@ece.concordia.ca

devices, sensors and other multipurpose devices with powerful data processing capabilities are being connected to each other or to central servers via the internet or/and telecommunication operators. The advancements in pervasive computing power, electronic miniaturization and networking have made the implementation of IoT closer to reality more than ever.

Researchers have anticipated a potential impact of IoT on the existing internet and telecommunication operators' infrastructures and the economy worldwide over the next few years. While Cisco [2] projects more than 24 billion connected devices to the internet by 2019, Huawei [3] predicts 100 billion IoT connections by 2015. Economically, McKinsey Global Institute [4] forecasts that the financial impact of IoT on the global economy varies in the range of 3.9 to 11.1 trillion Dollars by 2025.

The large scale implementation of IoT affects many practical and promising applications, some of which are: (1) smart grids [5, 6] for intelligent monitoring, control and efficient delivery of electricity to consumers. (2) Smart cities [7] for increasing security, green energy, smart traffic monitoring and intelligent parks. (3) Smart homes [8] for enhancing security and efficient energy consumptions through home automation components. (4) Smart healthcare systems [9] for efficient continuous monitoring of people with disabilities and the elderly, enabling improved level of independence [10]. (5) Intelligent transportation systems [11] for building smarter infrastructure and toward smart logistics systems. (6) Smart tracking and tracing [12]. (7) Public safety systems. Figure 1 illustrates the idea of a large number of wirelessly equipped devices connected together in a smart city.

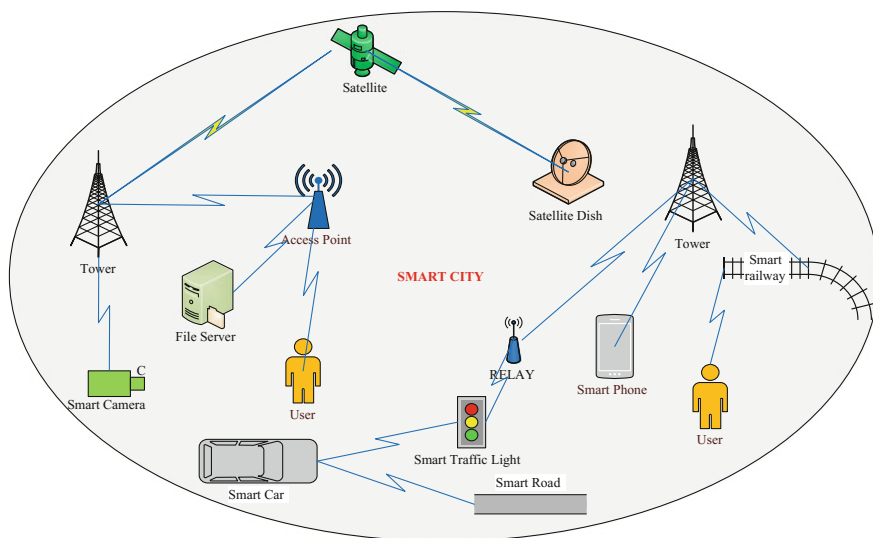


Fig. 1 Smart city supported by MTC devices with interconnected users, surveillance systems, traffic lights, vehicles, roads, railways, sensor devices and servers via telecommunication infrastructures

On the other hand, machine to machine communication (M2M) has emerged as a promising technology enabling billions of multipurpose devices, namely machine-type communication (MTC) devices, to communicate with each other without human intervention [13–15]. MTC refers to existing and future autonomous networked sensing devices which gather and exchange information over the network. Another flavor of M2M communication is the device to device (D2D) communication which similarly refers to establishing direct communication links between devices. In the sequel, D2D or MTC will be used interchangeably.

In this chapter, a bandwidth efficient scheme for a large number of MTC devices that want to exchange their messages is proposed. The process of exchanging messages among a large number of devices requires short communication sessions compared to other types of communicated information over the network. Example of this type of information is localization of devices, reading of the same data from different perspectives, emergency messages, control and monitoring readings and so on.

1.1 Communication Models for the Internet of Things

Earlier this year, the internet architecture board (IAB) released an architectural guide for networking of smart objects (RFC 7452) [16] which outlines the framework of communication models used by IoT devices. The IoT common communication models are categorized into four patterns which will be summarized next.

- A. *Device to Device Communication patterns*: D2D communication models a scenario where two or more devices establish direct communication links between one another to exchange information, rather than going through a centralized controller or an application server. D2D communication uses many existing protocols and infrastructures. D2D is a promising key technology for the next generation of mobile networks where mobile devices establish D2D sessions underlying LTE networks [17]. Furthermore, in the context of smart homes, D2D communication utilizes protocols like Bluetooth [18], ZigBee [19], or Z-Wave [20] networks to establish D2D sessions. This communication model is commonly used in home automation systems where information is exchanged between devices small data packets with relatively low data rate requirements. For example, in a home automation setup, light bulbs and switches, thermostats, door locks and possibly other domestic devices transmit messages to each other.
- B. *Device to Cloud Communication patterns*: D2C communication models a scenario where an MTC device establish a direct communication link to a cloud server over the internet to exchange all types of messages. Using the IP based

protocol, WiFi links connect the MTC devices to the cloud. For example, a smart thermostat transmits data to a cloud database server where the data can be gathered for statistics or other analysis purposes. Home users can activate or deactivate the heating system from their smart phones via the cloud. Therefore, D2C enhances user capabilities to extend their controls beyond home environment.

- C. *Device to Gateway Communication patterns*: In D2G, MTC devices communicate with a local gateway where some processing occurs before transmitting the data to the cloud server. A home automation system or any portable smart device can act as a local gateway. D2G is mainly used for interoperability purpose where a new non compatible MTC device wants to join the network. However, D2G communication method requires an additional application layer gateway software which increases complexity and cost.
- D. *Back End Data Sharing patterns*: in back end data sharing, multiple application server providers (ASP) exchange information over the cloud for several purposes. For example, in a big complex, the data that is collected from MTC sensors is gathered in a standalone data silo. Back end data sharing architecture allows other ASP to access, analyze the data and possibly sharing it with another ASP.

Despite the large benefits of IoT, there are many challenges [16] that have to be tackled before implementation. Some of which are: (1) Security issues. (2) Privacy considerations. (3) Interoperability/standards issues. (4) Regulatory, legal, and rights issues. (5) Emerging economy and development issues. (6) Implementation of the new internet protocol IPv6. While IPv4 can support 4.3 billion devices connected to the internet, IPv6 can provide 2^{128} addresses. A survey on MTC challenges in the context of Long Term Evolution (LTE) networks is provided in [17].

Another core challenge is the large amount of data exchanged between sensors, meters and local or remote application servers. Particularly, when the number of MTC devices is large which makes it difficult to instantly manage and process the data. An alternative approach for reducing the overall data exchanged over the network from a large number of MTC devices is the data aggregation, processing and actuation at the network level [14] where data is processed prior to being exchanged over the network. This in-network processing approach reduces the data transfer over the entire network and related data to a specific sub-network is only transmitted. Furthermore, a probabilistic rateless multiple access scheme for a large number of MTC devices is investigated in [21] to efficiently minimize the access delay. The delay requirement is achieved by allowing MTC devices to transmit simultaneously on the same channel in an optimal probabilistic manner based on individual MTC devices delay requirement. It is shown that this scheme outperforms existing MTC schemes used in LTE advanced standard. Moreover, cognitive medium access control protocols for MTC networks from a stack perspective is studied in [22] to exploit the available spectrum holes in the neighboring environment.

Destination cooperation in interference channel is another flavor of D2D communication where one device can act as a relay for another device. In [23], the

authors, propose a cooperative communication scheme for two mobile MTC located in a common coverage area. The main idea of the scheme is that both MTC are opportunistically able to receive and decode each other messages based on the signal-to-interference-plus-noise ratio (SINR). If one MTC has correctly decoded both own and other MTC messages, the MTC can cooperate by relaying the other MTC message. It was shown that this scheme achieves a diversity order of 2. The same authors propose an extension to this scheme [24] where cooperation may occur not only when both own and other MTC messages are received correctly, but when just own message is decoded correctly by relaying own message. This own message is considered as an interference from the other MTC perspective. Due to the short distance between MTC, the other MTC receives and decodes the other MTC message correctly, then removes it from the main received signal to recuperate its own message. It has been shown that the extended scheme achieves additional gain while maintaining the same diversity order of 2. The authors in [25] propose a cooperative coding scheme for three MTC in vicinity underlying LTE networks with raptor coding and amplify and forward (AF) relaying scheme.

In this Chapter, we will show that the interference resulting from coexisting radio frequency channels can be exploited to improve the overall bandwidth efficiency of a communication channel. Interference management techniques are mainly divided into the following categories: (1) avoiding interference by using orthogonal channels, (2) treating the interference as noise, (3) exploiting interference by decoding it. The first technique is used when the power of interference is equivalent to the wanted signal. This results in dividing the degrees of freedom (DoF) of the channel among the users. The second technique is used when the level of interference is weak. Then single user decoding is applied. When treating interference as noise, the information contained in the interference is lost. This reduces the overall rate. The first two techniques, which are conventionally used in practice, result in an inefficient use of resources. Alternatively, exploiting interference is used when the interfering signal is stronger than the desired signal. Decoding the interfering signal first, then apply successive interference cancellation (SIC) [26], the desired signal is then decoded interference free [27].

The relay plays a crucial role in the performance of the proposed cooperative MTC strategy. Since the relay is not interested in decoding individual MTC messages, the simple AF relaying technique can be used at the relay. However, AF scheme introduces noise propagation from one hop to the other. Recently, a new promising relaying scheme termed De-Noise and Forward (DNF) has been devised by Popovski et al. [28] to mitigate the noise propagation. Koike-Akino et al. [29] propose a DNF scheme where an optimized design of constellation and mapping was studied. Sorensen et al. [30] extend the concept of DNF to non-coherent BFSK modulation scheme where the requirement for phase tracking is avoided. They show that BFSK with DNF exhibits lower performance which is due to the larger SNR required in order to yield similar achievable rates compared to BPSK. R. Chang et al. [31] propose a joint DNF scheme that exploits the correlation among multiple received MTC signals at the relay to enhance the de-noising

process during the MA phase. The scheme exhibits high complexity and implementation concerns at the relay with respect to the negligible gain achieved.

1.2 Contributions and Organization of the Chapter

The contribution of this chapter is twofold. On the one hand, it aims at investigating a new cooperative joint network and channel coding strategy for MTC devices in the multicast settings where three or more MTC devices dynamically form a cluster to disseminate messages between themselves. In the basic cluster, MTC devices simultaneously transmit their messages to the relay during the first phase. The relay broadcast back the combined messages to all MTC devices within the cluster. Given the fact that each MTC device can remove its own message, the received signal in the second phase is reduced to the combined messages coming from the other two MTC devices. Hence, this results in exploiting the interference caused by one message on the other and therefore improving the bandwidth efficiency. Moreover, the basic scheme extends to N devices. The extension of the basic scheme is discussed in details in Sect. 3. On the other hand, the proposed strategy is implemented using practical Raptor codes with two relaying schemes namely AF and DNF. It is demonstrated that DNF relaying scheme outperforms AF scheme at the expense of a little processing. It is shown that the overall proposed scheme achieves a near optimal sum rate performance. The optimal sum rate is defined as the summation of all individual rates such that reliable decoding at each MTC device's end is possible. Therefore, the rates configuration, at each MTC device and per each TS as shown in Table 1, is the optimal configuration to guarantee that the decodability constraint is met (i.e., at least a half-rate coded message is available at each MTC device end).

The rest of the Chapter is organized as follows. Section 2 presents the system model and discusses some related materials. Sections 3 and 4 compile the major research contributions of this Chapter, i.e. the proposed coding strategy and the proposed De-Noise and Forward relaying scheme, respectively. Section 5 describes the implementation of the proposed scheme and presents the simulation results. Conclusions of the Chapter and an outlook to future research directions are included in Sect. 6.

Table 1 Transmission rates configuration during phase one. The core idea behind such rate distribution is to guarantee that, in any TS, there exists at least one half-rate coded message at each MTC device end, after removing its own message

Time slot	MTC device 1	MTC device 2	MTC device 3
1	1	1/2	1/2
2	1/2	1	1/2
3	1/2	1/2	1
Sum rate	2	2	2

2 System Model and Background

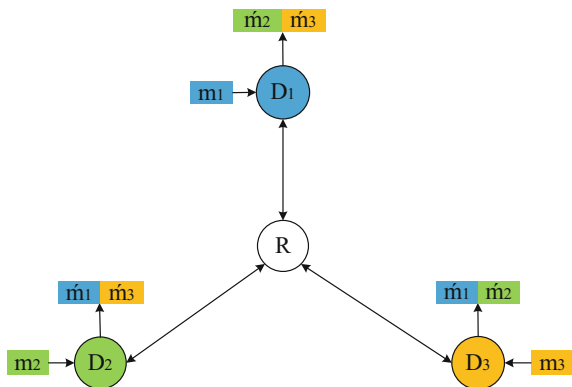
In this Section, we first introduce the system model along with some notations, then we present some background materials which will be used throughout the chapter.

2.1 System Model

Consider the scenario where three MTC devices wish to exchange their messages via a relay R as illustrated in Fig. 2. Let D_i denote the i th MTC device with respective message m_i for $i = \{1, 2, 3\}$. The devices can be any radio frequency identification (RFID) devices, sensors or meters, wireless enabled consumer electronics, wireless enabled home appliances, mobile phones or tablets, or any other type of networked MTC devices. The relay can be any repeater, remote radio head (RRH), wireless router, or another close by MTC device with limited short range transmission to cover three or more devices in proximity forming a cluster. With the rapid grows in chip design and signal processing, highly integrated chips are now available in the market. These powerful chips offers a complete and self-contained WiFi networking solution, allowing them to host applications. Furthermore, these chips have powerful on-board processing and storage capabilities that allow them to be used with sensors and other application specific devices with minimal development up-front and minimal loading during runtime.

The transmission strategy is over two phases namely multiple access (MA) and broadcast (BC). Note that due to the short distance between MTC devices, the channel in both MA and BC phases is considered as an additive white Gaussian noise (AWGN) channel with equal received power. Additionally, the MTC devices considered are half-duplex for practical reasons, therefore, MTC devices in vicinity cannot directly receive the messages from each other and communication occurs via the relay.

Fig. 2 Three MTC devices in proximity exchanging messages via a relay. Each MTC device sends one message and receives two messages. MTC devices are half duplex. \hat{m}_i denotes the decoded message from MTC device i



In MA, all MTC devices transmit their respective messages to the relay. The received signal at the relay is characterized by

$$Y_R = X_1 + X_2 + X_3 + Z_R \quad (1)$$

where $X_i \in \{-1, 1\}$, and Y_i are the channel input from the i th MTC device and the channel output, respectively, in the MA phase. Z_R is a zero mean Gaussian noise with variance σ_R^2 .

In the BC phase, the relay performs AF or DNF to broadcast back the combined messages to MTC devices within the same cluster. The received signal at each MTC device within the cluster is characterized by

$$Y_i = X_R + Z_i \quad (2)$$

where X_R is the combined message transmitted from the relay during the BC phase.

At the end of the BC phase, each MTC device removes its own message first, then the remaining received signal consists of the combined messages coming from the other two MTC devices. In the absence of noise, the received signal at MTC device i , Y_i , has three signal levels that correspond to the constellation points [32] d_1 to d_3 $\{-2, 0, 2\}$ with probabilities $p = \{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$, respectively. In this case, the channel model from each MTC device's perspective reduces to a two user multiple access erasure channel [26] over two hops with erasure probability $\alpha = 0.5$. Since all MTC devices transmit equally likely symbols at equal power with the same modulation, opposite symbols get erased. Therefore, a half-rate code is required to recover the erased symbols. In Section three, we propose a rateless coding strategy to solve this problem.

2.2 Background

The related materials in this chapter are briefly described in this Section. These include successive interference cancellation technique, the binary erasure multiple access channel model, the notion of physical layer network coding, the concept of de-noise and forward relaying scheme, and the principles of rateless coding.

- A. *Successive Interference cancellation*: Successive interference cancellation (SIC) [26] is an iterative process that starts with the strongest coded signal first, decodes it and then re-encodes it. The second step is to subtract the re-encoded message from the original received signal, and then decode the remaining messages.
- B. *Binary Erasure Multiple Access Channel*: For clarification purposes, we refer the reader to example 15.3.3 in reference [26]. Briefly, consider a multiple access channel (MAC) with binary inputs $\chi_1 = \chi_2 = \{0, 1\}$ and a ternary output $Y = X_1 + X_2$. When the combination at the input is either (0,1) or (1,0), this

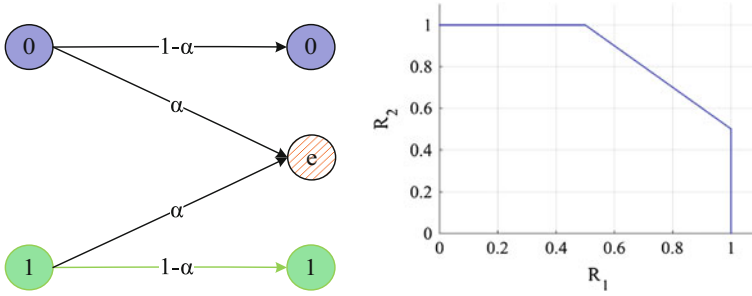


Fig. 3 (Left) Equivalent binary erasure channel for the second MTC device. (Right) Capacity region for the binary erasure MAC where the corner points are achieved with rates (0.5, 1) and (1, 0.5) and other rate points on the capacity region are achieved using time sharing

results in an ambiguous output, otherwise, it is clear what combination was sent from the inputs. Assuming that the rate $R_1 = 1$, MTC device 2 can send up to rate $R_2 = 0.5$. Hence, for X_2 the channel is like a binary erasure channel with capacity $1 - \alpha = 0.5$ as shown in Fig. 3 (left). Therefore, MTC device 2 can send additional 0.5 bits when MTC device 1 is sending at maximum unit-rate as shown in Fig. 3 (right).

- C. *Network Coding and Physical Layer Network Coding*: Network coding (NC), which is the notion of coding at the packet level, has changed the model under which communication networks are designed. While in the traditional scheme, the intermediate network nodes route a packet from the incoming link to the outgoing link without any additional processing, NC combines packets, for example, by simply bit-wise XOR'ing them. NC was originally introduced [33] to increase the rate over wired networks. However, given the unreliability and broadcast nature of wireless networks, NC can provide a natural solution for the characteristic of wireless communication that affects routing [34]. Physical layer network coding (PNC) was independently and concurrently proposed by three groups [28, 35, 36] to exploit the natural operation of NC that occurs from the broadcast of two or more non-orthogonal signals. Nazer et al. [37] proposed a reliable PNC by removing the noise at each communication phase using appropriate error correcting codes.
- D. *De-Noise and Forward Relaying Strategy*: The DNF scheme is a useful relaying strategy when the relay is not interested in decoding individual messages. DNF process detects the superposition of multiple signals at the relay and it does not decode individual messages, instead it removes completely the AWGN. The de-noising process [28] follows a certain mapping function to re-map the superposed signal into a symbol of the same constellation instead of using joint decoding. Figure 4 illustrates a simple example of DNF for two MTC devices with BPSK modulation where $X_1, X_2 \in \{-1, 1\}$. In AWGN channel, the received signal at the relay is $Y_R = X_1 + X_2 + Z_R$, where Z_R is the noise at the relay. In noiseless scenario, the received signal at the relay has three

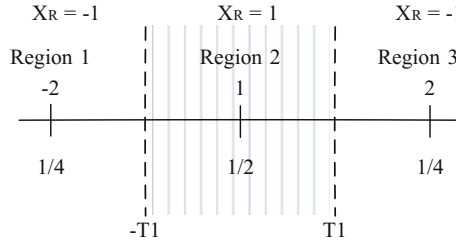


Fig. 4 Decision regions and de-noise mapping for two MTC devices exchanging messages via a relay. When received symbols at the relay fall in regions 1 or 3, the relay knows what both MTC devices sent. Whereas received symbols in region 2 are considered erased and the relay only knows that MTC devices sent opposite symbols

possible values $\{-2, 0, 2\}$. If $Y_R = \pm 2$, the relay knows what each MTC device has transmitted, however, if $Y_R = 0$ the relay cannot jointly decode both messages. Yet, the relay can apply DNF by mapping the received signal as follows. If $Y_R = \pm 2$, then $X_R = -1$, otherwise $X_R = 1$. The MTC device interprets the received signal during the BC phase as follows. If the received signal is -1 , this means that the other MTC device has sent the same symbol as his. If the received signal is 1 , the other device has sent the opposite of what this MTC device has sent during the MA phase. The optimal threshold T_1 is given by [38]

$$T_1 = 1 + \frac{\sigma_R^2}{2} \ln(2) \quad (3)$$

- E. *Raptor Codes*: Raptor codes, the first known practical class of fountain codes with linear time encoding and decoding, were originally developed for the erasure channel [39] and later investigated on binary symmetric channels [40]. Raptor code is a rateless code in the sense that given a sequence of data symbols, the encoder generates and transmits coded symbols until receiver acknowledges correct reception of the original data sequence.

3 Proposed Coding Strategy for MTC Devices

The core idea of the scheme is to increase the spectral efficiency of the channel by exploiting the interference due to the fact that more than one MTC devices transmit non-orthogonally during the MA phase. The useful interference is strongly coded to recuperate the erased symbols in the received composite signals, and therefore, the key to successful decoding of messages remains in the ability to first recover the erased symbols, then the other message is decoded interference free.

Furthermore, the proposed coding scheme extends to N MTC devices. In the first round, each ensemble of three MTC devices forms a basic cluster of order 1 to exchange their messages. In the second round, a logical cluster also consisting of three MTC devices is formed based on the only constraint that each MTC device within the logical cluster of order 2 is randomly selected from one basic cluster of order 1. This way messages are sent from lower order clusters to higher order clusters. This process continues until all $N - 3$ messages are received at the highest order cluster. The last step is to send desired messages from the highest order cluster to lower order clusters and so on until desired messages are received at basic clusters of order 1.

3.1 Cluster with Three MTC Devices

Consider the scenario in Fig. 2 where one cluster contains three MTC devices in vicinity that want to exchange their messages. In the conventional approach, one MTC device is active at a time while the other two MTC devices are silent. So the conventional scheme requires three time slots (TS) to multicast 3 bits, hence this scheme is not efficient. On the other hand, in the proposed scheme, MTC devices in proximity may exchange messages with each other through a close by relay or another MTC device acting as a relay. The relay is not interested in individual messages, in contrast, it performs AF or DNF to broadcast back the composite signal to the MTC devices. Let the messages m_{ji} and u_{ij} be row vectors of length k and $\frac{k}{2}$, respectively. The message m_{ji} carries information to be multicast from MTC device i to MTC devices j and l at unit-rate during time slot (TS) t , where $i, j, l, t \in \{1, 2, 3\}$ and $i \neq j \neq l$. On the other hand, the message u_{ij} carries half-rate coded information to be multicast from MTC device j during TS t . For symmetrical and fair data exchange between MTC devices, we consider three TS based transmission strategy.

Figure 5 illustrates an example of three consecutive TS. Since half of the bits get erased at each MTC device end, the receiver requires a half-rate coded message to be able to resolve both messages. i.e., the receiver first decodes the half-rate message, then, performs successive decoding to resolve the other message. The main idea of the coding scheme is that one MTC device transmits at a unit-rate whereas the other two MTC devices transmit at half-rate such that in the BC phase (after removing its own message), at least one half-rate message is available at each MTC device to first decode the half-rate message, then removes its effect from the received signal and therefore decode the unit-rate message, or the other half-rate message. In the case where both received messages are at half-rate, the decoder randomly decodes one of the messages first, then decodes the second message in order to preserve fairness. The introduction of TS is to maintain equal average transmission rate at each MTC device. This fairness is guaranteed through the rate configuration in each TS as shown in Table 1. Hence, MTC devices in each cluster

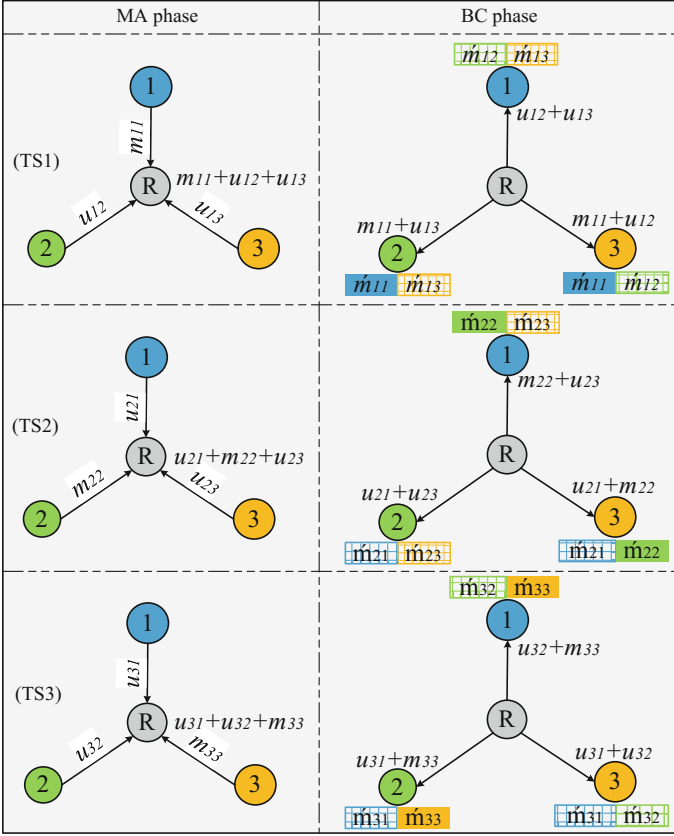


Fig. 5 Two phase proposed coding scheme for a cluster with three MTC devices. In the MA phase, m_{i1} is the unit-rate message sent from MTC device i during TS t , u_{ij} is the half-rate coded message sent from MTC device i during TS t where $i, j, t \in \{1, 2, 3\}$ and $i \neq j \neq t$. In the BC phase, MTC device i removes first its own message, then decodes the half-rate message, applies SIC to decode the unit-rate message if it is available or the other half-rate message. \hat{m}_{i1} with solid filled color denotes the unit-rate decoded message. \hat{m}_{i1} with large grid pattern fill color denotes the half-rate decoded message

know at what rate to transmit in each round. Note that a protocol can be devised to choose the rate configuration (selection of a row in Table 1) based on the amount of information each MTC device has to share with the other two MTC device. Next, we describe the example in Fig. 5.

- A. *MA Phase:* In the first TS of the MA phase, all MTC devices within the same cluster transmit simultaneously their corresponding messages m_{1i} , $i = \{1, 2, 3\}$ at rates 1, $\frac{1}{2}$ and $\frac{1}{2}$ respectively as illustrated in Table 1 to a nearby relay. In the absence of noise, the received signal Y_{1R} at the relay during the first TS is given by

$$Y_{1R} = m_{11} + u_{12} + u_{13} \quad (4)$$

Similarly, in the second and third TS with rate selection from Table 1, the received signal at the relay is, respectively, given by

$$Y_{2R} = u_{21} + m_{22} + u_{23} \quad (5)$$

$$Y_{3R} = u_{31} + u_{32} + m_{33} \quad (6)$$

Table 2 illustrates the transmitted messages from each MTC device in every TS.

B. *BC Phase*: In the first TS of the BC phase, the relay broadcasts back the composite signal to nearby MTC devices. Each MTC device, removes first its own message, then decodes the two remaining messages from the other two MTC devices. So, each MTC device decodes first half-rate message, applies SIC to decode the unit-rate message or the other half-rate message. Therefore, the received signal at MTC devices 1, 2 and 3 after removing their own messages is respectively given by

$$Y_{11} = u_{12} + u_{13} \quad (7)$$

$$Y_{12} = m_{11} + u_{13} \quad (8)$$

$$Y_{13} = m_{11} + u_{12} \quad (9)$$

MTC device 1 randomly selects message u_{12} or u_{13} to decode first, then applies SIC to decode the other message. Alternatively, both messages u_{12} and u_{13} can simultaneously be decoded by two independent decoders at the receiver since both are half-rate coded messages. On the other hand, MTC device 2 decodes first the half-rate message u_{13} , then applies SIC to decode the unit-rate message m_{11} . Similarly, MTC device 3 decodes first the half-rate message u_{12} , then applies SIC to decode the unit-rate message m_{11} . Similarly for the remaining TS 2 and 3. Table 3 illustrates the number of messages decoded at each MTC device during each TS. Therefore, after three TS, each MTC device has decoded three messages of a total rate 2. The sum rate of all MTC devices is 6 per 3 TS. That is $\frac{6}{3} = 2$, i.e., $\frac{2}{3}$ per MTC device as shown in Table 1.

Table 2 Transmitted messages m_{it} from MTC device i during time slot t

Time slot	MTC device 1	MTC device 2	MTC device 3
1	m_{11}	$\frac{m_{12}}{2}$	$\frac{m_{13}}{2}$
2	$\frac{m_{21}}{2}$	m_{22}	$\frac{m_{23}}{2}$
3	$\frac{m_{31}}{2}$	$\frac{m_{32}}{2}$	m_{33}

Table 3 Decoded messages \hat{m}_{ij} during time slot t from MTC device j

Time slot	MTC device 1	MTC device 2	MTC device 3
1	$\frac{\hat{m}_{12}}{2}, \frac{\hat{m}_{13}}{2}$	$\hat{m}_{11}, \frac{\hat{m}_{13}}{2}$	$\hat{m}_{11}, \frac{\hat{m}_{12}}{2}$
2	$\hat{m}_{22}, \frac{\hat{m}_{23}}{2}$	$\frac{\hat{m}_{21}}{2}, \frac{\hat{m}_{23}}{2}$	$\frac{\hat{m}_{21}}{2}, \hat{m}_{22}$
3	$\frac{\hat{m}_{32}}{2}, \hat{m}_{33}$	$\frac{\hat{m}_{31}}{2}, \hat{m}_{33}$	$\frac{\hat{m}_{31}}{2}, \frac{\hat{m}_{32}}{2}$

Note that a ternary channel is assumed in phase two. Note also that, the relay can be any other additional MTC device in proximity, an access point within the cluster, a remote radio head (RRH), or any type of networked device with in-network processing capabilities.

The advantage of the proposed coding strategy over the conventional scheme is in manifold. Due to the short distance between cooperative MTC devices within the cluster, the transmit power is significantly reduced resulting in a major power saving at active MTC devices. Furthermore, less transmitting power usage at MTC devices leads to less interference caused to other MTC devices in neighboring clusters, hence improving frequency reuse factor and increasing network capacity. In addition, directly established links between MTC devices with in-network processing capabilities reduce latency and release radio resources on the core network which also result in increasing network capacity. Moreover, the proposed scheme efficiently uses the spectrum due to the PNC which increases the sum rate. Finally, the relay, which can be any device in vicinity, performs a simple AF or DNF relaying.

Note that for symmetrical data rates, the considered transmission strategy is over three time slots (TS) in a round robin fashion. However, for asymmetrical data rates between MTC devices, a protocol can be devised to assign transmission rates at each MTC device based on the size of the actual locally available file for sharing. Therefore, each TS (row) of transmission rates configuration in Table 1, can be more (dynamically) selected based on the available local messages at each MTC device to exchange. Table 1 illustrates the transmission rate at which each MTC device is agreed to transmit during phase one. After a symmetrical round of transmission (three consecutive TS), the average sum rate is 2 per MTC device. Let m_{ti} denotes the transmitted message from MTC device i during TS t . In total, each MTC device transmits two messages and receives four other messages.

3.2 Aggregated Cluster with N MTC Devices

The proposed coding strategy extends to N MTC devices in proximity. The extended scheme consists of $\log_3(N)$ rounds each of which is over two phases. In the first round, each three MTC devices in closed vicinity are organized in a basic

cluster which results in $\frac{N}{3^d}$ basic clusters of degree $d = 1$, where d is the degree of the cluster. Then, the scheme devised in Sect. 3.1 is applied at each cluster of degree 1. At the end of the first round, each MTC device receives the messages of the other two MTC devices within the same basic cluster of order 1 as indicated in Table 3.

In the second round, a new logical cluster is formed based on the following constraint. Only one MTC device from a basic cluster of order 1 is allowed to be in the newly formed logical cluster of degree $d = 2$ which results in $\frac{N}{3^d}$ clusters of degree 2. Then, similarly to round 1, the scheme devised in Sect. 3.1 is applied at each cluster of degree 2 to exchange all messages within the logical cluster. In order to disseminate all messages across all MTC devices, this process continues until $d \leq \log_3(N)$. At the end, each MTC device will have received $N - 1$ new messages. Note that each randomly selected MTC device from a cluster of order l contains 3^l exchanged messages within this cluster of order l where $1 \leq l < \lfloor \log_3(N) \rfloor$. Once this procedure reaches the highest logical cluster degree, at this stage, the MTC devices within this highest cluster contain all messages in the network. Then, this process continues from the highest order logical cluster downward to basic clusters to only disseminate the wanted messages at each logical cluster of lower degree until reaching each basic cluster of degree 1.

Figure 6 illustrates an example where $N = 27$. In the first round, $\frac{N}{3^d} = \frac{27}{3^1} = 9$ basic clusters of degree 1 (clusters with green interconnecting links in Fig. 6) are formed. The formation of basic and logical clusters is described in Table 4. In each cluster of degree 1, MTC devices exchange messages as described in Sect. 3.1. For example, in cluster 1 of degree 1, MTC devices 1, 2, and 3 exchange their respective messages m_1, m_2 and m_3 . In the second round, $\frac{N}{3^d} = \frac{27}{3^2} = 3$ logical clusters of degree 2 (clusters with blue interconnecting dashed links in Fig. 6) are formed.

Figure 7 illustrates the logical cluster 1 of degree 2 where one MTC device is randomly selected from each basic cluster of degree 1 (MTC devices 3, 5 and 7) to form the logical cluster of degree 2. At the end of round 2, each MTC device (MTC devices 3, 5, and 7) will have received all messages of MTC devices within the logical cluster 2 as indicated in Fig. 7b. Then to disseminate these messages within clusters of lower degree, i.e., degree 1, only messages that are not previously available in each cluster of lower degree are exchanged. For example, in cluster 1 of degree 1 (MTC devices 1, 2, and 3) messages m_4 to m_9 are only sent downward from higher degree to lower degree. In the third round where $d \leq \log_3(27) = 3$, $\frac{N}{3^d} = \frac{27}{3^3} = 1$ logical cluster of degree 3 (cluster with thick orange interconnecting links in Fig. 6) is formed.

As shown in Fig. 6, only one MTC device is randomly selected from each of the three logical clusters of degree 2 (MTC devices 9, 11, and 22). Each MTC device belonging to a logical cluster of degree 2 contains all the messages from that logical cluster of degree 2 to be exchanged with the other two logical clusters of the same degree 2. At the end of round 3, each MTC device will have received $N - 1 = 26$ messages coming from all other devices participating in the MTC network. For example, MTC device 9 has initially message m_9 to exchange with $N - 1 = 26$ MTC devices as illustrated in Fig. 6a. In the first round, MTC device 9 is part of cluster 3

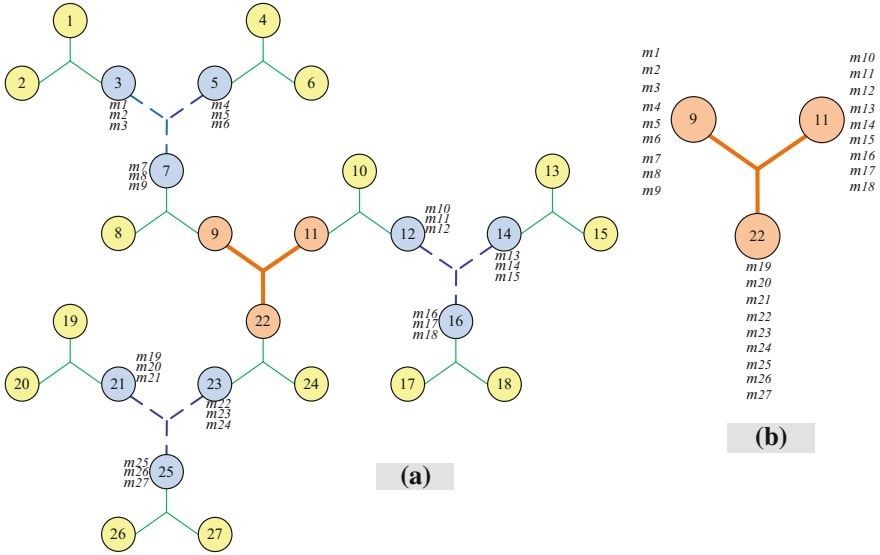


Fig. 6 Example of MTC strategy with clustering technique of dimension $N = 27$. Clusters with *green* interconnecting links are of degree 1, clusters with *blue* interconnecting links are of degree 2 and cluster with *orange* interconnecting links is of degree 3. In total, we have 9 basic clusters of degree 1, 3 clusters of degree 2, and 1 cluster of degree 3. The *blue* and *orange* interconnecting links guarantee dissemination of messages over the entire MTC network. Messages moving across clusters go through interconnecting links based on the scheme devised in Sect. 3.1. Note that the intersecting point that connects that joins three interconnecting links is a relay node regardless the degree of the cluster. **a** Illustrates all MTC devices in the network including the interconnecting links for each cluster degree. **b** Illustrates the highest degree cluster where all messages are gathered

Table 4 Formation of basic and logical clusters of the example in Fig. 6

Cluster degree No	Cluster given No	MTC device No
1	1	1, 2, 3
	2	4, 5, 6
	3	7, 8, 9
	4	10, 11, 12
	5	13, 14, 15
	6	16, 17, 18
	7	19, 20, 21
	8	22, 23, 24
	9	25, 26, 27
2	1	3, 5, 7
	2	12, 14, 16
	3	21, 23, 25
3	1	9, 11, 12

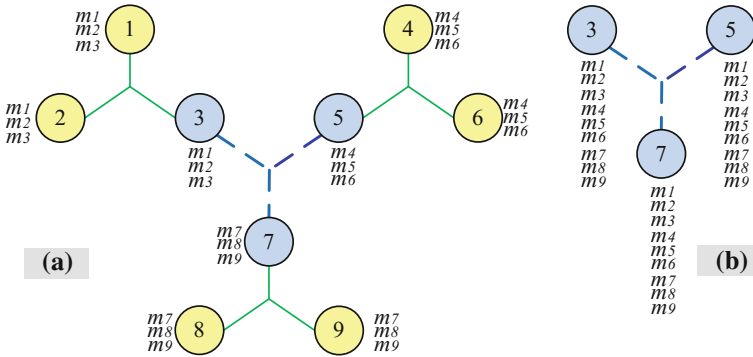


Fig. 7 **a** Three clusters of order 1. In each cluster, MTC devices exchange messages. In cluster 1, MTC device 1, 2, and 3 exchange messages m_1, m_2, m_3 . Similarly for clusters 2 and 3. In the second round, three MTC devices are randomly selected from three cluster of degree 1 with the constraint that one MTC device is selected from each cluster of degree 1. MTC devices 3, 5 and 7 are selected to form cluster 1 of degree 2. **b** Illustrates all the exchanged messages at each MTC device of cluster 1 of degree 2 at the end of the second round

of degree 1 (MTC devices 7, 8 and 9). At the end of round 1, MTC devices 7, 8, and 9 contain the messages $m_7, m_8,$ and m_9 . In the second round, MTC device 7 forms a logical cluster of degree 2 with MTC device 3 and 5. At the end of round 2, MTC device 7 contains the messages $m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8,$ and m_9 . Newly exchanged messages from this round at MTC device 7 (i.e., m_1, m_2, m_3, m_4, m_5 and m_6) will be exchanged internally with in the basic cluster 3 and therefore, MTC device 9 will have all the messages of logical cluster 1 of order 2 as illustrated in Fig. 6b. In the third round, one logical cluster of order 3 is formed which contains MTC devices 9, 11, and 22 each of which belongs to one logical cluster of order 2 as shown in Fig. 6a. At the end of this round, each of three MTC devices 9, 11, and 22 will contain all the messages in the network. Finally, only desired messages at each logical cluster of lower order are disseminated. i.e., m_{10} to m_{27} are sent through logical cluster 1 of degree 2 and eventually through basic clusters 1, 2 and 3 of degree 1.

As per the overhead related to cluster formation, a protocol can be devised to manage MTC devices in vicinity that wish to exchange messages. In such scenarios, the nearest relay receives requests from close by MTC devices wishing to form a group for message sharing. Based on the number of available relays for servicing these MTC devices, the protocol allocates each ensemble of three MTC devices a single relay. Then each relay sends a response message to MTC devices in vicinity to declare the formation of a cluster. Note that the function of a relay in clusters of various degrees (basic or logical clusters) is similar and therefore, clusters are aggregated as shown in Fig. 6.

On the other hand, cooperative communication [41] is a key enabling technology for increasing spectral efficiency. The relay plays a vital role in the proposed communication strategy. In domestic areas, the relay can be a home access point

(AP) or any other smart device with in-network processing capabilities [42]. The number of MTC devices in a domestic environment is proportionally small and therefore can be served by the home AP which is acting as a relay. However, in larger areas such as malls, universities, airports, and so on, many AP's are deployed all over to ensure nonstop hotspots Wi-Fi services in the entire place. While these AP's are mainly used for Wi-Fi access, they can also serve as relays.

Let L denote the number of relays required for a given network of N MTC devices. Based on the quality of service requirement and the type of application, L can take any value between 1 and L_{max} where L_{max} denotes the maximum number of relays, as a function of N and is given by

$$L_{max} = \sum_{i=1}^{\log_3 N} \frac{N}{3^i}. \quad (10)$$

However, many uncritical applications, such as file sharing, data mirroring and so on, can use any available number of deployed relays in the neighborhood that is $< L_{max}$. When $L = L_{max}$, each cluster is assigned a dedicated relay. When $L < L_{max}$, clusters share relays in time division manner. In the example of Fig. 6, $L_{max} = 13$.

4 De-Noise and Forward Relaying Scheme

The proposed DNF relaying strategy is simple and exhibits lower complexity. The de-noising process consists of mapping the received signal at a relay in vicinity to the nearest constellation point. This process removes the noise propagation to the next hop. However, this process may introduce decision error. By appropriately selecting the decision threshold, we minimize this error.

4.1 A Simple De-Noiseing Strategy

In the MA phase, the received signal at the relay in noiseless scenario is the summation of three messages coming from three MTC devices. With the fact that each MTC device utilizes BPSK modulation, this results into four regions symmetric constellation $\{-3, -1, 1, 3\}$ with probabilities $\{1/8, 3/8, 3/8, 1/8\}$ respectively as shown in Fig. 8. The decision rule that minimizes the probability of error is the maximum a posteriori (MAP) probability decision rule [32]. Hence, the optimal threshold T_2 is defined as the edge that separates two decision regions in a

given constellation [38]. T_2 is computed as follows $T_2 = 1 + \frac{3-1}{2} + \frac{\sigma_R^2}{2} \ln\left(\frac{3/8}{1/8}\right)$. Therefore, T_2 becomes

$$T_2 = 2 + \frac{\sigma_R^2}{2} \ln(3) \tag{11}$$

4.2 Decision Regions and De-Noising Mapping

Figure 8 illustrates the decision regions and the per-symbol de-noise mapping applied at the relay. The transmission of $(X_1, X_2, X_3) = (-1, -1, -1)$ and $(X_1, X_2, X_3) = (1, 1, 1)$ are distinctive at the relay and fall in region 1 and 4 respectively. In these two cases the relay knows without ambiguity what each MTC device has transmitted. However, the relay is not interested in decoding individual messages, therefore, the relay applies the de-noising process which maps any received signal in the unshaded regions to 3. Actually, when the signal is in either regions one or four which is considered as reliable information, each MTC device knows what the other MTC devices transmitted during the MA phase. Therefore, we only require a ternary code to represent the two ambiguous regions (two and three) and the clear region (regions one and four).

On the other hand, the transmissions of $(X_1, X_2, X_3) = (-1, -1, 1)$, $(X_1, X_2, X_3) = (-1, 1, -1)$ and $(X_1, X_2, X_3) = (1, -1, -1)$ are indistinguishable at the relay. They all result in the same region two. Similarly for the other remaining three cases which result in region three. In those six cases, the relay does not know what exact combination was transmitted during the MA phase, however, the relay removes the additional AWGN noise by mapping the signal that falls in the shaded region two to -1 and by mapping the signal that falls in the shaded region three to 1 .

In the BC phase, the relay broadcasts one superimposed/de-noised message to all MTC devices in vicinity. Each MTC device first removes its own message. Then the resulting signaling reduces to the model with constellation regions similar to the

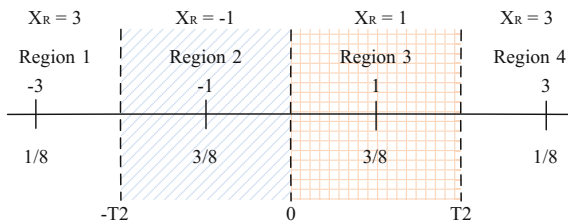


Fig. 8 Decision regions and de-noise mapping for the proposed DNF relaying strategy. Received messages that fall in both regions 1 and 4 are considered as reliable information where MTC devices know what each of them sent and therefore can be merged as one region. On the other hand, regions 2 and 3 are not reliable and therefore each region is mapped to a different signal

one in Fig. 4. Hence, if the received signal (after removing its own message) falls in either of the unshaded regions one and three, the MTC device knows that the other two MTC devices have sent similar message to his during the MA phase. Otherwise, the received symbols are erased and hence, a half-rate code is required to resolve the intended messages for each MTC device.

5 Simulation Results and Discussion

We evaluate the performance of the proposed coding strategy with both AF and DNF relaying schemes using Raptor code [39] which is a two layer code. i.e., the pre-code [43] and the LT code [44]. However, the scheme can be evaluated using any half-rate channel code. We first study the performance of the system in the absence of noise, then we consider the presence of noise with both AF and DNF relaying schemes. We perform successive decoding at each MTC device to first decode the lower rate stream, then subtract it from the received signal to decode the higher rate stream if it is available or decode the other lower rate stream. Moreover, we evaluate the overheads required for each stream at all MTC devices to achieve a specific BER goal. The pre-code for Raptor code is the LDPC code [31] of rate 0.98 and left degree 4. The number of LDPC decoding iterations is 50 and the number of LT decoding iterations is 300.

The optimized output degree distribution for Raptor code [39] is given by

$$\begin{aligned} \Omega(x) = & 0.008x + 0.49x^2 + 0.166x^3 + 0.073x^4 + 0.083x^5 \\ & + 0.056x^8 + 0.037x^9 + 0.056x^{19} + 0.025x^{65} + 0.003x^{66}. \end{aligned} \quad (12)$$

We consider a block length $k = 65536$ of information bits that are pre-coded with LDPC code at rate 0.98 to generate $k' = 66873$ intermediate bits, then n encoded symbols are generated from the intermediate bits by the LT encoder according to the distribution in (12). The relation between the number of produced output symbols n and the overhead ε is given by

$$n = k(1 + \varepsilon) \quad (13)$$

5.1 Mechanism to Assign Transmission Rates for MTC Devices

The initial step is to decide at what rate each MTC device will transmit. This can be managed by allowing the MTC devices to initially transmit few bits to the relay declaring the amount of data each has to exchange, then the relay selects the rates accordingly by assigning unit-rate to MTC device with the largest file and so on.

Table 5 Adaptive rate transmission mechanism between MTC devices. Relay assigns a rate selection scheme (Rows in Table 1) based on inputs from MTC devices. However, random selection or equal Average rate selection (default selection code 111) can also be assigned

Broadcast code by the relay	Selected rows from Table 1
001	Row 3
010	Row 2
011	Row 2, row 3
100	Row 1
101	Row 1, row 3
110	Row 1, row 2
111	Row 1, row 2, row 3

Alternatively, the relay can randomly pick up any rate combination (any row from Table 1) or select the fair combination between the MTC devices (Rows 1, 2, and 3 from Table 1, in a round robin fashion). Table 5 illustrates the various rate allocation schemes and their associated codes. When one single row is selected, i.e., code 001, 010 and 100, each MTC device will transmit at the assigned rate in every TS until an update is received to change the selected rates. On the other hand, when more than one row is selected, the rows are assigned in a round robin fashion. In the equal average rate scenario i.e., code 111 (default setting), the transmission rates between MTC devices is changed in a round robin fashion as indicated in Table 1.

In each TS, one MTC device transmits uncoded source symbols stream while the other two MTC devices transmit n encoded symbols using Raptor code. Encoded symbols from respective MTC devices are generated and transmitted until an acknowledgment from both respective MTC devices is received at the relay. Then, the relay will send one stop message to both respective encoders. This message can be a 3 bit all zero code (000).

Note that the Raptor code rates are not known a priori. It depends on the channel quality. However, in an erasure channel with 50 % of the received symbols are erased on average and links are considered as AWGN channel, the average number of symbols required at the respective MTC devices is nearly double to be able to resolve the received messages. Therefore, the average rate is around half.

5.2 System Performance in the Absence of Noise

The performance is only affected by the erased symbols. In any TS and at any MTC device, the receiver decodes first half-rate message, re-encodes it, and then removes its effect from the received signal to resolve the unit-rate message (where it is available) or the other half-rate message. Theoretically, we need exactly half-rate (twice the number of output symbols) to be able to compensate for the erased symbols and decode the message at the receiver, however, with Raptor code we need a few more reliable symbols to fully resolve the message.

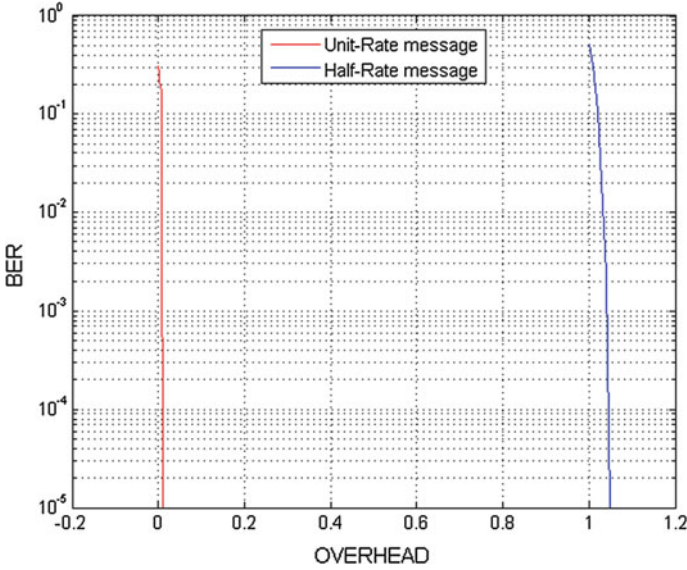


Fig. 9 BER versus overhead for unit-rate and half-rate messages. (Note that for half-rate message, half of the additional coded symbols are erased as well)

As shown in Fig. 9, the overhead required for considerably low BER is $\epsilon_{0.5} = 1.05$ and $\epsilon_1 = 0.01$ for half-rate and unit-rate messages, respectively. In other words, the decoder requires about 1 % of additional symbols in the case of unit-rate messages and about 5 % of additional symbols (on top of the number of symbols that were erased) to fully decode their respective messages. Furthermore, the overhead required for half-rate message is slightly higher compared to unit-rate message since first, half of the additional coded symbols (above overhead 1) are also erased, and second, the distribution in (12) is optimized [39] for a block length k and not $\frac{k}{2}$. Consequently, for equal length codewords at all MTC devices, the simulation overhead $\epsilon_{sim} = 0.025$ as given by the following maximization function

$$\epsilon_{sim} = \max \left\{ \epsilon_1, \frac{\epsilon_{0.5} - 1}{2} \right\} \quad (14)$$

Additionally, since the distribution in (12) is mainly optimized for block length $k = 65536$, the second term in (14) is most likely to dominate. Hence, the average sum rate over three TS is $R = 1.952$ as given by

$$R = \frac{k}{n} = \frac{2}{(1 + \epsilon_{sim})} \quad (15)$$

The sum rate R decreases as ϵ_{sim} increases. In the theoretical case, i.e., when $\epsilon_{sim} = 0$, $R = 2$. Figure 10 illustrates the sum rate as a function of the overhead.

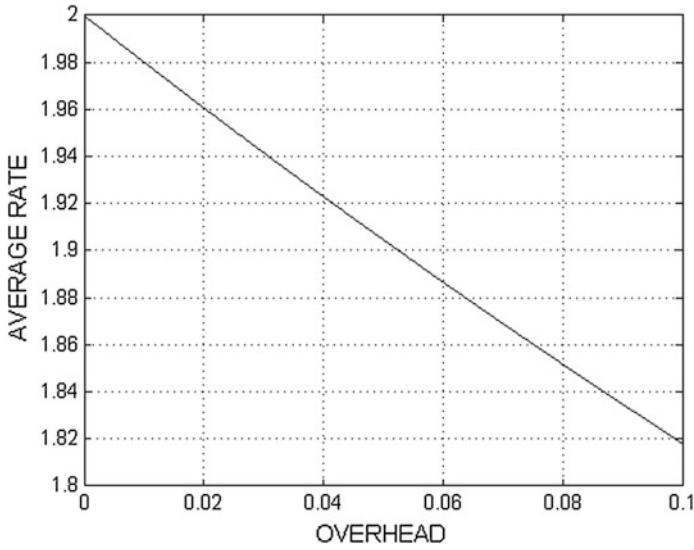


Fig. 10 Average sum rate as a function of the simulation overhead ϵ_{sim} . As the overhead increases, the average sum rate decreases

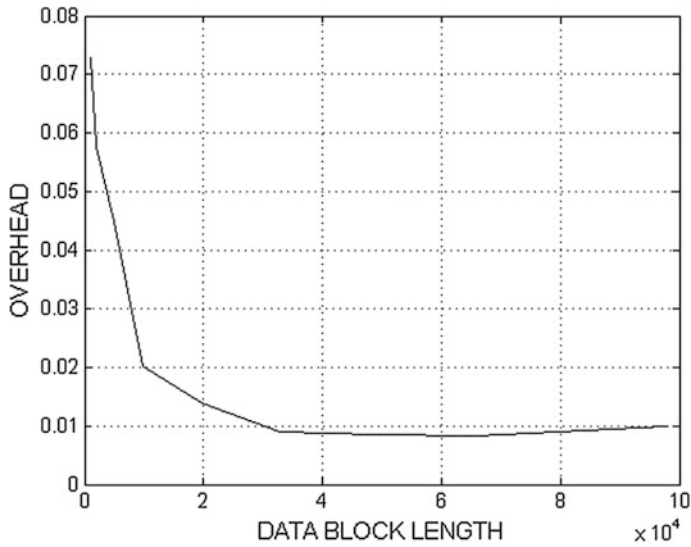


Fig. 11 Overhead as a function of the data block length k . The lowest overhead is exhibited for $k = 65536$

Figure 11 shows the overhead as a function of the data block length. Note that the lowest overhead [39] is for block length equal to 65536 since the output distribution in (12) is optimized for this value.

5.3 System Performance in the Presence of Noise with AF and DNF Relaying Schemes

We consider the performance in the presence of the Gaussian noise that is produced at a close by relay and at each MTC device. We assume the same noise variance at all nodes. The received signal at any MTC device (after removing its own message) is the superposition of two messages coming from the other two MTC devices. Note that, although the unit-rate and half-rate messages are lower than one and half respectively (due to noise), we keep referring to both messages as unit-rate and half-rate for simplicity. The key factor of successful decoding of all messages at any MTC device begins with the ability to decode half-rate message. Note that in the noiseless scenario, the half-rate message requires almost double the transmitted symbols to be able to fully decode the message. Whereas in the noisy scenario, it obviously requires further symbols (more overheads) to compensate for the noise. The unit-rate message is almost surely decodable since ε_{sim} is most likely greater than ε_1 the overhead required for successful decoding of unit-rate message.

- A. *AF Relaying Scheme*: We first consider AF relaying scheme where the relay amplifies the received signal that is received during the MA phase and then broadcast it to all MTC devices during the BC phase. AF scheme is simple and does not require any processing at the relay. However, due to noise propagation, it performs poorly in the low SNR regime. In the BC phase the relay amplifies the received signal by a factor $\beta = \sqrt{\frac{1}{3 + \sigma_R^2}}$ and broadcast βY_R to MTC devices, where β is the amplifier gain to maintain the average power constraint at the relay and σ_R^2 is the noise variance.

MTC device i first removes own message m_i from the received signal. Then, the remaining composite signal at MTC device i is the superposition of the other two messages coming from MTC devices j and l , plus noise. Assuming that $i = 1$, the remaining messages M_j and M_l have been encoded at half-rate and full-rate respectively (rate selection as row 3 in Table 1). We first decode the half-rate message, then re-encode \hat{X}_j and remove it from the received signal to decode \hat{X}_l interference-free.

For every SNR point, we first simulate the amount of overheads required to achieve a BER target less than or equal to 10^{-5} on the half-rate message as shown in Fig. 12 (AF). Then, we apply SIC to recover the unit-rate message. The unit-rate message is easily decoded since the simulation overhead ε_{sim} is always greater than the overhead required for decoding unit-rate message ε_1 as illustrated in the maximization function in (14). Once this BER target is reached, we note the simulation overhead ε_{sim} and apply (15) to compute the sum rate as illustrated in Fig. 13 (AF).

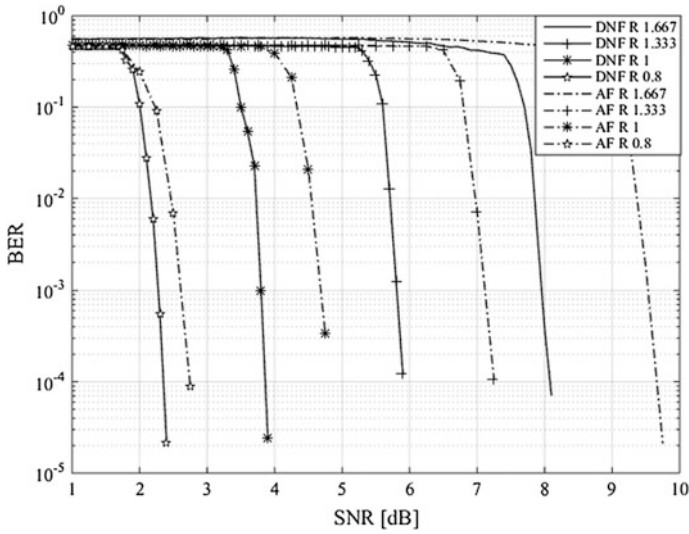


Fig. 12 BER as a function of the average SNR for various achievable sum rates R with AF and DNF relaying schemes. The various achievable sum rates reflect the various overheads required at various SNR's. Specifically, for every SNR point, we evaluate the required overhead $\epsilon_{0.5}$ to achieve a specific BER target $\leq 10^{-5}$. Then using (14), we calculate the simulation overhead ϵ_{sim} , and finally, we calculate the average sum rate using (15). On the other hand, the legend can be represented in terms of the overhead $\epsilon_{0.5}$. i.e., for sum rates $R=1.667$, $R=1.333$, $R=1$, and $R=0.8$, the $\epsilon_{0.5}$ overheads are 1.4, 2, 3, and 4, respectively

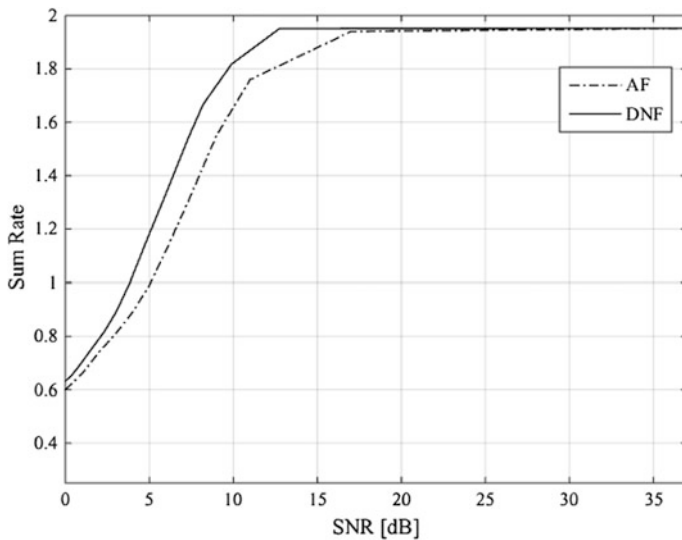


Fig. 13 Sum rate as a function of the average SNR with AF and DNF relaying schemes. The overhead required for low SNR increases and therefore results in a lower average sum rate

B. *DNF Relaying Scheme*: In DNF, the relay applies the per-symbol de-noising process at the received signal as follows.

$$X_R^{DNF} = \begin{cases} -1 & \text{if } -T_2 < Y_R < 0 \\ 1 & \text{if } 0 \leq Y_R < T_2 \\ 3 & |Y_R| \geq T_2 \end{cases} \quad (16)$$

where Y_R is the received composite signal at the relay during the MA phase, T_2 is the optimal threshold as indicated in (11). To maintain the power constraint at the relay, the mapped symbols are also scaled by β .

Simulation results show that the optimal de-noising threshold T_2 for $SNR \leq 2$ is the MAP [32] as in (11). However, for $SNR > 2$, the optimal threshold is fixed to the minimum possible with $T_2 = 2$. Similarly to AF relaying scheme, at each SNR point, we simulate the overheads required to achieve a BER target less or equal to 10^{-5} on the half-rate message as shown in Fig. 12 (DNF). Then we apply SIC to recover the unit-rate message. Then after, we note the simulation overhead ε_{sim} and apply (15) to compute the sum rate as illustrated in Fig. 13 (DNF).

Error free transmission is assumed when $BER \leq 10^{-5}$. To illustrate the importance of the lower rate message, Fig. 12 shows the BER as a function of the average SNR for various achievable sum rates with AF and DNF relaying schemes. As SNR increases, the overhead diminishes and therefore the sum rate increases. On the other hand, at very low SNR the sum rate decreases due to the high overhead required to resolve the half-rate message. As illustrated in Fig. 12, the performance of DNF outperforms AF for the same amount of overheads.

This Gain in dB for the same overhead is interpreted into higher sum rate. Figure 13 illustrates the average system sum rate over three TS. At lower SNR, the sum rate is bounded by the average overhead required for error free transmission. The dashed line represents the sum rate with AF relaying. It is clearly shown that DNF outperforms AF relaying scheme particularly at low to moderate SNR. Since the performance of both schemes results in the same performance at high SNR, i.e., $SNR > 17$ dB, it is more appropriate to use AF as it is simpler and does not require further processing. As the overhead increases, the overall sum rate decreases. There is an interesting operational regime in which the sum rate increases linearly with the SNR, i.e. for SNR up to around 12 dB. Whereas at higher SNR, the operational sum rate approaches the upper bound. For instance, at 17 dB, the sum rate is 1.94. Furthermore, a combined DNF-AF selection relaying scheme can be used with selection SNR threshold to switch between AF and DNF.

C. *Overhead Analysis with DNF Relaying Scheme*: In Rateless coding, the overhead at each MTC device depends on the end to end channel quality. In this

paper, an equal receive signal power is assumed at MTC devices within a cluster. While in a noiseless scenario the overhead required for half-rate messages achieving error free transmission is mainly proportional to the number of erasures (50 % on average) in the received combined codeword, the overhead required in the presence of noise is higher to compensate for the errors due to the Gaussian noise. Therefore, the overhead experienced at each MTC device is the overhead required to successfully decode both intended messages. For instance, the received signal at MTC device 3 after removing its own message is given by

$$Y_3 = \beta X_R + Z_3 \tag{17}$$

where X_R is the de-noised combined message at the relay which is a function of the coded messages X_1 and X_2 from respective MTC devices with coding rates as illustrated in Table 1. i.e., (X_1, X_2) can take the rate pairs (1, 0.5), (0.5, 1), and (0.5, 0.5). Using successive decoding, half-rate message is first decoded, subtracted from received signal, then second message is decoded interference free. The overhead versus signal to noise ratio is illustrated in Fig. 14 for both half-rate and unit-rate messages with DNF relaying scheme. It is clear that the overhead decreases as the SNR increases and the overhead for half-rate message is higher than the overhead for unit-rate message for a given SNR point.

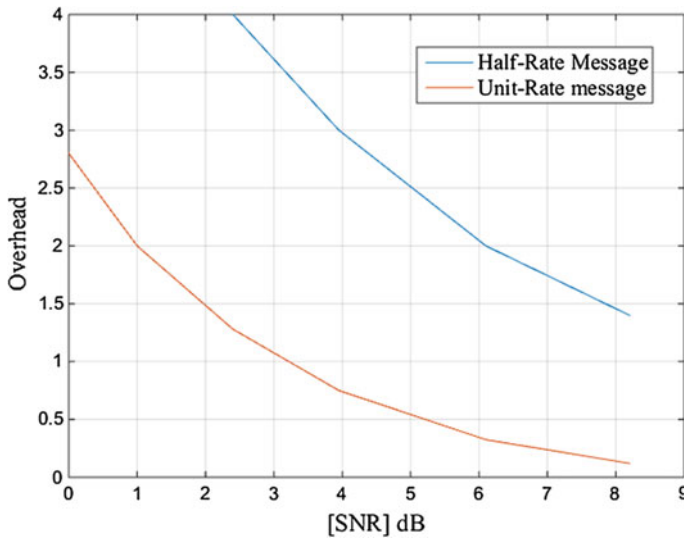


Fig. 14 Overhead versus SNR for half-rate and unit-rate messages. Half-rate message is decoded first

5.4 Performance Comparison of the Proposed Coding Strategy with Existing Traditional Single Device Communication and Functional Decode and Forward Coding Strategies in the Presence of Noise

In this Section, we compare the performance of the proposed coding strategy with two coding schemes: (1) single device communication where one MTC device is active at a time. (2) Functional decode and forward where two MTC devices are active at a time during the MA phase. In the following, we describe and evaluate the performance of both strategies, then we compare them with the proposed scheme

- A. *Traditional Single Device Communication Strategy*: In this strategy, one MTC device is active at a time. i.e., one MTC device directly transmits its message to the other two MTC devices (which are in receiving mode). Therefore, in each cluster, three TS are required to exchange three messages between MTC devices. Hence, the sum rate is $\frac{3}{3} = 1$ and therefore, this scheme is not efficient. Figure 15 illustrates the traditional communication strategy over one TS where MTC device 1 is active. The received signal at any of the inactive MTC devices is characterized by

$$Y_{ij} = X_i + Z_{ij} \quad (18)$$

where X_i is the unit-rate BPSK modulated encoded message, Y_{ij} and Z_{ij} are the received signal and the Gaussian noise at MTC device j during TS t , respectively. To compensate for the Gaussian noise at each MTC device end, the messages are Raptor encoded before sent to the other MTC devices. For instance, during the first TS, the received message at MTC device 2 is characterized by

$$Y_{12} = X_1 + Z_{12} \quad (19)$$

Since the performance of all messages is similar during any TS (also assuming that the received signal at the inactive MTC devices is the same), we evaluate, the

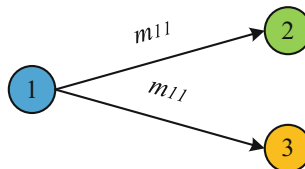


Fig. 15 In the first TS, MTC device 1 transmits its message to the other MTC devices 2 and 3. Communication between each active MTC device and the other two MTC devices is interference free. m_i is the unit-rate message sent from MTC device i

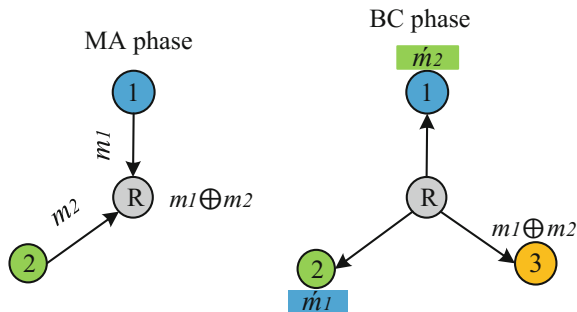


Fig. 16 FDF coding scheme for three MTC devices where MTC device 1 and 2 are paired in the MA phase. In the BC phase, MTC device 1 and 2 after removing their own messages, can decode the other MTC device's message interference free. However MTC device 3 cannot decode both messages at this stage and messages are fully resolved in the next TS

performance of m_1 during the first TS for illustration purposes as indicated in Fig. 15. The simulation results are shown in Fig. 17, where for every SNR point, we simulate the overhead required to achieve a BER target $\leq 10^{-5}$. Using the following relation, we compute the achievable sum rate

$$R_{TRAD} = \frac{1}{(1 + \epsilon)} \quad (20)$$

where ϵ is the required overhead to achieve a specific BER target.

B. Functional Decode and Forward Relaying Strategy: The idea of functional decode and forward (FDF) was initially proposed for two MTC devices in [35] and [45]. Using time division multiple access (TDMA) and user pairing, this idea was extended to more than two MTC devices in [46]. More specifically, the MA transmission is split into $N - 1$ time slots. In each TS, two MTC devices are active only. For instance when $N = 3$, two TS are required to fully exchange three messages and therefore, the sum rate is $\frac{3}{2}$. Figure 16 illustrates the FDF scheme for three MTC devices during one TS. In this TS, MTC device 1 and 2 are paired during the MA phase. The relay is also not interested in decoding individual messages, however, the relay decodes a function of the received combined messages and transmits it during the BC phase. Then, the paired MTC devices are capable of decoding each other messages from the function message that was received during the BC phase and their own message. The third MTC device is unable to decode the function message at this stage and it will be fully resolved in the next TS. The FDF process avoids the noise propagation at the relay. A simple example of FDF scheme [46] is when the relay uses the XOR function i.e. $m_1 \oplus m_2$. The relay broadcast $m_1 \oplus m_2$ to all MTC devices within the cluster. Paired MTC devices can decode the

exchanged messages by using the XOR function of their own message with the received function message. The received signal at the relay during TS t is characterized by

$$Y_{tR} = X_i + X_j + Z_{tR} \quad (21)$$

where X_i and X_j are the unit-rate BPSK modulated (uncoded) messages from the paired MTC devices during TS t , respectively. Z_{tR} is the Gaussian noise at the relay during TS t . In the BC phase, the relay broadcast $X_{tR} = f(Y_{tR})$. The received signal at MTC device j during TS t is characterized by

$$Y_{tj} = X_{tR} + Z_{tj} \quad (22)$$

The performance of messages at paired MTC devices is similar during any TS. To compensate for the Gaussian noise at the relay and all MTC devices, the messages are Raptor coded before transmitted to the relay. The simulation results are illustrated in Fig. 17, where for every SNR point, we simulate the overhead required to achieve BER target $\leq 10^{-5}$. Using the following relation, we compute the achievable sum rate

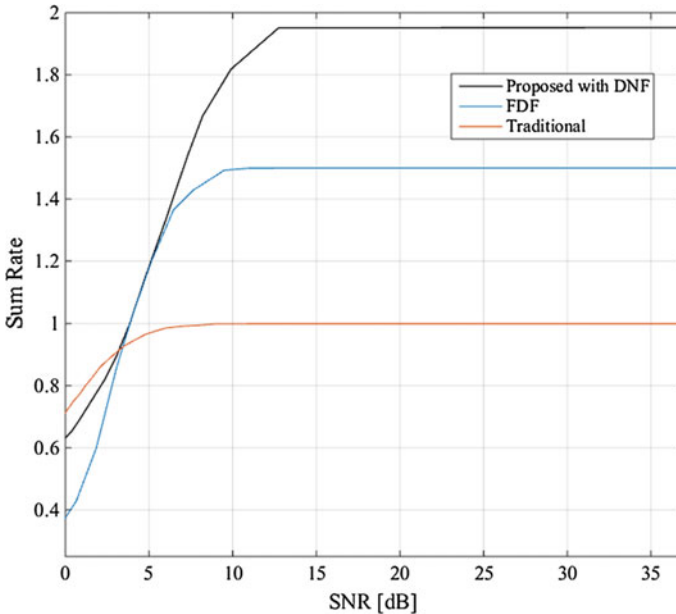


Fig. 17 Performance comparison between different schemes. The proposed scheme is the most efficient as it exploits the interference to increase the sum rate. On the other hand, in FDF scheme, messages are decoded interference free with TDMA and pairing. The traditional scheme is the most inefficient and it is used for illustration purposes

Table 6 Comparison between various schemes. Note that the proposed scheme achieves 4 % bandwidth saving while it requires 12 % of additional Power particularly in the BC phase

		Traditional	FDF	Proposed
No of TS	MA	3	2	1
	BC	–	2	1
Bandwidth	MA	3	2	1
	BC	–	2	1.56
Power	MA	3P	4P	3P
	BC	–	2P	3/2P
Transmitted messages		3	3	2
Messages/Bandwidth		3/3	3/4	2/2.56
Power/Message		P	2P	2.25P

$$R_{FDF} = \frac{1.5}{(1 + \varepsilon)} \quad (23)$$

where ε is the required overhead to achieve a specific BER target.

C. *Performance Comparison Between Proposed and Other Coding Schemes*: The performance comparison between the proposed, FDF and traditional coding schemes for the basic cluster (3 MTC devices) is illustrated in Fig. 17. However, Table 6 shows the comparison in terms of number of TS, bandwidth and power.

6 Conclusion

In this paper, a new cooperative joint network and channel coding strategy for MTC devices in the multicast settings is proposed. In this scheme, three or more devices dynamically form a cluster to disseminate messages between them. Specifically, a coding scheme for MTC devices in proximity to exchange messages via a nearby low cost relay is proposed. The key components of the proposed scheme are the use of PNC in the first phase and the fact that each MTC device removes its own message in the second phase. Additionally, the core idea of the scheme is to increase the spectral efficiency of the channel by exploiting the interference due to the fact that more than one MTC devices transmitting non-orthogonally to the end MTC device. The useful interference is strongly coded to recuperate the erased symbols in the received composite signals, and therefore, the key to successful decoding of messages remains in the ability to first recover the erased symbols, then the other message is decoded interference free. Furthermore, a systematic approach to extend the scheme to any N MTC devices by employing the concept of clustering is proposed. Messages are disseminated first within the basic clusters, then spread

out from one layer of logical cluster to another until the last logical layer. Then desired messages within each logical layer are sent from higher logical layer to lower logical layer until the basic clusters.

Additionally, the performance of the proposed scheme using practical Raptor codes with two relaying schemes namely AF and DNF was evaluated. Particularly, it was shown that with very little processing at the relay using DNF relaying strategy, performance can be enhanced. In the absence of noise, simulation results showed that a very small overhead is required to fully resolve the messages and hence this represents a small fraction of sum rate loss. Therefore, a sum rate of 1.952 is achievable. Whereas in noisy scenario, simulation results showed that the performance degrades and requires additional overhead to compensate for the errors due to noise at all nodes. Furthermore, results show that the sum rate increases linearly at low SNR then it saturates close to the upper bound at higher SNR. Moreover, the overhead required at each MTC device to successfully decode intended messages was evaluated. Additionally, the performance of the proposed scheme with functional decode and forward and the traditional schemes was compared.

References

1. K. Ashton, "That 'Internet of Things' thing in the real world, things matter more than ideas," *RFID J.*, Jun. 2009 URL <http://www.rfidjournal.com/article/print/4986>.
2. Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014-2019", white paper, Feb 2015.
3. Huawei Technologies Co., "Global Connectivity Index", September 2015.
4. M. James, M. Chui, P. Bisson, J. Woetzel, R. Dobbs, J. Bughin, and D. Aharon. "The Internet of Things: Mapping the Value Beyond the Hype." McKinsey Global Institute, June 2015.
5. M. Dohler and C. Anton-Haro, "Machine-to-Machine (M2 M) communications – Architecture, performance and applications", Woodhead Publishing, January 2015.
6. ETSI TR 102 935 V2.1.1, "Machine-to-Machine communications (M2 M): Applicability of M2 M architecture to Smart Grid Networks; Impact of Smart Grids on M2 M platform", Technical Report, 2012-09.
7. A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities", in *Internet of Things Journal*, IEEE, vol. 1, no. 1, pp. 22–32, Feb. 2014.
8. N. Skeledzija, J. Cesic, E. Koco, V. Bachler, H. N. Vucemilo, H. Dzapov, "Smart home automation system for energy efficient housing," in *Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2014 37th International Convention on, vol., no., pp. 166–171, 26-30 May 2014.
9. K. Chen, "Machine-to-Machine Communications for Healthcare", *Journal of Computing Science and Eng.*, Vol. 6, No. 2, pp. 119–126, June 2012.
10. R. Leeb, L. Tonin, M. Rohm, L. Desideri, T. Carlson, and J.D.R. Millan, "Towards Independence: A BCI Telepresence Robot for People With Severe Motor Disabilities," in *Proceedings of the IEEE*, vol. 103, no. 6, pp. 969–982, June 2015.
11. Z. Yongjun, Z. Xueli, Z. Shuxian and G. shenghui, "Intelligent transportation system based on Internet of Things", in *World Automation Congress (WAC)*, 2012, vol., no., pp. 1–3, 24-28 June 2012.

12. L. Zhang, J. Liu, and H. Jiang, "Energy-efficient location tracking with smartphones for IoT," in *Sensors*, 2012 IEEE, vol., no., pp. 1–4, 28-31 Oct. 2012.
13. G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. Johnson, "M2 M: From mobile to embedded Internet", *IEEE Communications Magazine*, vol. 49, no. 4, pp. 36–43, Apr. 2011.
14. I. Stojmenovic, "Machine-to-Machine Communications With In-Network Data Aggregation, Processing, and Actuation for Large-Scale Cyber-Physical Systems", in *Internet of Things Journal*, IEEE, vol. 1, no. 2, pp. 122–128, April 2014.
15. C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey", *IEEE Communication Surveys Tutorials.*, vol. 16, no. 1, pp. 414–454, First Quarter 2014.
16. H. Tschofenig, et al., "Architectural Considerations in Smart Object Networking", Tech. no. RFC 7452. Internet Architecture Board, Mar. 2015.
17. F. Ghavimi, and C. Hsiao-Hwa, "M2 M Communications in 3GPP LTE/LTE-A Networks: Architectures, Service Requirements, Challenges, and Applications", in *Communications Surveys & Tutorials*, IEEE, vol. 17, no. 2, pp. 525–549, Second quarter 2015.
18. <http://www.bluetooth.com>.
19. <http://www.zigbee.org>.
20. <http://www.z-wave.com>.
21. M. Shirvanimoghaddam, Y. Li, M. Dohler, B. Vucetic, and S. Feng, "Probabilistic Rateless Multiple Access for Machine-to-Machine Communication", in *Wireless Communications*, *IEEE Transactions on*, vol., no. 99, pp. 1–12, 2015.
22. A. Aijaz, and A.H. Aghvami, "Cognitive Machine-to-Machine Communications for Internet-of-Things: A Protocol Stack Perspective", in *Internet of Things Journal*, IEEE, vol. 2, no. 2, pp. 103–112, April 2015.
23. B. W. Khoueiry and M. R. Soleymani, "Destination cooperation in interference channels," in *Proc. IEEE ICCE*, Las Vegas, USA, Jan. 2012.
24. B. W. Khoueiry and M. R. Soleymani, "A novel destination cooperation scheme in interference channels," in *Proc. IEEE VTC 2014*, Vancouver, Canada, Sep. 2014.
25. B. W. Khoueiry and M. R. Soleymani, "A novel coding strategy for device-to-device communications," in *Consumer Communications and Networking Conference (CCNC)*, 2015 12th Annual IEEE, vol., no., pp. 200–205, 9-12 Jan. 2015.
26. T. M. Cover and J. A. Thomas, "Elements of Information Theory", 2006 Wiley-Interscience.
27. A. Carleial, "A case where interference does not reduce capacity", *IEEE Transactions on Information Theory*, vol. 21, no. 5, pp. 569,570, Sep 1975.
28. P. Popovski and H. Yomo, "The anti-packets can increase the achievable throughput of a wireless multi-hop network", in *Proc. IEEE ICC*, Istanbul, Turkey, June 2006, pp. 3885–3890.
29. T. Koike-Akino, P. Popovski, and V. Tarokh, "Optimized constellations for two-way wireless relaying with physical network coding", *Selected Areas in Communications*, *IEEE Journal on*, Vol. 27, No. 5, pp. 773–787 2009.
30. J.H. Sorensen, R. Krigslund, P. Popovski, T. Akino, and T. Larsen, "Physical layer network coding for FSK systems", *IEEE Communication Letters*, vol. 13, no. 8, Aug. 2009.
31. R. Chang, S. Lin, and W. Chung, "Joint-denoise-and-forward protocol for multi-way relay networks", *Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1– 4 2013.
32. J. Proakis and M. Salehi, "Digital Communications", McGraw-Hill, New York, USA, 5th edition, 2008.
33. R. Ahlswede, N. Cai, S. R. Li and R. W. Yeung, "Network information flow", *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204 –1216, 2000.
34. D. S. Lun, M. Médard, and R. Koetter, "Efficient operation of wireless packet networks using network coding", in *Proceedings International Workshop on Convergent Technologies (IWCT)*, 2005.

35. S. Zhang, S. C. Liew, P. P. Lam, "Hot Topic: Physical-layer Network Coding", ACM Proceedings of the 12th annual international conference on Mobile computing and networking, pp. 358–365, 2006.
36. B. Nazer and M. Gastpar, "Computing over multiple-access channels with connections to wireless network coding", Proceedings. IEEE International Symposium Information Theory, pp. 1354–1358 2006.
37. B. Nazer and M. Gastpar, "Reliable physical layer network coding" Proceedings. IEEE, pp. 438–460 2011.
38. H. V. Poor, "An Introduction to Signal Detection and Estimation", 2nd Edition, Springer, 1998.
39. A. Shokrollahi, "Raptor codes", IEEE Transactions on Information Theory, vol. 52, no. 6, pp. 2551–2567, 2006.
40. O. Etesami and A. Shokrollahi, "Raptor codes on binary memoryless symmetric channels", IEEE Transactions on Information Theory, vol. 52, no. 5, pp. 2033–2051, 2006.
41. J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior", IEEE Transactions on Information Theory, vol. 51, no. 12, pp. 3062–3080, December 2004.
42. M. Hasan and E. Hossain, "Resource allocation for network-integrated device-to-device communications using smart relays", in Proceedings IEEE Globecom workshop, 2013, pp. 597–602.
43. A. Shokrollahi, "LDPC Codes: An Introduction, in Coding, Cryptography and Combinatorics", Computer Science & Applied Logic, pp. 85–110, 2004.
44. M. Luby, "LT codes", in Proc. 43rd. Annual IEEE Symposium Foundations Computer Sciences, Vancouver, BC, Canada, pp. 271–280, Nov. 2002.
45. S. Katti, H. Rahul, H. Wenjun, D. Katabi, M. Medard, J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding", in Networking, IEEE/ACM Transactions on, vol. 16, no. 3, pp. 497–510, June 2008.
46. S. Ong, S. Johnson, and C. Kellett, "An optimal coding strategy for the binary multi-way relay channel", IEEE Communications Letters, vol. 14, no. 4, pp. 330–332, 2010.

Energy-Efficient Network Architecture for IoT Applications

P. Sarwesh, N. Shekar V. Shet and K. Chandrasekaran

Abstract Internet of Things is an emerging technology, which connects smart devices with internet and allows to share their data globally. The major goal of IoT technology is to improve the resource utilization. In IoT network the edge devices (sensors, RFID, actuators, Bio-Chip, etc.) are operated by battery power and they are connected with low power links such as IEEE 802.15.4, IEEE 802.11, etc., which makes IoT network as energy constrained. Designing energy efficient network architecture is a greatest challenge for energy constrained IoT Networks. The network is said as energy efficient, based on its network lifetime. In this chapter, effective combination of two different techniques such as node placement technique and routing technique is proposed in single network architecture, to improve the lifetime of IoT networks. In node placement technique, uneven data traffic is addressed by introducing hierarchical node placement. In routing technique, uneven energy consumption is addressed by residual energy based path computation. Splitting the energy related parameters by two different techniques (node placement technique and routing technique) highly reduces the complexity of network. Our result says, effective combination of node placement technique and routing technique improves the uniform energy consumption and provides better network lifetime.

P. Sarwesh (✉) · N.S.V. Shet · K. Chandrasekaran
National Institute of Technology, Karnataka, Mangalore, India
e-mail: sarweshpj@gmail.com

N.S.V. Shet
e-mail: shekar_shet@yahoo.com

K. Chandrasekaran
e-mail: kchnitk@gmail.com

1 Introduction

Internet of Things is a smart technology that holds a number of intelligent devices in a single network platform (global network infrastructure). Internet of Things (IoT) converges physical objects with cyber system, to utilize natural resource in smarter way. Huge amount of data created by nature or physical objects can be captured, processed, communicated and stored by the help of IoT technology, which can be utilized by present as well as future generations. Expanding the smart device (sensors, actuators, RFID, bio-chip, etc.) network and accessing them through internet, involves various technologies. According to research view, IoT is the combination of technologies such as, communication, networking, data processing, cloud computing, embedded systems, smart device etc. Smart device is a “Thing” in Internet of “Things”, it works autonomously with its basic capabilities such as, sensing, actuation, computation and communication [1]. The major challenge for IoT network is utilizing power in efficient way, maintaining energy efficiency without affecting the reliability and quality of service (QoS) is the main aim of IoT technology [2–5]. IoT network is resource constrained network, because devices used for IoT applications are low power (battery operated). Network that provides reliable data transfer between low power devices and low power links can be considered as suitable network for IoT applications. ITU vision statement for IoT is, “anytime, anywhere, anything can be connected with each other”, battery operated smart devices can bring this statement as real time scenario [2]. Energy should be considered as valuable resource for battery operated applications, frequent battery replacement in remote area is difficult task, it affects customer as well as service provider. Non uniform energy consumption reduces the network lifetime, where lifetime affects the cost of the network. Hence suitable energy consumption mechanism is required for IoT network.

In IoT and WSN network data traffic is many to one, one to one and one too many, in most of the applications many to one (nodes to base station) traffic pattern is used [6–9]. Due to many to one traffic, nodes near to sink drain out its battery soon, because these nodes carry their own data and forwards data generated by other nodes, this leads to quick energy depletion and quick node death in short span of time. This problem is named as energy hole problem, it stops entire network communication, nodes near to the sink drain out its battery due to data overload, data transmission from nodes to the sink stops and network need to be reconfigured, energy hole problem affects the network lifetime severely. In research papers its mentioned that energy is not utilized completely (nodes far from the sink cannot utilize its whole energy) due to network disconnects. Many techniques such as, Node placement, Routing, MAC layer optimization, etc. [10, 11] are proposed to solve the non-uniform energy drainage problem. Some of techniques to improve the energy efficiency are described in the following sub-sections.

1.1 Node Placement

Placing nodes strategically in sensor field is one of the important solutions for non-uniform energy drainage problem. Node placement is a effective optimization technique for resource constrained networks. Every node in WSN and IoT network has its own radio frequency range, so connectivity and coverage area are the important aspects of IoT network. Effective node placement provides energy efficiency and prolongs network life time; many graph based algorithms are used for optimizing the node placement technique [12]. The main aim of node placement technique is to minimize density of nodes (number of nodes) with better connectivity [13]. Node placement technique concentrates on maximum coverage with minimum energy cost.

1.2 Routing Technique

Routing mechanism is used to find the optimum path (Energy Efficient, Reliable, QoS Aware, Secure etc.) for sending data from source to destination. Routing process happens between sensor to base station, sensor to sensor, base station to sensor, all three kind of traffics follows same procedure, initially source send request packet (control message) to destination, based on parameter information, destination finds the suitable path and sends replay packet (acknowledgement control message) to source [14]. This process is referred as flooding, to find energy efficient path flooding is required, when flooding exceeds it reduces energy efficiency, Hence there is severe tradeoff between energy efficiency and flooding, wide variety of routing techniques (flat based, hierarchical based, adaptive based) are emerging to balance the tradeoff. Routing is one of the important technique to achieve energy efficiency and network life time. Efficient routing protocol avoids over flooding, maximum retransmission, minimum residual energy path, non-reliable path and maintains energy efficiency and reliability in the network [4, 15–17].

1.3 MAC Layer

Media Access Control (MAC) protocol is responsible for node access to the communication medium, sharing the medium in effective way increases the energy efficiency. In MAC two types of approaches are followed to access the medium one is centralized approach and other is distributed approach, in centralized approach master node (base station) grants permission to sensor nodes to access the media, where as in distributed approach sensor nodes will dynamically access the media based on neighbour nodes information. The MAC layer protocols are classified into

four types, Contention-based, Round-Robin, Reservation-based and Channelization-based. The main function of these techniques is to prevent collisions and reduce the error in communication media, if collision and error are avoided; network can be said as energy efficient and reliable network. MAC protocols plays great role in improving energy efficiency [18–20].

The above techniques which importance of energy efficiency is discussed and literature says many standard algorithms concentrate on particular parameter (residual energy or link quality or traffic, etc.), they are not concentrating multiple parameters together. If multiple parameters are handled by single standard protocol, which has energy efficient feature with reliability and QoS, it can be said as suitable protocol for IoT as well as WSN applications (small scale and large scale). In further sections need of energy efficiency and suitable techniques for IoT network is elaborated.

2 Resource Constrained Nature of IoT Network

Most of the smart devices used for IoT application are Low Power devices, which are operated by battery power. Power consumption, physical size and cost are the major challenge while designing a smart device. Bandwidth, transmission range and power (transmission power and reception power) are the major challenges while assigning network links to low power devices. The phrase “resource constrained” refers low power links and battery operated devices. In most of monitoring applications (Earth quake, Land slide, forest fire, battle field etc.) often battery replacement is difficult task, providing power through regular power line is impossible in harsh environment.

In low power radio communication, chance of data loss is high, retransmission increases when data loss is high, which drastically reduces battery power, many research paper describes 70 to 80 % of energy is utilized for communication purpose. Processing unit consumes more energy when compared to sensing unit. Sensed data need to be processed and redundant data should be avoided with the help of data fusion. Huge amount of energy is consumed by communication and processing unit. Consumer in any kind of application needs low cost smart device. The challenging task for service provider is to design low cost device with good processing capability and compact size. Both consumer and service provider require better network life time. Sever tradeoff occurs between implementation cost and network lifetime, which majorly depends on energy consumption. Hence IoT network is smart as well as resource constrained. The following table describes the features of Internet and IoT network.

Table 1 describes the features of IoT network and Internet, from the above description it is clearly understood that IoT network is constrained in nature.

Table 1 Features of Internet and Internet of Things network

Features	Internet	IoT network
Nodes	Routers	Sensors/Actuators, Bio-chip etc.
Links	High power and stable links	Low power and unstable links
Nature of device	Non-constrained device	Constrained device (Limited in battery)
Address	Internet protocol address	Internet protocol address
Routing	Non-application aware routing	Application aware routing
Power source	Main grid power	Battery (Most of the applications)

2.1 Device Level Power Constraints

Smart device is the “thing” in Internet of “Things”. It is equipped with components such as sensor (sensing unit), microprocessor and signal processing board (processing unit), transmission and reception antenna (Communication Unit) and Battery (Power unit). Cost of the smart device is more important in large scale implementation like IoT applications. Physical size of the smart object should be small, which is the highly expected from end users [1]. While designing smart device, the major goal of smart device technology is to provide better device life-time without compromising the size and cost of the device. So battery utilization is a critical challenge for smart device. Power constraints in smart device leads to development of hardware and software. The following table describes the smart devices used for IoT and WSN applications (Table 2).

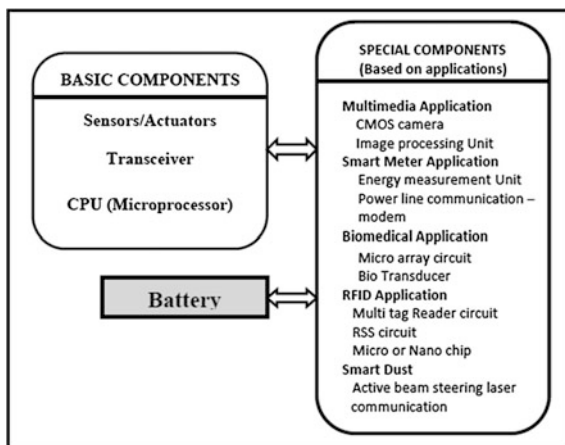
(a) Hardware

Hardware design for a smart device should satisfy cost and size with better device life time. In hardware components, current leakage is minimized to improve the power efficiency. Processor plays a important role in hardware components, power consumption of hardware components depends on the processor speed. Mother board in smart devices is designed in such a way that, it should allow some components to sleep mode, to minimize the energy consumption [1]. Efficient low power amplifiers should be implemented with respect to application specifications. These techniques helps smart devices to improve its energy efficiency.

Table 2 Features of smart device

Device	Signal type	Applications	Power source
RFID tag	Low power radio	Tracking objects, etc.	Battery
Multimedia sensor	Low power radio	Wildlife monitoring, etc.	Battery
Bio-chip	Low power radio	Healthcare monitoring, etc.	Battery
Sensors	Low power radio	Household, industry, etc.	Battery
Smart dust	Low power radio	Military applications, etc.	Battery

Fig. 1 Structure of smart device



(b) *Software*

Efficient utilization of hardware components depends on software design, software limits battery usage by controlling hardware components, it makes hardware components to sleep mode, when they are in idle state. The major role in software is taken care by operating system, when sensor has no event to detect operating system turns hardware components into low power operation mode, the drawback of the sleep mode is it affects the latency of the system. Hardware and software keeps track the information of energy spent and it provides the information to other layers (data link layer and network layer). And also hardware and software are responsible for computational and communication complexity, computational speed and idle mode of hardware component highly influence the energy consumption (Fig. 1).

Designing smart device for resource-scarce environment must be better in power efficiency and it is greatest challenge for hardware and software developers. In present scenario most of smart devices used for commercial and other applications are battery operated (energy constrained), the following table describes the list of smart devices and their constraints [1].

2.2 *Network Level Power Constraints*

IoT network is unstable in nature, because it uses low power radio to communicate remote devices. In other networks (Internet, LAN, WAN) Ethernet links and SONET/SDH links are used to communicate devices, which are highly stable in nature, because routers in internet are operated by main power supply. In IoT network low power radio links (IEEE 802.15.4, IEEE 802.11) are used to interconnect routers and other IoT devices, which are battery powered (remote area

applications, they highly constrained in nature. These feature results in high loss rates, low data rates, and instability. Due to fairly unstable nature of links (lossy nature), there is chance of heavy packet loss and link may be disconnected, due to many reasons, such as interference, weak signal strength, etc. Devices in remote area (for example forest fire monitoring) may be scattered in a unplanned way and they are small in size and limited in battery. For battery operated devices, low power radio (for example IEEE 802.15.4) is more suitable, which offers only low duty cycle, so it is necessary to monitor the energy level for devices [1].

The neighbor discovery process exhibits high energy consumption, so necessary steps are required to minimize it. Energy is one of the important issues in resource constrained network. If energy is not consumed in efficient way, it leads to node death and node death severely affects the lifetime of entire network. The resource constrained network is characterized by irregular data traffic and non uniform distribution of nodes, which results in huge packet drop. Some of nodes in LLNs have more neighbors than other nodes; Hence data flow in particular nodes will be more, which leads to energy depletion (nodes drain out), in many nodes and network will be disconnected in particular area. Hence prevent measures should be taken care for load imbalance condition. Nodes in resource constrained network suffer with unreliability problems, the device in this network cannot adapt other links quickly because of frequent variations in link quality and instability, So concentrating on link quality for IoT network is very important. If link quality is power, packet loss may occur, which leads to increase in retransmissions, when retransmission increases energy consumption. Hence Reliability directly affects the energy efficiency in low power networks. Dynamic change in link quality will affect the network connectivity, if problem arises in connectivity, entire network will be failure network. To solve the above issues the efficient protocol design is needed. Smart device network is smart as well as lossy, these condition leads to development of various routing protocols.

3 Common Factors that Influence Energy Efficiency

In wireless network there are several issues that highly influence the energy efficiency and performance of the network. These issues need to be addressed by efficient protocol design and network architecture. The following describes about, the major factor that affects life time and performance of the network [21].

3.1 Collisions

Collision is a major problem in wireless network; it affects the performance and lifetime of the network. During the data transmission from transmitter node and receiver node in particular channel, new entry of transmission signal from another

node in same channel leads to collision. Due to collision transmitter and receiver re-establishes the connection for packet re-transmission, this leads to energy wastage and also increases the congestion. Highly affected parameters by collision are latency and energy. Possibility of collision is more in carrier-sense multiple access technique (CSMA) protocols, because If node X and node Z want to transmit data to node Y. Node X can sense Y and node Z can sense Y, but X and Z cannot sense each other, these leads to collision. In research field necessary steps are taken to avoid collision. One of the suitable technique to avoid collision is TDMA (time division multiple access). In TDMA based schemes time is subdivided into time slots and particular time slot is allotted to each node for accessing the channel. In some application scenario, events may happen rarely, but TDMA allocation will occur continuously, which increase control overhead and energy consumption. In such cases CSMA-type protocols are more suitable [22].

3.2 Overhearing

Overhearing occurs when a node receives unwanted packets, which are not destined for it. In unicast communication, receiver of all the nodes is in active state. To know the information about current traffic, but this leads to more energy consumption. In low power network scenario, cost of receiver energy is more valuable, multiple receiver receives control packet which are not destined for it, this leads to unnecessary power loss. In TDMA-based schemes, occurrence of over-hearing is very less, where as in CDMA-based schemes overhearing need to be taken care. Necessary steps are required to control overhearing in resource constrained applications. Efficient decoding techniques are suitable solution for overhearing issue, if receiver decodes the packet header and determines whether it is the intended recipient, wastage of energy will be reduced. Overhearing is tolerable in small scale applications, where as in large scale applications it severely affects the energy efficiency of the network [21].

3.3 Protocol Overhead

Additional information (control packets) required for particular protocol, to establish connection and communication is said to be packet overhead. For example MAC (IEEE 802.11) requires RTS and CTS for establishing connection, AODV requires RREQ and RREP for establishing communication. This additional information (overhead) consumes energy, these information helps for better connectivity and to find energy efficient path. So overhead is needed and it should be limited in such a way that it should not affect energy efficiency of the network. Implementing suitable addressing scheme with respect to physical, MAC and network protocol can minimize the packet overhead significantly [21].

3.4 *Idle Listening*

Idle listening is the major issue need to be addressed in low power networks. If a node is in active state (ready to receive) without receiving any data is referred as idle listening. In monitoring applications sensor nodes spend long time in idle state, where as receiver board requires significant amount of energy at this state, this leads to power loss. In event detection applications, most of the time sensor node waits for incoming packets. In some cases nodes will be allowed to be in sleep state and wake up in periodic manner to listen incoming packets. The time taken to switch wakeup mode between one node to other node is said to be duty cycle, switching nodes to different states also requires significant amount of energy. In this mechanism packet should reach the receiver in receivers listening times, else packet loss occurs. Hence idle listening is the big challenge in low power radio scenarios. Efficient framework with the help of beacon packets can bring optimum solution for this problem [21].

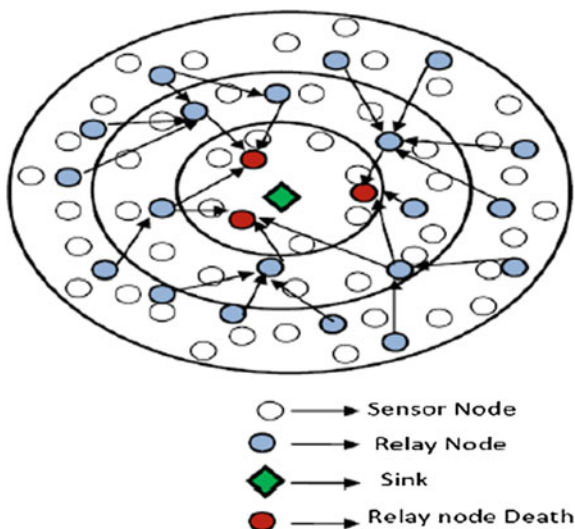
3.5 *Energy Hole (Network Overload)*

The nodes which are near to the base station carries huge amount of data traffic when compared to other nodes, this leads to uneven energy consumption and quick node drain out this is referred as energy hole problem. Nodes near to base station carry their own data and forwarded data of other nodes, these leads to huge overload and quick node death, when all the nodes near base station run out its power, communication to the base station stops and entire network need to be reinstalled [23]. This problem severely affects the performance of the network; preventive measures can be taken by help of efficient routing mechanism.

In the Fig. 2, red circle indicates quick node drain out, dude to node over load and uneven power consumption [4].

3.6 *Interference*

Interference is a unwanted signal that destroys the original signal. Interference can cause due to different reasons such as, multiple radio sharing the bandwidth on single channel, environmental conditions, noise, etc. Assigning default value to each radio set may cause severe interference, because chance of interference is more, when multiple radio share the bandwidth on single channel. In some cases hidden nodes produce cyclic redundancy check (CRC) code errors, which leads to interference. Hidden nodes are referred as the nodes that are out of range of other nodes. Interference that occurs due to channel overlaps is referred as co-channel interference or adjacent channel interference. The other devices which use same set

Fig. 2 Energy hole problem

of frequency, which is used by nodes will also cause interference. In low power radio network interference severely affects energy efficiency and network performance, because the chance of bit error rate (BER) is more in unstable links (low power links), so necessary steps are required to avoid interference in resource constrained network [4].

These factors highly disturbers energy efficiency and performance of the network, to avoid these issues preventive steps are taken in research field. In high power networks (Internet) these issues may be tolerable. In low power networks such as WSN and IoT, it severely affects the life time of the network. Hence efficient physical, data link and network layer protocols are highly needed to resolve these problems.

4 Energy Efficient Techniques for IoT Network

The huge demand for Internet of Things and wireless sensor networks, leads to rapid development in MEMS technology and protocol design. MEMS technologies has brought the concept of low power and low cost device, which has basic sensing, processing, and communicating capabilities. In large scale as well small scale applications IoT became the promising technology. Extensive effort is taken for every layer in the IoT network model such as, node placement, routing, media access control, TCP-IP, domain-specific application design etc. In IoT architecture nodes are defined as smart devices and resource constrained routers, were as in Internet, nodes are defined as computers and main-powered routers, which consist of good memory capacity such as, extensive flash memory and powerful CPU

interconnects with stable links (main powered links). But for IoT network (smart device network or resource constrained network) fairly unstable links are used with IP connection. So IoT network comes between the category of wireless sensor networks and Internet. It is concluded that energy efficient and reliable network mechanisms are highly required for IoT applications. The following techniques describes the energy efficient mechanisms to prolong the lifetime of IoT network.

4.1 Node Placement Technique

Node placement is the effective technique to achieve uniform energy consumption in low power network. An efficient node placement meets the requirements such as, energy efficiency, cost of the device, connectivity, density of nodes, lifetime and latency. The major goal of node placement technique is to prolong network lifetime and improves the network connectivity. In remote applications resource constrained nodes are deployed in unattended environments, it operates autonomously with the help of onboard radio, so onboard transmission range and position of nodes need to be assigned by the help of node placement technique [12]. In most of the deployment strategies node placement technique is proven as NP-Hard problem, to solve that problem several algorithms such as, biological inspired algorithms, Particle swarm optimization (PSO)-based algorithms, virtual force directed co-evolutionary PSO (VFCPSO), genetic algorithm (GA), artificial bee colony (ABC), optimized artificial fish swarm algorithm (OAFSA), territorial predator scent marking algorithm (TPSMA), multi-objective optimization (MOO), etc.

Node placement techniques is the well know technique for prolonging the network lifetime In node placement technique, non uniform node density causes severe bottlenecks such as, unbalanced data traffic and network overload. In some cases uniform distribution of nodes may also lead to energy hole, because the nodes which are near to base station consumes more energy when compared to nodes far from base station, because nodes near to base station, carry their own data as well forwarded data of other nodes. In [24, 25] authors investigated maximum lifetime of nodes network with coverage constraints. In this work average energy consumption per data collection round is considered as lifetime metric, so the load to nodes are spread in balanced way and nodes have been relocated based on energy level of nodes. In node placement scheme node density severely affects the lifetime of network, in [26, 27] authors studied the effect of node density on network lifetime, in this work network life time per unit cost is formulated analytically for one dimensional placement scenario [12, 13]. This paper interrelates node density with cost and lifetime of network, as first step authors of this paper consider multi-variant non-linear problem and solved it numerically, as second step density of node is minimized by help of analytical solutions, to maximize the network solution. Optimal spacing is also one of the efficient technique to reduce the density of the network, it ensures density of nodes as well network connectivity [24–27].

Table 3 Role of nodes

Role of the node	Sensor	Relay	Base station
Sensing	Yes	No	No
Path computation	No	Yes	Yes
Data processing	Yes	Yes	Yes
Transmitting	Yes	Yes	Yes
Receiving	Yes	Yes	Yes
Power constraints	Yes	Yes	Yes

Life time of the network can be optimized by deploying nodes, which does different role in network architecture. Nodes can be categorized to sensor, relay, cluster head or base-station. Sensor nodes are responsible for sensing, computation (processing) and communication (transmission and reception). Relay nodes are responsible for computation and communication. Sensor does sensing (collect the environmental data), processing the sensed data and transmits the data to base station. Relay collects the data from sensors and other relays, after data aggregation, relay node transmits the data to base station. Base station collects information from sensors or relays and sends the data to storage device, sensors and relays can be limited by energy, where as base station cannot be limited by energy. If forwarding process and sensing process are split, complexity of the device will be reduced, where sensor will be free from forwarding process and relays will be free from sensing process. Hence assigning various roles to node in network architecture improves network performance and energy efficiency of the network. Literature describes, sensor and relay combination network architecture gives better network lifetime. The following table describes the role of nodes in heterogeneous scenario [12, 28–30].

Table 3, describe role of sensor, relays and base station, which are necessary information required to develop node placement technique. Efficient node placement can bring the unified design and better network life time in IoT network. By implementing node placement scheme number of nodes can be minimized, which is required for IoT network.

4.2 Routing Technique

Routing is the one of the efficient mechanism to reduce the energy consumption, in any kind of network. Most of the power consumption problems are solved by routing framework. In grid powered network (Internet), most of routing mechanism are designed to govern reliability and Quality of Service, where as in resource constrained network, routing mechanism are mainly designed to improve the power efficiency. Major amount of power consumption occurs in communication unit, when power consumed communication unit is minimized, energy efficiency can be improved with better network life time [14, 15, 31]. Routing plays great role in minimizing power consumption in communication unit, it finds energy efficient

path to send the data, when data is forwarded through energy efficient path, data loss can be prevented, data retransmission can be prevented and quick node death can be prevented. There are several energy efficient routing techniques (protocols) developed for optimizing energy efficiency. In any kind of low power network, routing protocols are broadly classified into six categories. The following elaborates the categories of routing protocol [16, 17, 32, 33].

(a) *Attribute-based protocols*

Attribute-based protocols handle routing mechanism based on content of the packet, this approach is not device specific; these are the major benefits of these protocols. This can also be defined as data centric routing approaches. Since nodes within the network are aware of routing mechanism and the nodes can take their decision, to forward the packet or to drop the packet. This type of network analyze the content of data in every hop. These are some of routing protocols that works based on attribute-based approach, directed diffusion, energy-aware data-centric routing, constrained shortest-path energy-aware routing, rumor, etc. [4]. In this framework details of routing metric is fetched in packet, raw data is aggregated and compared with other sensor data, necessary decisions (either to forward or drop the packet) are taken by the node, then data is forwarded in energy efficient path. Hence it improves the energy efficiency.

(b) *Flat protocols*

This types of protocols are designed for efficient data forwarding in large-scale dense networks. In flat based network large number of nodes work together to collect environmental data and forward it to the destination. In flat based approach all the nodes are similar to each other (same in node configurations). Some of the well know routing protocols used in flat based network are gradient broadcast, sequential assignment routing, minimum cost forwarding algorithm etc. These protocols evaluates cost of neighbour devices, the cost may be signal strength, energy level, etc., if the cost of node is better when compared to its neighbour node, it takes response for forwarding the data, this reduces power consumption in the network [4].

(c) *Geographical routing*

The reason behind the development of geographic routing is to reduce the overhead of the packets. In this technique location aware devices are used to find the location of nodes in prior and forwards the data packets towards the destination. In geo-graphical routing, neighbour nodes exchange the location information and transmit data to the node, which is close to the destination. This type of protocol performs better localized interactions. Protocols used in geographical routing are stateless protocol for soft real time communication, geographic routing with no location information and geographic routing with limited information in sensor networks. This type of protocols, reduces the packet overhead, which increases the energy efficiency of the network [4].

(d) *Hierarchical protocols*

Hierarchical routing protocols provides better efficiency to constrained network environment, the nodes which operate in this type of network performs different roles, sensor node senses the environmental information and transmits to cluster-heads, where cluster-head forwards the collected information to base station. Hence this type of hierarchical structure balances the energy consumption, because entire network area is divided into clusters, these clusters operate based on node functionality. Some of popular hierarchical protocols are low-energy adaptive clustering hierarchy, power-efficient gathering in sensor information system, threshold sensitive energy-efficient sensor network protocol etc. [4].

(e) *Multipath routing*

Multipath routing technique is developed to avoid route failures and packet re-transmissions. Single path routing protocols need to generate additional control packets in case of link failure, which leads to wastage of energy, where as in multipath routing multiple paths are computed from source to destination. In case of route failure, it sends data through alternative links, hence packet loss is highly prevented, which is directly related to energy efficiency. The following are basic protocols of multipath routing technique, meshed multipath routing, energy efficient multipath routing in wireless sensor networks, reinForM etc. [4].

(f) *QoS-based protocols*

Routing protocols which were discussed earlier describes routing metrics such as, cost of node, control overhead, residual energy etc. So improving energy efficiency with better QoS facilitate the performance of the network. Basic QoS assured routing protocols are stream-enabled routing, algorithm for robust routing in volatile environment, etc. All these categories of protocols are developed to improve energy efficiency, reliability and quality of service in resource constrained network. In routing following parameters are estimated to achieve energy efficiency they are, residual energy, energy consumed per packet, time to network partition, cost per packet and maximum mobile cost. By using these parameters efficient path computation can be done and data can be forwarded in energy efficient path, which increases the life time of the network [4].

4.3 MAC Based Optimization Technique

Media access control (MAC) layer based optimization is one of the efficient techniques to achieve energy efficiency in low power network. MAC layer is the part of data link layer; it can be said as sub layer of MAC. MAC handles the scheduling mechanism, it schedules the transmission among the nodes and also it provides the coordination between the nodes. MAC protocol controls the mobile terminals, to achieve energy efficiency, Efficient MAC protocol satisfies the

following network requirements such as, it should be flexible during the creation of network infrastructure, it should share the wireless channel in efficient way, so that collision can be avoided and bandwidth can be utilized completely. It should be energy aware for extending the network lifetime. MAC protocol can be optimized by using following techniques, Scheduling-based mechanism, Power Control Techniques, Power off mechanism, Multi Channel Mechanism, Antenna-based mechanism [18, 21, 22].

(a) *Scheduling-based mechanism*

Major amount of energy is wasted due to collision. Collision avoidance can reduce major wastage of energy. There are several scheduling-based mechanisms developed for low power MAC, in that three common techniques are frequency-division multiple access (FDMA), time-division multiple access (TDMA) and code-division multiple access (CDMA). In FDMA frequency band is divided into several bands and nodes can access different bands at same time. In TDMA time is divided into several timeslots and nodes access same frequency in different time slots. In CDMA node can access same frequency channel at the same time with the help of unique code. In recent years several hybrid MAC protocols are developed to achieve energy efficiency such as, IEEE 802.11 standard, EC-MAC protocol, PAMAS protocol, etc. [18, 19].

(b) *Power control techniques*

Contention based MAC protocol is used in most of the MAC mechanisms, contention based mechanism works better in dynamic topology changes. Contention based mechanism is robust in nature and also it requires power for channel sensing, ACK schemes, retransmissions etc. To avoid wastage of power, efficient power control techniques are required for energy reservation in contention based schemes, some of the efficient power control mechanisms tune the transmission power level based on distance and signal strength and achieve energy efficiency. Power control techniques allow per-packet selection of transmit power and variable transmit power levels for per-packet. Some of the well known protocols used for power control mechanism are Power Control MAC (PCM), Power Controlled Multiple Access (PCMA) Protocol, Dynamic Channel Assignment with Power Control (DCA-PC), Power Controlled Dual Channel (PCDC), Common Power (COMPOW) Protocol, etc. [18, 19].

(c) *Power off mechanism*

In wireless systems, receivers will be in active mode (powered on) for entire network lifetime to detect the signals, but most of time it will be in active state without sensing signals, this state is said to be idle state, which consumes energy without receiving data (idle listening). Idle listening must be avoided, power off mechanism should be introduced to turn radio power off and allow the node to sleep state. This mechanism prolongs the network life time by allowing hardware components to sleep state. There are several techniques introduced based on power off

mechanism they are, Power-aware Multi-access Protocol with Signaling (PAMAS), MACA protocol, synchronization-MAC, Pico Node Multi-Channel MAC, Power management using multi sleep states, etc. [18, 19].

(d) *Multi channel mechanism*

Multichannel mechanism is broadly classified into two types, Multi Channel scheme and Busy tone scheme. In multi channel scheme, one control channel will be utilized to control multiple data channels, the overall bandwidth will be utilized for one control channel and multiple data channel and this avoids the disturbance or collision in contention based schemes. In this technique control channel is to resolve the contention and data channel is used to carry the data and acknowledgement. Busy tone scheme resolves the contention based issue by Introducing the busy tone, which resolves the hidden terminal problem. Protocols used in this schemes are Dynamic Channel Assignment (DCA) protocol, Dual Busy Tone Multiple Access protocol (DBTMA), etc. [18, 19].

(e) *Antenna-based mechanism*

One of the efficient way to utilize energy is antenna based scheme, the antenna based optimization scheme can reduce transmission and reception power of antennas, which are the major sources of power consumption. In antenna based scheme Omni-antenna, directional antenna (smart antenna) are more suitable for achieving power efficiency in low power wireless network [19].

The MAC layer techniques improve the power efficiency by resolving collision, ideal listening, re-transmission etc. Hence many of the MAC layer techniques are more suitable to achieve energy efficiency in resource constrained IoT network.

4.4 Transport Layer Techniques

Transport layer techniques mainly consider reliability as the optimization parameter. The efficiency of transport layer protocols depends on various parameters such as overhead costs, frequency, link failure, retransmissions, etc. These factors are directly related to energy efficiency, because if network suffers in reliability issues, it directly affects the energy efficiency. Even though there is no direct techniques to address energy efficiency in low-power transport layer protocol some of the versions of TCP is estimated with energy consumption. In transport layer protocol energy efficiency is defined as the average number of successes per transmission attempt. Hence transport layer protocol also indirectly addresses the energy efficiency of the network [34].

4.5 OS/Middleware Techniques

Integration of wireless technologies with computing technologies increases the connectivity and performance of the network. In any kind of network mobility influences the middleware and operating system. And also it is responsible for power constraints and network disconnects. In many large scale and low-power applications it is difficult to deploy expensive processor, which performs better in band-width utilization and energy efficiency. In recent research majority of applications considers, efficient utilization of bandwidth as main objective function, which directly impacts on energy efficiency. The main responsibility of operating system is to utilize the CPU usage in efficient way. In many low power applications, operating system can be designed in such a way that, it can scale down the supply voltage. One of the energy efficient technique in OS/middleware is predictive shutdown strategies. This technique describes the DRAM chips that supports different mode of operation such as active, standby, nap and power down. This features decreases 6 to 55 % of power consumption. Some of the CPU scheduling techniques also introduced efficient power consumption mechanism [34].

All the above techniques are considered as suitable energy efficient techniques for resource constrained IoT applications.

5 Suitable Network Architecture for IoT Applications

In resource constrained network, many optimization techniques are developed to prolong the network life time. In most of network architecture, optimization is carried by any particular technique (layer) for example, either routing is used or MAC based scheme is used to improve energy efficiency. In network architecture design, optimization is concentrated on particular layer, various parameters are computed together and they are implemented in some particular optimization technique, this increases the complexity of network. To increase the performance of IoT network in resource constrained environment, multiple optimization techniques should be implemented in single architecture and each technique should take care the responsibility of their own parameters. In [34] we have proposed node placement technique and routing technique in a single architecture, in our proposed method residual energy is taken care by routing (network layer). Data traffic is handled by node placement (physical layer) by means of varying the density of relay node.

5.1 Energy Efficient Network Architecture

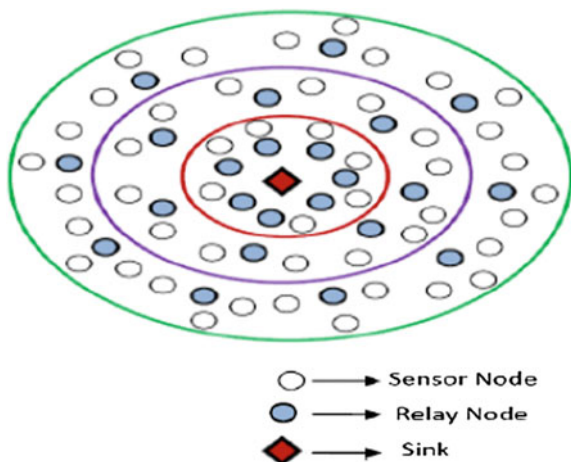
Proposed architecture consist of sensor nodes, relay nodes and base station, functions of sensor nodes and relay nodes are described in previous sections. In proposed network architecture, two methods are used to improve the energy efficiency. One is node placement and other is routing mechanism. Energy hole problem can be taken care by, efficient node placement and energy efficient routing mechanism based on residual energy. The basic idea to split the task is, to reduce the complexity of both routing protocol and node deployment. The following elaborates the proposed network architecture [34].

(a) Hierarchical node placement

In proposed node placement scheme [34], density of relay nodes are increased towards the sink, ratio of sensor and relays are varied with respect to traffic area. Relay nodes are placed one hop neighbor to sensors and relays. Sensors and relays are placed in random fashion, based on application requirements. Relay carries the data from one hop sensor neighbor and one hop relay neighbor.

In Fig. 3, red circle indicates high data traffic area, blue circle indicates medium data traffic area and green circle indicates low data traffic area. By considering the traffic area, the relay nodes are assigned to sensor nodes. The ratio of sensor and relay are described based on traffic area.

Fig. 3 Hierarchical node placement



(b) *Basic node placement assumptions*

- (1) For every sensor node, one relay node is assigned in high traffic area (red circle).
- (2) For two sensor nodes, one relay node is assigned in medium traffic area (blue circle).
- (3) For three sensor nodes, one relay node is assigned in low traffic area (green circle).

The other important reason of varying the relay node density is implementation cost, because relay nodes are high in cost (high battery capacity). Hence our proposed architecture satisfies both data traffic and network cost [34].

(c) *Routing mechanism*

In proposed architecture AODV routing protocol is used for data transmission. The reason for choosing AODV protocol is its reactive nature, no topology messages exchange is required for communication along the links, which reduces bandwidth utilization. The most important advantage of AODV is its ability to heal itself in case of node failures. It finds the shortest path from source to destination, based on the hop count [35]. For resource constrained IoT network, energy level of the node has to be considered. In proposed work residual energy is considered for route discovery process. The nodes with good energy level can be considered as intermediate nodes from source to destination. The residual energy (RE) of node is defined as,

$$RE = E_r / E_{max}$$

E_r is remaining energy of the node and E_{max} is maximum energy available in the node.

(d) *Packet format*

RREQ packet format: AODV protocol use Route Request (RREQ) packet for route discovery from source node to destination node. To implement the RE in AODV, it should be added in RREQ control packet.

Figure 4, describes the RREQ packet format with Residual Energy (RE) information.

By adding this information in control packet, AODV selects the path based Hop Count and Residual Energy.

(e) *Route selection by destination node based in RE value*

Route selection of AODV protocol is done by destination node. When the destination node receives route request, it discards further route request and starts sending the route replay to the source. In Fig. 5, (flow chart) refers the route selection procedure of destination node [36].

Fig. 4 RREQ packet format

Type	Flags	Reserved	Hop count
RREQ (broadcast) ID			
Destination IP Address			
Destination Sequence Number			
Original IP Address			
Original Sequence number			
Residual Energy			

Fig. 5 Route selection mechanism of AODV protocol

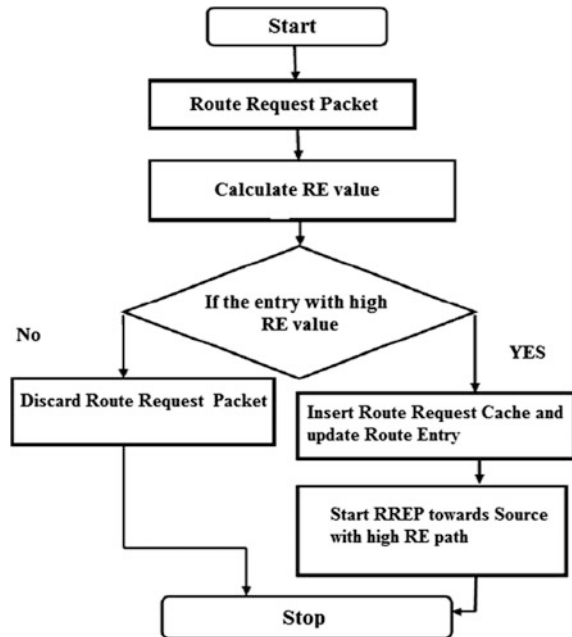


Figure 5, explains the route selection of destination node based on RE. It selects the node, which has good RE. After starting RREP timer, destination node sends reply RREP to each RREQ packet stored in cache. After data transmission it removes all the entries in the cache [35].

5.2 Performance Evaluation

The proposed network architecture is implemented in NS-2.35 simulation tool and following results are obtained.

(a) *Network lifetime*

The network is said to be energy efficient network based on its network lifetime. Balancing the energy consumption will prolong the network lifetime and prevent the network from energy hole problem. The lifetime of the network is estimated based on first death node, because when first node starts drain out its energy, within a short span of time all other nodes will drain out its energy [34].

Figure 6, describes the first node death in uniform node placement of relay nodes occur at 140th second, in proposed network architecture first node death occur at 200th second. In random placement of relay nodes, all the nodes lose their energy in 400 s. In proposed architecture, only 15 nodes losses its energy after entire simulation period. This shows, the proposed network architecture performs uniform energy consumption and gives better network lifetime.

(b) *Average energy consumption of nodes*

Energy efficiency of the network is directly related to average energy consumption of nodes. The performance and lifetime of the network depends on balanced energy consumption of nodes.

In Fig. 7, the average energy consumption of relay nodes are in balanced way (uniform). This says the proposed network architecture, gives balanced energy consumption of nodes. From above results it is understood that, the effective combination of node placement and routing mechanism gives energy efficient network.

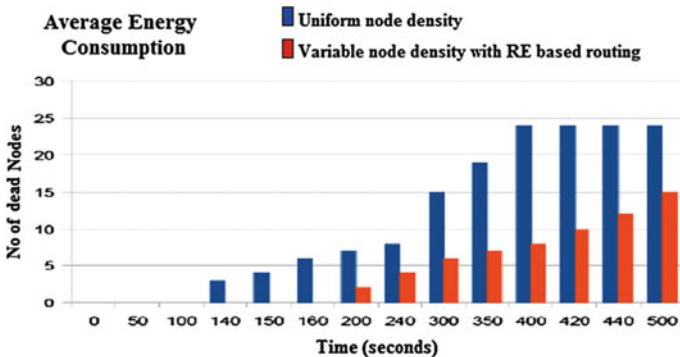


Fig. 6 Network lifetime estimation

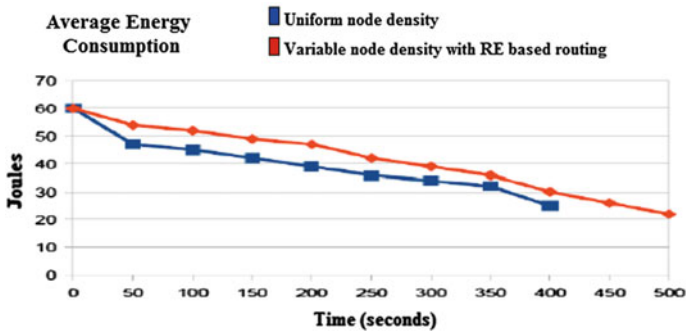


Fig. 7 Average energy consumption of nodes

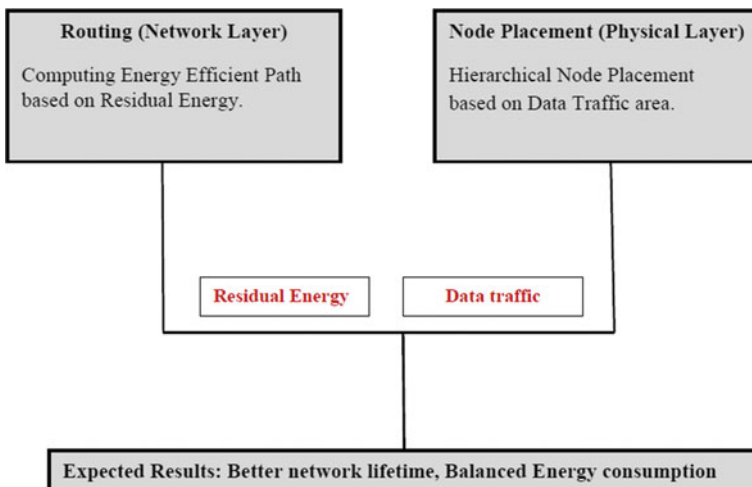


Fig. 8 Energy efficient network mechanism

The basic idea behind the proposed work is to consider major parameters for energy efficiency and reliability and to split the burden of layers (physical and network). In proposed architecture residual energy and expected transmission count are taken care by network layer by means of routing. Data traffic is handled by physical layer by means of node placement (varying the density of relay node) (Fig. 8) [34].

Hence proposed architecture will be good in load balancing, energy balancing and link quality balancing, which will be suitable for most of the IoT and WSN applications.

5.3 *Envisioned Network Architectures for Internet of Things*

Similar to the above architecture various network architectures can be proposed to improve the energy efficiency of IoT network, The following examples gives various ideas to improve the IoT network architecture.

Relation between density of node and power scheduling can be done with the help of node placement technique and MAC scheduling schemes, which can improve energy efficiency as well implementation cost of the network. Density control technique can be interrelated with scheduling scheme to increase the life time of the network and to reduce the cost of the network (Fig. 9).

Relation between data traffic and routing can be done with the help of routing technique and TCP estimation technique, which can improve the energy efficiency and reliability of the network, because TCP estimation protocol are most efficient in reliability and residual energy based protocol are better in energy efficiency. Hence interrelating transport layer protocol with routing protocol improves energy efficiency without compromising the reliability (Figs. 10 and 11).

Similarly interrelating middleware technology with Power scheduling schemes gives better performance in energy efficiency. When predictive shutdown strategies are interrelated Power Control MAC (PCM) energy efficiency can be improved in better way.

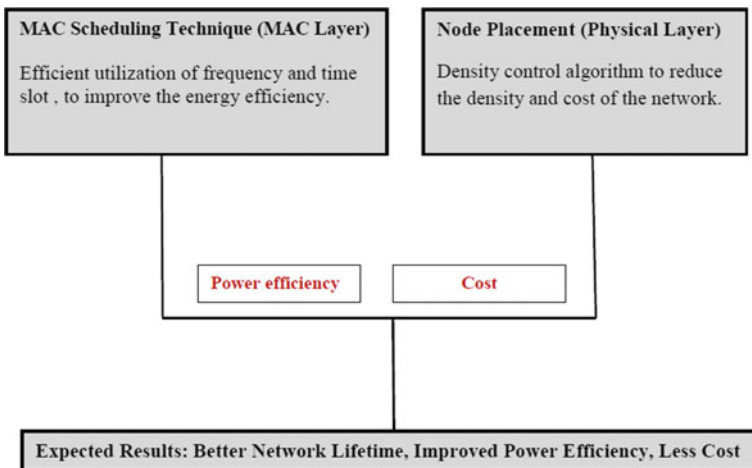


Fig. 9 Power and cost efficient network design

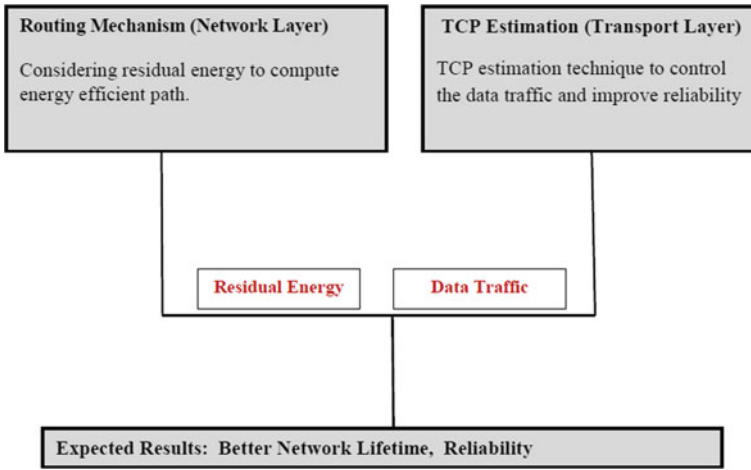


Fig. 10 Network design to improve energy efficiency and reliability

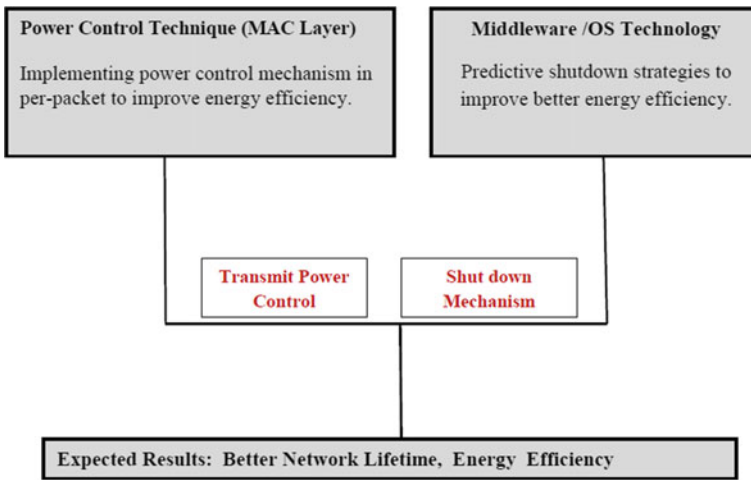


Fig. 11 Network design to prolong network lifetime

6 Conclusion

IoT has a good rate of acceptance among private as well public sectors, in gartners hype cycle IoT lies in the peak of inflated expectations. In 21st century IoT will be the promising technology that can handle wide variety applications in centralized manner. Researchers working towards the development of IoT in various aspects such as, energy management, connectivity, identification, reliability, scalability and

security etc. In applications such as smart health, smart market, smart water, smart home, smart grid etc. Devices in IoT network work autonomously with basic capabilities such as sensing, computation, communication, identification etc. Even though they are smart in working nature they are constrained by energy. The energy efficient techniques discussed in above sections are more suitable for IoT network and envisioned network architecture described gives better solution to prolong the network life in IoT network.

References

1. Jean-Philippe Vasseur, Adam Dunkels, *Interconnecting Smart Objects with IP*. Elsevier, 2010.
2. *The Internet of Things*. ITU Internet reports, November 2005.
3. Gyu Myoung Lee, Jungsoo Park, Ning Kong, Noel Crespi and Ilyoung Chong, *The Internet of Things - Concept and Problem Statement*. Internet Research Task Force, July 2012.
4. Azzedine Boukerche, *Algorithms and Protocols for Wireless Sensor Networks*. Wiley-IEEE Press, October 2008.
5. Ian G Smith, Ovidiu Vermesan, Peter Friess, Anthony Furness and Martin Pitt, *Internet of Things European Research Cluster*. 3rd edition, 2012.
6. JeongGil Ko, Andreas Terzis, Stephen Dawson-Haggerty, David E. Culler, Jonathan W. Hui and Philip Levis, *Connecting Low-Power and Lossy Networks to the Internet*. IEEE Communications Magazine, 49(4), April 2011, pp. 96–101.
7. Francis daCosta, *Rethinking the Internet of things - A scalable Approach to Connecting Everything*. Apress, 2013.
8. Jim Chase, *The Evolution of Internet of things*. Texas Instruments, white paper September 2013.
9. Rolf H. Webera and Romana Weber, *Internet of things the legal perspectives*. Springer, 2010.
10. J. Mongay Batalla, G. Mastorakis, C. X. Mavromoustakis and J. Zurek, *On co-habiting networking technologies with common wireless access for Home Automation Systems purposes*. To appear in IEEE Wireless Communication magazine, 2016.
11. Y. Kryftis, C. X. Mavromoustakis, G. Mastorakis, E. Pallis, J. Mongay Batalla and G. Skourletopoulos, *Resource Usage Prediction for Optimal and Balanced Provision of Multimedia Services*. 19th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Net works (CAMAD), Athens, Greece, 2014.
12. H. Zainol Abidin, N. M. Din, I. M. Yassin, H. A. Omar, N. A. M. Radzi and S. K. Sadon, *Sensor Node Placement in Wireless Sensor Network Using Multi-objective Territorial Predator Scent Marking Algorithm*. Arabian Journal for Science and Engineering, Springer, 39(8), August 2014, pp. 6317–6325.
13. Mohamed Younis and Kemal Akkaya, *Strategies and techniques for node placement in wireless sensor networks: A survey*. Elsevier, Ad Hoc Networks, 2008, pp. 621–655.
14. J. L. Gao, *Energy efficient routing for wireless sensor networks*. Ph.D. thesis, Electrical and Computer Engineering Department, UCLA, June 2000.
15. Bahuguna Renu, Mandoria Hardwari lal and Tayal Pranavi, *Routing Protocols in Mobile Ad Hoc Network: A Review*. Quality, Reliability, Security and Robustness in Heterogeneous Networks, Springer, 2013, pp. 52–60.
16. K. Akkaya and M. Younis, *A survey on routing protocols for wireless sensor networks*. Ad Hoc Networks Journal, 2005, pp. 325–349.

17. M. A. Youssef, M. F. Younis, and K. Arisha, A constrained shortest-path energy aware routing algorithm for wireless sensor networks. In Proceedings of WCNC, 2002, pp. 794–799.
18. Fang Liu, Kai Xing, Xiuzhen Cheng and Shmuel Rotenstreich, Energy efficient MAC layer protocols in ad hoc networks. Resource Management in Wireless Networking, The George Washington University, 2004.
19. A. Willig, Wireless sensor networks: concept, challenges and approaches. Springer, e & i Elektrotechnik und Informationstechnik, 123(6), 2006, pp 224– 231.
20. Feng Xia and Azizur Rahim, MAC Protocols for Cyber-Physical Systems. Springer, June 2015.
21. Rajendran, V., Obraczka, K., Garcia-Luna-Aceves, J. J., Energy-efficient, collision-free medium access control for wireless sensor networks. In: Proc. ACM SenSys 03, Los Angeles, California, November 2003.
22. Christine E. Jones, Krishna M. Sivalingam, Prathima Agrwal and Jyh Cheng Chen, A Survey of Energy Efficient Network Protocols for Wireless Networks. Wireless Networks, 2001, pp. 343–358.
23. C. X. Mavromoustakis, A. Andreou, G. Mastorakis, A. Bourdena, J. Mongay Batalla and C. Dobre, On the Performance Evaluation of a novel Off loading-based Energy Conservation mechanism for Wireless Devices. 6th Springer International Conference on Mobile Networks and Management Springer-MONAMI 2014, Wuerzburg, Germany, 2014.
24. Peng Cheng, Chen-Nee Chuah and Xin Liu, Energy-aware Node Placement in Wireless Sensor Networks. IEEE Communications society, Globecom, 2004, pp. 3210–3214.
25. Koustuv Dasgupta, Meghna Kukreja and Konstantinos Kalpakis, Topology- Aware Placement and Role Assignment for Energy-Efficient Information Gathering in Sensor Networks. Proceedings of the Eighth IEEE International Symposium on Computers and Communication (ISCC'03), 2003, pp. 341–348.
26. Santpal Singh Dhillon and Krishnendu Chakrabarty, Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Networks. International Conference on Wireless Communications and Networking, IEEE, 2003, pp. 1609–1614.
27. Kirankumar, Y. Bendigeri and Jayashree D. Mallapur, Energy Aware Node Placement Algorithm for Wireless Sensor Network. Advance in Electronic and Electric Engineering, 2014, pp. 541–548.
28. Ataul Bari, Relay Nodes in Wireless Sensor Networks: A Survey. University of Windsor, November 2005.
29. Jian Tang, Bin Hao and Arunabha Sen, Relay node placement in large scale wireless sensor networks. Computer Communications, Elsevier, 29(4), February 2005, pp. 490–501.
30. Xiuzhen Cheng, Ding-zhu Du, Lusheng Wang and Baogang Xu, Relay Sensor Placement in Wireless Sensor Networks. IEEE Transactions on Computers, 56(1), 2001, pp. 134–138.
31. J. N. Al-Karaki and A. E. Kamal, Routing techniques in sensor networks: A survey. IEEE Communications, 11(6), 2004, pp. 6–28.
32. F. Ye, A. Chen, S. Liu, and L. Zhang, A scalable solution to minimum cost forwarding in large sensor networks. In Proceedings of the Tenth International Conference on Computer Communications and Networks (ICCCN), 2001, pp. 304–30.
33. K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, Protocols for self-organization of a wireless sensor networks. IEEE Personal Communications Magazine, 7(5), 2005, pp. 16–27.
34. Sarwesh P., N. Shekar V. Shet, K. Chandrasekaran, Energy Efficient Network Architecture for IoT Applications. IEEE, International Conference on Green computing and Internet of Things, 2015, PP. 784–789.
35. Lin-Huang chang, Tsung-Han Lee, Shu-Jan Chen and Cheng-Yen Liao, Energy Efficient Oriented Routing Algorithm in Wireless Sensor Networks. IEEE international conference on Systems, Man and Cybernetics, 2013, pp. 3813–3818.
36. Hasan Farooq and Low Tang Jung, Energy, Traffic Load, and Link Quality Aware Ad Hoc Routing Protocol for Wireless Sensor Network Based Smart Metering Infrastructure. International Journal of Distributed Sensor Networks, Hindawi Publishing Corporation, 2013.

ID-Based Communication for Access to Sensor Nodes

Mariusz Gajewski, Waldemar Latoszek, Jordi Mongay Batalla,
George Mastorakis, Constandinos X. Mavromoustakis
and Evangelos Pallis

Abstract Home automation and intelligent building are the areas where Internet of Things (IoT) has been applied relatively early. Everyday intuitive use of smart things can increase comfort at home and productivity at work. Introducing new solutions for smart devices controlling the natural human environment rises challenges— especially when fast and reliable communication within the hierarchized network is needed. Specifically, we consider scenario, where the network structure is aligned to a building structure. Moreover, we used the hierarchized addressing structure based on unique identifiers to benefit from Information Centric Network based architecture. This chapter describes the design, implementation and test results of the ID-based communication network with a particular emphasis on interworking with a sensor node.

M. Gajewski (✉) · W. Latoszek · J.M. Batalla
National Institute of Telecommunications, Warsaw, Poland
e-mail: M.Gajewski@itl.waw.pl

W. Latoszek
e-mail: W.Latoszek@itl.waw.pl

J.M. Batalla
e-mail: jordim@interfree.it

G. Mastorakis · E. Pallis
Department of Informatics Engineering, Technological Educational Institute of Crete,
Heraklion, Greece
e-mail: gmastorakis@staff.teicrete.gr

E. Pallis
e-mail: pallis@pasiphae.eu

C.X. Mavromoustakis
Department of Computer Science, University of Nicosia, Nicosia, Cyprus
e-mail: mavromoustakis.c@unic.ac.cy

1 Introduction

The concept of intelligent building was born in 80s [1] and since that time dedicated solutions are developed by many contributors. The high growth of popularity of this concept, observed in the past few years, has coincided with the growing popularity of cheap hardware platforms, well-tailored to customer needs. Following this trend, electronics manufacturers supplied to the market small computers (Single Board Computers, SBCs) and a number of cheap electronic components: sensors, actuators, communication modules [2]. Mainly, the last category of devices (known also as the System on Module, SoM) solutions are popular on the market, because they are equipped with broad range of I/O ports which satisfy the needs of professionals and amateurs. Moreover, many of them are able to run Linux (or Linux based) operating system what raises new opportunities for open source community. The SoM solutions became popular both for prototyping and target platform, however the target hardware is often optimized for properties such as. dimensions, energy usage, etc. Therefore, the target device requires redesigning the PCB (Printed Circuit Board) to match the specific project requirements. In turn, prototypes are often developed using popular SoM platforms. One of the most popular and available on the market since the 2012 is the Raspberry Pi platform [3]. Primarily, demand for that platform was driven by hobbyists and education sector. Currently, there is also interest in industrial sector, in which the Raspberry Pi has been used to support various factory machines and control systems.

This article presents results of the Raspberry Pi implementation of the sensor node which communicates using an ID-based Service-oriented COMMunications system (called IDSECOM system). As described in [4, 5], this system proposes a novel approach to data transmission for IoT—particularly in hierarchized networks. It is based on integration of the identifiers assigned to objects, services and network services. In IDSECOM, the functionalities of the ID layer and the network addressing are performed by the use of IDentifiers related to the physical location of the sensors/actuators creating, in this way, a location-oriented network topology.

This article extends previous work which basically aimed at implementation of the full assumed functionality of the IDSECOM network node. The results described in [6] encouraged us to implement also the sensor node which is able to support the ID communication while providing sensor readings (ID layer communication). Therefore, the main research effort were focused on implementation of the sensor node in a hardware platform. In that case, adaptation and launching the IDSECOM functionality in hardware running Linux OS have raised a lot of challenges. Section 2 presents the main limitations of hardware platforms and implications for its choice of the sensor node. In Sect. 3 we describe existing IDSECOM architecture and software modules including changes related to the implementation of the sensor node. Section 4 describes implementation issues including changes in the IDSECOM network node implementation that have been made on the basis of

the conclusion presented in [5]. Finally, in Sect. 5 we present test results of the completed IDSECOM system implementation with emphasis on interworking with the sensor node.

2 Hardware Requirements for the IDSECOM Sensor Node and Review of the Market Available Solutions

The intelligent building solutions (also known as home automation or smart home) comprises typically three categories of elements: (1) edge devices (nodes), which often perform one dedicated function (e.g. temperature measurement); (2) hubs or gateway nodes, which gather data from edge devices; and (3) larger processing elements which aggregate collected data and perform computation. The first category is based on constraint devices while the others are usually based on powerful equipment. This classification has been proposed in [6, 7]. More precisely, this distinction means that constrained devices offer limited power, memory and processing resources strictly tailored to its tasks. In turn, powerful devices are typically powered by the mains supply which may offer enough computational power, memory and communication interfaces to perform additional tasks.

We focused on the first category of devices while looking for the hardware platform dedicated for interworking within the IDSECOM system. Essential hardware requirements result from the concept of the architecture node and are the following:

- availability of the Linux-kernel-based operating systems for porting the implementation described in [5],
- sufficient number of I/O ports for connecting peripherals and sensors,
- support for multitasking,
- at least one physical network interface for connecting to the IDSECOM border node.

2.1 ARM Based Hardware Platforms

Above requirements are met by ARM processors and Single Board Computers (SBCs) built around it. Currently, many off-the-shelf SBCs are ARM-based including one of the most popular Raspberry Pi (RPi) System on Module (SoM). The Raspberry Pi 2 was chosen for the prototype implementation of the sensor node.

The Raspberry Pi 2 is the result of the first deep hardware revision since the Raspberry Pi was issued in 2012. The first generation of Raspberry Pi boards was based on single-core 700 MHz BCM2835 ARMv6 processor. During that time two RPi models were issued. The first one, Model A, was equipped with 256 MB of

RAM and one USB port. The second and most popular, Model B, was equipped with two USB ports, and additionally the LAN port.

The base (A and B) Raspberry Pi models had a 26-pin GPIO (General Purpose Input/Output) connector used for reading from various sensors and for writing output to external devices. According to several users the number of available pins was insufficient. Moreover, in original RPi linear 3,3 V regulators may cause overheating under a high load and in consequence excessive power consumption. This inconvenience was removed in the Raspberry Pi “plus” series designed separately for A and B models. Furthermore, the RPi + boards received the micro SD card slot instead of the full size SD slot.

The second generation of the RPi keeps modifications known from the RPi + series related to the board layout, GPIO, USB design and power distribution, preserving the same board size. The most substantial change relates to CPU for which hardware designers proposed more powerful processor—quad-core 900 MHz ARMv7 99BCM2836. It causes the increase of the power consumption over the Model B+, to a level similar to the original B. In return, the new SBC offers about six times better performance comparing to model B, (see [8, 9] for details).

Several Linux distributions have been ported to the Raspberry Pi’s BCM2835 chip and its successors since launching Raspberry Pi to the market. Together with distributions like Debian, Fedora Remix and ArchLinux, Raspberry Pi has become hardware platform for running many Linux based applications and tools. The use of ARM CPU architecture means that applications based on the source code written for Intel x86 or x64 processors will not run without code rewriting. However, porting applications from x86 architecture varies from a source code. As long as it does not depend on code supported only on x86 (e.g., closed source libraries dedicated to x86, x86 assembler routines, etc.) the application code is portable. Research efforts in this area was presented in [10, 11] and many practical hints are published on blogs, forums and company pages (e.g., see [12]).

The organization of the kernel source code depends on architecture. Moreover, porting methods of the new kernel to a new board varies on performance and physical board characteristics. The source code may be compiled on target hardware platform or cross-compiled on the developer machine. The first approach, however requires installing a set of compiling tools and libraries, ensures that the code will run on the target machine.

In turn, compiling a Linux kernel for another CPU architecture is usually much faster than compiling natively, particularly when the target system is slower than the developer Linux workstation. Moreover, the development tools are suited mostly for high performance workstations.

The Linux kernel supports a lot of different CPU architectures which make the Linux capable of working in hardware diversified environments. In most cases, each kernel version is maintained by a different group of contributors who are mainly related to SoC vendor or SoM manufacturer. But the great part of the kernel source code is derived from the Linux main line and supported by the contributors community. It ensures that the code is always up-to-date including security fixes,

bug fixes, drivers for popular devices as well. On the other hand, relying on Linux main line sources does not ensure support for specific devices like GPU, VPU etc. This may be particularly troublesome for different SoM solutions, which are manufactured on the basis of different components in short series. In this case, support for the drivers is usually ensured by chip manufacturers and the Linux enthusiasts community.

2.2 *Sensors*

Our further works were concentrated on implementation of IDSECOM sensor node. We decided to implement this functionality on Raspberry Pi 2 using Ubuntu 14.04 distribution. Moreover, we also connected DS18B20 [13] temperature sensor and HC-SR04 [14] ultrasonic sensor for tests. The temperature sensor was connected using 1-wire interface with the ARPI board for analog/digital (A/D) conversion. Software modules dedicated for temperature and distance readings made use of the library delivered by the ARPI 600 manufacturer [15].

The testing process of the IDSECOM sensor node implementation was based on measurement of data access time. For test purposes the sensor node was equipped with two sensor categories which differ in the data readings variability. The first category includes sensors which deliver slowly changing data (e.g. temperature). These sensor readings are valid for a longer period of time and may be used according to the user's preference. In consequence, "the validity time" value assigned to sensor readings which use time at a minute resolution. Thus, application which requests for a temperature readings much often receives data cached in IDSECOM nodes than received data from the sensor node.

The second sensor category includes sensors which expose fast changing data, e.g. proximity sensor which offers measurement of distance between the sensor and object. Due to the fact that the distance can vary rapidly, reading shall be performed at shorter intervals than in case of temperature sensors. We assumed that readings from the proximity sensor were performed maximum at second resolution. On the other hand, readings cannot be done too fast. It results from the measurement method, which is based on ultrasonic signals in our implementation (we used HC-SR04 ultrasonic sensor). The method used in our measurement is based on the fact that the sensor emits a pulse of sound and then receiver picks up the sound wave after it has bounced off any obstacle. The time between sending and receiving the echo signal is converted into the distance by the RPi. The sensor manufacturer suggests not using less than 60 ms interval between successive measurement to avoid overlapping signal pulse sent and rebound. In consequence, this value is the shortest period of time between readings of the distance from object.

Authors used C as the main programming language for sensor software development. This choice was dictated by the fact that the IDSECOM node source code is also written in C language. For a slowly changing sensing data, the probe sense rate does not exceed normally once every few seconds/minutes. In this approach,

temperature requests may be cached in network nodes. In consequence, direct sensor readings are performed rarely and their frequency depends of “validity time” of cached data. For fast changing data, this value is appropriately modified by the administrator. Basic sensor characteristics including i.e., IDSECOM Identifier and assigned validity time are stored in the sensor node in a configuration file. The contents of this file is used in the registration process. Moreover, the initial value (reading) is sent to the IDSECOM border node during the registration process.

3 IDSECOM System Architecture and Software Specification

The basic idea of the IDSECOM system assumes embedding the physical Objects/Service addressing in the network layer level. Additionally, it makes use of unique human-readable ASCII identifiers assigned to each object and to each offered service. It follows a hierarchical addressing scheme where each level of hierarchy is represented by one address label.

These identifiers are used as addresses for routing packets by IDSECOM nodes. The service/objects addresses are formed as the concatenation of all labels beginning from the root node; separated by a full stop character, e.g.: build001.floor001.room0001.fume_d01. Moreover, we assume mixed case characters and the special ASCII characters excluding: (1) the full stop character (“.”), which separates the successive labels, and (2) the asterisk character which is used as a wildcard in multicast/anycast addressing.

Besides ID based packet forwarding, the IDSECOM nodes perform functions which are specific to IoT applications, such as registration and resolution. Moreover, IDSECOM nodes offer additional features that may improve performance in IoT scenarios (i.e., packet caching). Apart from this functionality, the IDSECOM nodes are able to route standard IP traffic. Connected IDSECOM nodes constitute a network topology based on a hierarchical tree structure, which eliminates forwarding loops (connected nodes form the acyclic graph). Network nodes may be either physical or virtual L2 switches.

3.1 Functional Architecture of IDSECOM Nodes

The essential functionality of each IDSECOM network node encompasses two groups of functions: (1) functions responsible for packet forwarding, and (2) functions performing IoT specific tasks. The first group includes functions related to forwarding of requests to objects and data from objects in opposite direction. In turn, the IoT specific functions performed by each network node include:

- registration of new IoT objects and services in IDSECOM system;
- resolution of published IoT objects/services;
- publication of information about availability of registered IoT objects/services.

The above listed functionalities have been implemented as user and kernel space modules of a Linux operating system. Each IDSECOM node module includes Forwarding, Caching, Registration and Resolution modules. The first two modules are responsible for proper packet forwarding across the network. For that purpose the forwarding module analyses packet header and decides whether the received packet should be processed locally or send to proper output interface. If the packet has to be send to another node, the forwarding table is polled for outgoing interface. Copies of the forwarded packets are also stored locally in caching table, both: sent by applications requests and data sent by IoT objects in response to a request. This approach preserves sensor nodes from battery drain because forwarding network nodes offer also cached sensor data as long as the cached data are valid. For this purpose sensor node operator should be able to set this value depending on the sensor type. Thus, slow changing phenomena may be monitored with small temporal resolution and/or small spatial resolution.

3.2 Functional Architecture of IDSECOM Sensor Node

The Fig. 1 shows the architecture of the IDSECOM sensor node. The whole implementation of this architecture includes both kernel and user space. The blue color indicates the modules, that operates on the IDSECOM frames. The red color means the modules, which performs an operation on the data structures and the gray color shows modules responsible for configuration of connected devices (sensors).

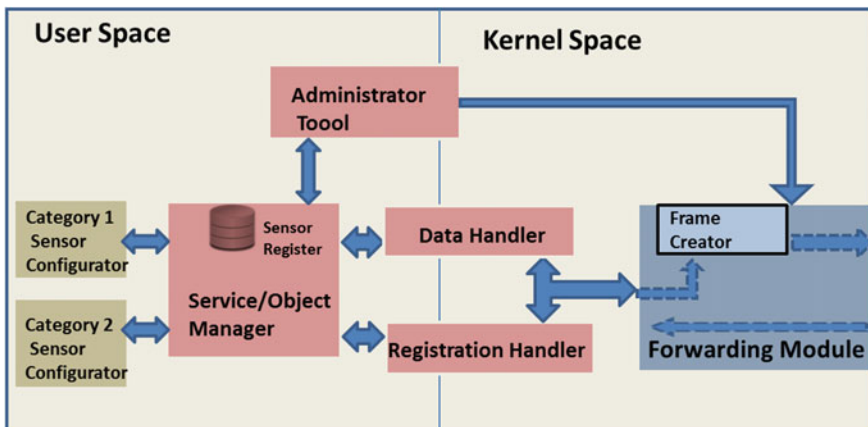


Fig. 1 Software modules within the IDSECOM sensor node

The connection of the module has to be preceded by an initial configuration on the node, which is performed by an **Administration Tool**. This module allows for configuration of both parts (user space and kernel space). In case of the kernel space it communicates with **Forwarding Module** in order to send parameters of default gateway (Ethernet interface to communication with the core network) to the edge IDSECOM node. The parameters are: source MAC, destination MAC, name of the interface) and full name of the node. As a part of the communication with Service/Object Manager (user space) it sends all necessary information to configure Sensor Register file.

The main module in the kernel space is the Forwarding Module. It has the following functionalities: (1) handling the incoming frame based on *the message info*, (2) communication with its sub-module—**Frame Creator** in order to fetch the frame to send it to the outgoing interface, (2) communication with **Data/Register Handlers** to send necessary data to the user space. The Frame Creator generates IDSECOM frame according to information set by the Administration Tool and other information from Handlers, that comes from the user space.

The communications between user space and kernel space is provided by the Data/Registration Handlers. Both Handlers are responsible for a correct transmission of the appropriate data structures, that are exchanged in the processes. The data Handler is responsible for the providing two-side communication kernel-user space in the process of data transmission. The Registration Handler does the same in the process of Registration the sensors.

The main module of the user space side is the **Object/Service Manager**. It has a full access to the **Sensor Register** file, which provides the information about the sensors. It can also update some data in the file. The information about the sensor can be taken direct from the Sensor Configurations. Furthermore, the module communicates with the Data/Register Handlers in order to exchange of information in appropriate processes.

The last **Sensor Configurators** have a direct access to the sensors. In our implementation there are two category of these communicators. The first category Sensor Configuration manages the sensors, that generate measurement data with the long validity time. In contrast, the second category deals with sensors, that are able to transmit the measurement data very fast.

3.3 *Communication Procedures in IDSECOM System*

Each IDSECOM node may play a dual role in the message forwarding:

- as a border network node for connected IoT objects, which performs registration processes and is responsible for accessing IoT services offered by them;
- as an IDSECOM network node which performs forwarding processes for L2 traffic including ID frames (data frames used in a ID layer are called ID frames, and, at a higher level, ID messages).

Moreover, there are also end nodes (sensor nodes) which play the role of IoT objects. They are attached to the IDSECOM system through the border network nodes and deliver sensor readings or perform actions desired by the user application. Although sensor nodes do not perform data forwarding (as IDSECOM network nodes), they have to be able to support communication at the ID layer.

Essentially, the IDSECOM network nodes communicate with each other at the ID layer using Register, Resolution, Request or Data messages. There are also messages used for communicating the status of the processed requests. The following table lists the functions performed by each IDSECOM nodes as well as messages exchanged between them. More detailed information on message format is set out in [5], (Table 1).

All the exchanged messages share a common generic format as defined in [5] and consist of 3 fixed fields and 2 or 3 additional fields depending on the message type. The fixed fields encompass:

- **Address length**, which indicates the number of concatenated labels which constitute the full domain address;
- **Address (coded using ASCII characters)**, which points either the source address in the case of Data messages or the destination address in other cases;
- **Message info**, which carries information about communication mode (multicast or unicast), message length and message type.

Table 1 Summary of messages exchanged between the IDSECOM nodes

Process	Communication parties	Exchanged messages
Registration	Sensor node (IoT object), IDSECOM node	The Register message is sent from the sensor node to the border node after attaching it to the IDSECOM system. In case of acceptance, the Register_accepted message is sent to the object. Otherwise, the Register_failed message is sent the cause referenced in the Failure info field
Resolution	Application server, IDSECOM node	The Resolution message is sent from the user application to the border node to obtain information about registered objects/services. The destination node responds by a series of Resolution_response messages, one for each object/service registered in the node. If the desired node does not respond, the user application can send a Resolution_alert message to the network node to inform the border node about the unavailability of the attached object
Data retrieving	Application server, Sensor node (IoT object)	The Request message is sent to the IoT object registered in a border node to force desired action. The IoT object either answers with Data message (with sensor readings) or executes the requested task without sending a response (e.g., sets a desired temperature)

The following paragraphs describe IDSECOM functionality implementation as Linux modules in sensor node (IoT object).

3.4 Registration Process

The Registration process (Fig. 2) is initiated by the Object/Service Manager immediately after the sensor node has been attached to the IDSECOM network. The Manager calls the function *register_init()* of the Registration Handler on the user-space side, taking as the parameter, the table with information of all sensors connected to the sensor node. Each element of the table contains a data structure consisting of three parameters: *ObjectID*, *sensor category* and *sensor type*. In the next step, the Registration Handler creates the connection with its kernel site and creates corresponding messages, that include above mentioned parameters. If the messages are complete, it starts sending them to the kernel space, where the second part of Handler should send back an acknowledgment of receipt. After successful transmission procedure, the data structure is stored in the kernel space. Moreover, the Registration Handler initiates the function *start_reg()* of Forwarding Module, that starts the registration process in the module by calling of its handler function. Next, the Forwarding Module redirects the process to the Frame Generator, which is responsible for preparing *Registration Request* message based on the information about: the default gateway parameters—source and destination MAC addresses ID layer addresses and the structure with the sensor parameters. According to the frame

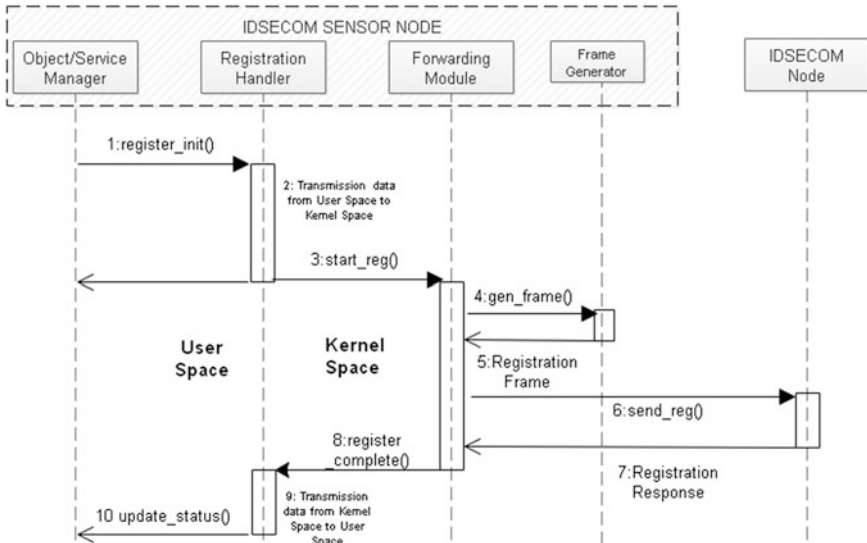


Fig. 2 Sensor node registration process

specification [5], it sets the *localhst* as an ID layer destination address. The parameters from the structure are copied to the *Information* field of the message in such way that each parameter precedes 1-byte value of its length, which can be required by an Application, that parse the *Information* string in resolution process [5]. After that, the Forwarding Module receives the message and sends it to the connected border node using its *send_reg()* function. The IDSECOM border node starts locally the registration process [5] and after successful procedure answers with the *Registration Response* message. The handler function of Forwarding Module completes its work by returning an information about successful registration process. Finally, this information is transmitted through the Registration Handler to the user space Object/Service Manager, that changes status of the sensor to the registered.

The above procedure is performed iteratively until each connected sensor will be registered.

3.5 Data Transmission Process

Figure 3 presents the sequence diagram of the data transmission process in the sensor node. In order to get information about sensor parameters, it is necessary to send corresponding *Request Message* [4] to the sensor. The demanded data can be fetched from: (1) Data Caching Table of any node in the path to the sensor,

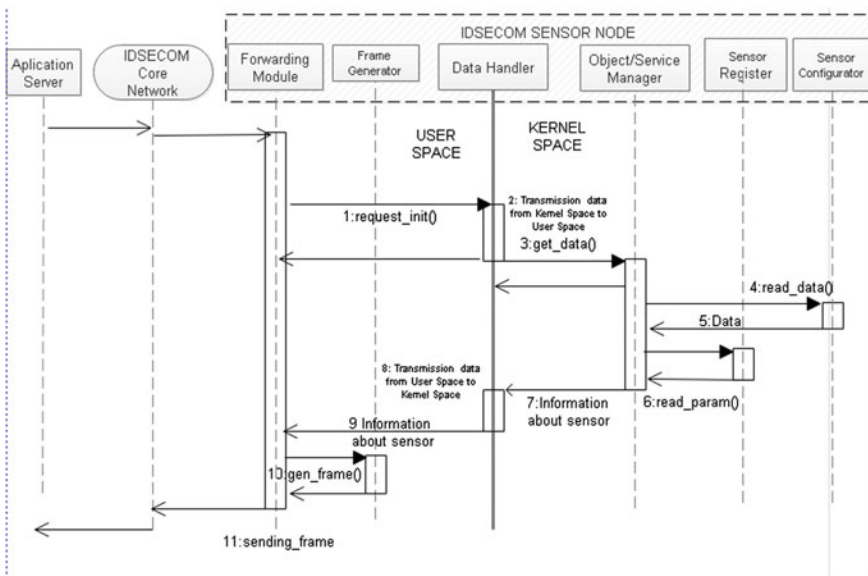


Fig. 3 Sensor node data transmission process

(2) sensor node, in case when any node on the path does not store the demanded data. The procedure of data transmission within the core network has already been described with details in the process of caching data in [5], therefore it will be omitted in the description. Hence, the presented case assumes, that the demanded data are not cached in the core network, which enforces the need to transmit the request to the sensor node.

The process is initiated by an application server connected to any network node. According to [4], the application sends the appropriate request (*Request Message*), which can contain e.g., the request to retrieve current measurements from the specific sensor. In the described case, the message was sent to the sensor node, which is indicated in the destination address of the *Request Message*. When the frame reaches the sensor node, it is handled by the Forwarding Module by the handler function. As the *Destination Address* of the message matches the receiving node name, the Forwarding Module handles the frame locally. For this purpose, it performs the following steps: (1) processes the frame in order to extract *ObjectID* field (sensor identifier) and *Information* field (information put by the application server), (2) stores the extracted data in an appropriate data structure, (3) calls the function *request_init()* of the Data Handler on the user space side with the created structure as a parameter. Then, the Handler calls the function *get_data()* of the Object/Service Manager with parameters of the structure. The Object/Manager performs the following operations: (1) calls the *read_data()* function of the Sensor Configurator identified by the *ObjectID* parameter in order to get the actual *measurement result* parameter. (2) creates the data by completing the rest of parameters from the Sensor Register. The completed structure that contains: *sensor category*, *sensor type*, *measurement result*, *validity time* is in the next step sent back to the Data Handler. The Handler transmits sequentially the following data of the structure to its user space side and sends the structure as a return of the *request_init()* function to the Forwarding Module.

The Forwarding Module combines the parameters of the structure into a single string and sends it to the Frame Generator as a parameter of the function *gen_frame()*. Note, that before each parameter of the string 1-byte value is put in order to specify its length. The Frame Generator generates a frame using the following data: (1) source and destination MAC addresses configured by the Administration Tool, (2) source and the destination ID layer addresses received by an interchanging the addresses of the of the *Request Message*, (3) the corresponding *Message Info* (3) the combined string, which is putted in the *Information field*. Finally, the created *Data Message* is returned back to the sender by the core network according to the rules specified in [4].

4 Implementation Issues

The implementation of the IDSECOM sensor node is performed in a Linux operating system (Ubuntu 14.4) and includes both user and kernel space modules. We used the standard Raspberry Pi Ethernet interface (100 Mbit/s) as a default gateway to the core network. The forwarding process operates in the kernel space and uses the same Linux kernel functions as the forwarding of the IDSECOM core node (see Fig. 4 for details). Thus, according to our assumptions [4] the sensor node should be able to generate an ID layer frame. For this reason, it was necessary to introduce changes in the structure of the Forwarding Module. The Forwarding process of the sensor node is very simplified. Unlike the core node, the sensor node does not need to keep the Forwarding Table. The general rule is that, if the frame destination address matches the receiving node name, the frame is handled in the user-space of that node. Otherwise, the frame is forwarded to the default gateway. The Forwarding Module uses own handler to process the ID layer frames.

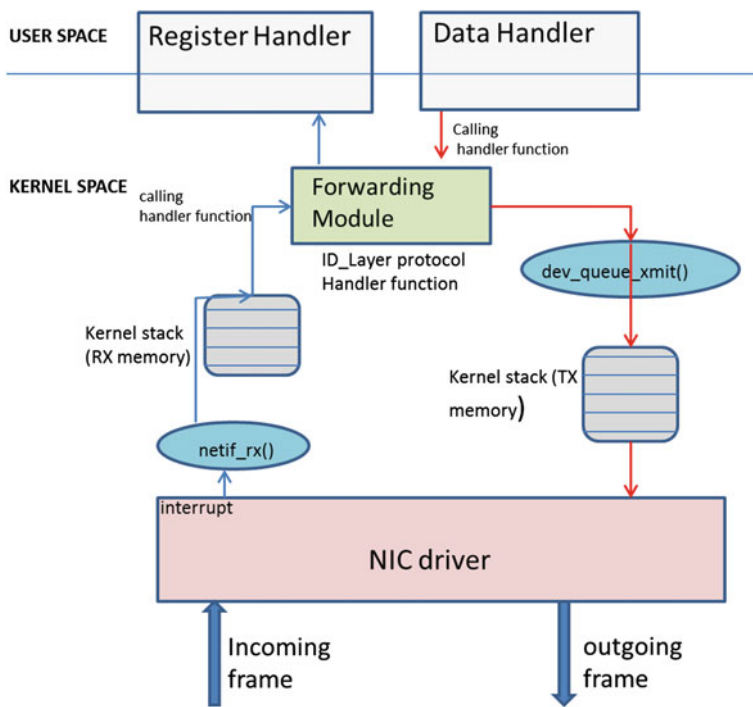


Fig. 4 The general architecture of frame transmission in Linux environment

4.1 *Implementation Issues in a Raspberry Pi Platform*

Each incoming frame causes a hardware interrupt, that finally after processing calls the *netif_rx()* (defined in *in/proc/net/dev.c*) function of the kernel space. This function receives a packet from a device driver and queues it for the upper (protocol) levels to process. When a frame is taken to handle, it generates software interruption, that causes calling the handler function in the Forwarding Module.

Finally, the frame is processed to extract necessary information, which depends on the process are further sent as a data structure to the user space through one of the Register or Data Handlers. In the case of the transmission in the second direction the handler function of the Forwarding Module is called by Registration or Data Handler in the result of corresponding request, that comes from the user space. Next, the Frame Creator (sub module of the Forwarding Module) creates corresponding for the process ID layer frame. Finally, the Forwarding Module sends the frame to the default gateway driver (using *dev_queue_xmit()* function defined in *in/proc/net/dev.c*).

4.2 *Software Changes in the IDSECOM Node Implementation*

Based on the test results presented in [5] we revised the source code of the IDSECOM node. On this basis, we introduced improvements, which aims to eliminate the events which generate errors. This improvement does not impact the basic functionality. The main change, that improves the overall system efficiency, was a modification of the caching algorithm.

The previous version assumed that Request/Data Caching Tables were searched sequentially from the beginning. If the searched entry is on the last position and the registry has many entries, this operation may take more time.

The main modification introduces sorting of the cached frames both in the Request and Data Tables. The sorting procedure was based on the ID layer destination address (for the Request Tables) or the source address (for the Data Tables), which identify uniquely all nodes within the network. The procedures of the caching process related to rules of searching in the Tables was presented in detail in [4].

For the above purpose we used an effective Binary Searching Algorithm [16], that computational complexity is $\log_2 N$ (where N is the number of entries in the Table). This algorithm was implemented in the following operations: (1) putting the data to Request/Data Tables, (2) fetching the demanded data from the Data Table. The first operation, that based on finding the proper position of the table for the new frame caused, that the Request/Data Tables was always sorted (from the start of the system to the finish). Therefore, the operation of getting the data is performed only on the sorted Data Table.

In the single iteration of the new algorithm the comparison of two strings (ID layer addresses) was realized using the function—*qsort()* [17] of the library *stdlib.h*.

5 Performance Tests of the IDSECOM Sensor Nodes

Essentially, the proposed test is based on the assumption of the caching performance test presented in [5]. The purposes of this test are: (1) to check access times (delay) in the network with the new introduced element—sensor node and modified algorithm of caching in the core nodes in the similar configuration and assumptions (2) to estimate the cost-effectiveness of caching the data for two categories of the sensors.

5.1 The Test Configuration and Assumptions

The test environment consists of three components:

- Packet generator/analyzer used for ID layer packet generation,
- Server platform which runs virtual machines (VMs) and contains connected IDSECOM nodes,
- IDSECOM sensor node which consists of Raspberry Pi with connected sensors and peripherals required for A/D conversion, logical level conversions, etc. For test purposes we used DS18B20 temperature sensor and the HC-SR04 ultrasonic distance sensor. Moreover, an additional component of the tested IDSECOM sensor node was the Arduino shields adapter For Raspberry Pi (ARPI600) which acts as a A/D converter for analog devices connected to Raspberry Pi.

Figure 5 shows the physical configuration of the test environment. The general assumptions of the test are similar to those presented in the test of caching performance [5]. The packet generator/analyzer (Spirent) simulates the application server, that generates the Request messages. The Request messages are configured with 8-segments destination addresses to allow for a transmission to the last node in the path (sensor node). If the data were in any intermediate node, then this node has to generate the Data message and sends it to the tester, because the validity time of the cached data was respectively long. In the following 8 different test scenarios the required data were cached in subsequent nodes on the path. The generator/analyzer measures the delay time between the sending of the Request message and the receiving of the Data message. In addition, each test scenario was performed for 3 different configurations of Request/Data Caching Tables, which differed in the number of entries in the tables, i.e., 2000, 10000, 1000000 entries. Each test was performed 10 times and, finally, the mean values of the delay time were calculated at 95 % confidence intervals.

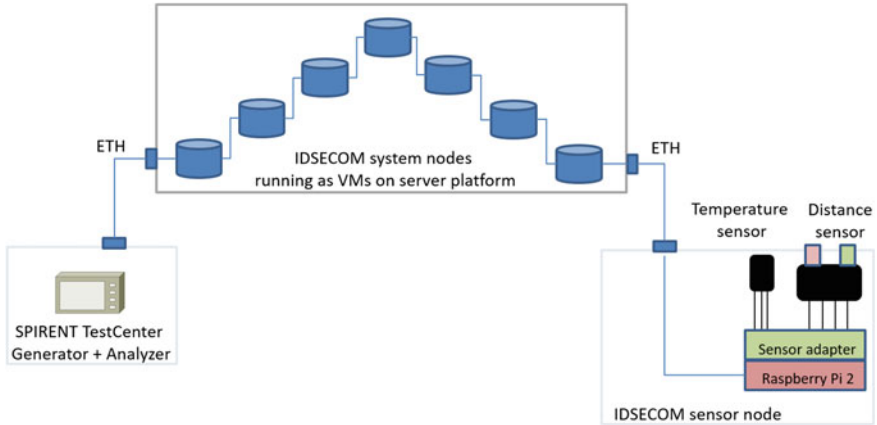


Fig. 5 A block diagram of the test environment

5.2 Test Results

The Fig. 6a presents the results of delays for each test scenarios (transmission to the following nodes on the path) and different configurations of the Request/Data Caching Tables. The 3 first functions (see the legend in the Fig. 6a) shows the impact of the sizes of the Request/Data Caching Tables on the delay. The red dashed line shows the result in the case of the transmission to the last node with caching disabled in all nodes on the path. This case indicates only the delay caused by the transmission and operations related to the forwarding process in the subsequent nodes. This test was performed for a specific configuration of the Forwarding Module of the sensor node, that was introduces only for the need of the scenario. In the test scenarios with caching, the demanded data was put in the Request/Caching Table in the position to simulate the worst case of the algorithm (this means the execution of such number of iterations, after which the temporary Request/Data Table will be limited to one element containing the position of the demanded data).

We can observed an increase of the delay time with increasing table size. Thus, in the difference of the test of previous version, where the increase was linear, the variability of the delay significantly decreases with increasing size of the table (for the test with the transmission to the 7th node the difference on the delay between scenarios with 2000 and 10000 Table sizes is about 0,2 ms. This difference growth only a little more than twice, when we consider scenarios with 10000 and 1000000). According to the theory of the used caching algorithm ($\log_2 N$) the increase of the delay is approximately logarithmic.

The next conclusion relates to the scenario, with the transmission to the sensor node, where we can observe an additional increase of about 3 ms. This increase is caused by the specific architecture of the sensor node, where data processing is

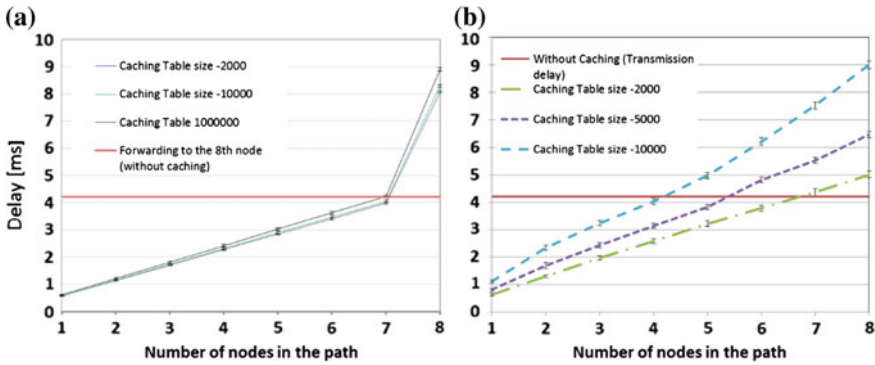


Fig. 6 The results of access time to the data stored in the following nodes of the path **a** actual version of caching algorithm **b** previous version of caching algorithm

performed partially in the user space. The need of the two-way transmission between kernel and user spaces and the processing in the user space influences significantly on the value of delay.

As stated above, the condition of the performed test is similar to the test of the delay of caching process for the previous version of the IDSECOM system presented in [5] (presented on the Fig. 6b). However, this test scenario presents the results for the improved caching process and under conditions of changed architecture of the last node (IDSECOM sensor node).

Taken into account only 2 test scenarios with (2000 and 10000 entries of tables) related to transmission to 1–7 nodes we can compare the differences caused only by the caching algorithm. For each of these scenarios we can observe the decrease of the delays. The achieved delays decreased respectively approximately 0,5 and 3,3 ms (case with the transmission to 7th node). The general conclusion is, that the increase of the Request/Data Tables sizes causes significantly advantage of current version of the caching algorithm in comparing to the previous version. That is obviously caused by the difference of linear (previous version) and logarithmic (current version) character of variability the delay relative to Table sizes.

The last conclusion refers to the main goal of our tests, that is evaluation of usefulness of the caching process for both type of sensor. The basic condition in the process of caching is that the time of storing the data in the table (validity time) should be longer than the time of access to the data. Our test results show, that the maximum value of the access time in the presented test configuration is about 9 ms independently of the category of the sensor, that forces to define validity times of the data longer than the 9 ms.

For the category 1 sensors (slowly changing data), where the typical validity time is determined with a minute resolution it is several orders of magnitude longer. Therefore, it can conclude, that there is no limitation of the caching use. In this case the caching process can significantly improve the access time to the data and reduce network traffic.

In case of the category2 (fast changing data) sensor, the typical validity times are determined with a millisecond resolution. Generally, it is possible to have similarly values of the access to the data and validity time. Therefore, it should be considered in the case using caching of the data. Although the access to the data can be shorter, there are also disadvantages of this solution. First, the caching will cause an additional traffic in the core network, Also there should be taken into account, that the data are delivered very frequently what can cause significant growth of the traffic in the core network. This can cause also a large allocation of the memory in the kernel space of the nodes, because of the growth of entries in the Data/Request Caching Tables. It is also necessary to speed up the procedure of free the memory allocated in the Request Caching Tables.

6 Conclusion

In IoT scenarios such as intelligent buildings or intelligent enterprises coexistence of various devices within one ecosystem is an important factor. In our research we focused on implementation the sensor node on an exemplary development platform. Our tests pointed out that we can observe an additional increase of about 3 ms in the scenario with the real sensor node. This increase is caused by the specific architecture of the sensor node, where part of the processing is performed in the user space. It requires the two-way communication between kernel and user space. Moreover, processing at the user space level increases significantly the observed delay.

The performance tests of the prototype conducted by us were aimed at checking the effectiveness of the implemented solution for delay in response to data requests from different sensors. As a result, we calculated the delays of the network and sensor nodes. The presented test results confirm the usefulness of kernel implementation which is suitable for transferring data from different sensors. However, our research has also shown that caching process should be applied only to specific data. Caching is beneficial if requests are being served from cache. For that purpose, the cached data must be valid longer than the time of access to the data from a sensor. On the other hand, fast changing data (e.g., object position) forces to set the validity time to adequate value. Moreover, short validity time value forces all network and sensor nodes to be synchronized. In consequence, setting the value of validity time should be geared to the data type.

Acknowledgments This work was undertaken under the Pollux IDSECOM project supported by the National Research Fund Luxembourg (FNR) and the National Centre for Research and Development (NCBiR) in Poland.

References

1. G. I. Fântână, Ș. A. Oae, “Evolution of Smart Buildings”, Proceedings of the 2013 International Conference on Environment, Energy, Ecosystems and Development (EEEAD 2013)
2. J. Mongay Batalla, G. Mastorakis, C. Mavromoustakis and J. Žurek, “On cohabitating networking technologies with common wireless access for Home Automation Systems purposes”. IEEE Wireless Communications (October 2016)
3. <https://www.theguardian.com/technology/2015/feb/18/raspberry-pi-becomes-best-selling-british-computer>
4. J. Mongay Batalla and P. Krawiec, “Conception of ID layer performance at the network level for Internet of Things”. Springer Journal Personal and Ubiquitous Computing, Vol.18, Issue 2, pp. 465–480 (2014)
5. J. Mongay Batalla, M. Gajewski, W. Latoszek, P. Krawiec, C X. Mavromoustakis, G. Mastorakis, “ID-based service-oriented communications for unified access to IoT”, (Computers & Electrical Engineering), pp. 98–113, vol. 52 (2016)
6. C. Lévy-Bencheton, E. Darra, G. Tétu, G. Dufay, M. Alattar, 2015, “Security and Resilience of Smart Home Environments Good practices and recommendations”, December 2015, European Union Agency For Network And Information Security (ENISA)
7. C. Bormann, M. Ersue, A. Keranen, 2014, RFC 7228: Terminology for Constrained-Node Networks
8. Raspberry pi 2 model b. RASPBERRY PI FOUNDATION. [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>
9. Raspberry pi 1 model b+. RASPBERRY PI FOUNDATION. [Online]. Available: <https://www.raspberrypi.org/products/model-b-plus/>
10. B. Chun-yue, Y. Liu, and R. Wang. “Research of key technologies for embedded Linux based on ARM.” 2010 International Conference on Computer Application and System Modeling (ICCASM 2010). Vol. 8. IEEE, 2010
11. S. Divya, and K. Kant. “Porting the Linux kernel to ARM development board.” International Journal of VLSI and Embedded Systems-IJVES, ISSN (2013): 2249–6556
12. “Free electrons” online training materials, [Online], Available: <http://free-electrons.com/docs/>
13. Resolution, DS18B20 Programmable. “1-Wire Digital Thermometer.” Data Sheets. DALLAS-MAXIM.[dostęp 11-07-2006]. [Online]. Available: http://www.maximic.com/quick_view2.cfm/qv_pk/2813 (2008)
14. Freaks, Elec. “Ultrasonic Ranging Module HC-SR04.”. [Online]. Available: <http://www.micropik.com/PDF/HCSR04.pdf> (2011)
15. ARPI600, Arduino Adapter For Raspberry Pi. [Online]. Available: <http://www.waveshare.com/wiki/ARPI600>
16. D. Knuth, “The Art of Computer Programming” Sorting and Searching (2nd ed.), Addison-Wesley Professional, ISBN 0-201-89685-0
17. Qsort documentation [Online]. Available: https://www.tutorialspoint.com/c_standard_library/c_function_qsort.htm

QoS/QoE in the Heterogeneous Internet of Things (IoT)

Krzysztof Nowicki and Tadeus Uhl

Abstract Applications provided in the Internet of Things can generally be divided into three categories: audio, video and data. This has given rise to the popular term Triple Play Services. The most important audio applications are VoIP and audio streaming. The most notable video applications are VToIP, IPTV, and video streaming, and the service WWW is the most prominent example of data-type services. This chapter elaborates on the most important techniques for measuring QoS/QoE in VoIP, VToIP/IPTV and WWW applications.

1 Introduction

Quality of Service (QoS) is of crucial importance to modern digital networks, not least to the Internet, which is increasingly going under the name Internet of Things (IoT). The term QoS is becoming a household phrase, and has long been anchored in the definition of Next-Generation Networks in ITU-T Standard Y.2001 [1]. Defined in Report TD 109rev2 [2], Quality of Experience (QoE) has been the subject of the most recent activities of the ITU-T Study Group 12. In November 2009 the European Parliament and the European Commission adopted the so-called Communications Packet combining their directives 2009/140/EC [3] and 2009/136/EC [4], that underline the importance of QoS. Both QoS and QoE are to be monitored continually, and preferably automatically, in modern networks, and that obviously means in the IoT too. EU research projects such as Leone [5] and mPlane [6] and standardisation organisations such as the Internet Engineering Task

K. Nowicki (✉)
Gdansk University of Technology, Gdańsk, Poland
e-mail: know@eti.pg.gda.pl

T. Uhl
Flensburg University of Applied Sciences, Flensburg, Germany
e-mail: tadeus.uhl@fh-flensburg.de; t.uhl@am.szczecin.pl

T. Uhl
Maritime University of Szczecin, Szczecin, Poland

Force (IETF) [7] have been hard at work to achieve this. However, monitoring the quality of service throughout the EU is proving to be an especial challenge, given the dynamic structure of the Internet. New services are forever being introduced and older ones restructured. The IETF aims to meet these challenges with the all-embracing Framework for Large-Scale Measurement of Broadband Performance (LMBP [7]) and, in keeping with the EU research project mPlane [6], promotes a flexible all-purpose solution.

Whilst these frameworks do contain fundamental concepts and specifications for the diverse areas of a distributed measuring system, such as the exchange and storage of acquired measurement data, they are by no means intended as complete systems. First of all, the suitable measuring equipment, storage units and results analysis software of various providers must be pieced together. In keeping with the trend of shifting more and more services into the cloud [8], the individual components of big-data systems are being designed to work flexibly in the cloud. Alongside large-scale, complex yet adaptable systems there is also a demand for specific, complete quality-assessment solutions that are easy to implement and use. Existing systems for monitoring Triple Play Services might benefit from the new possibilities of cloud computing. Since cloud-based solutions can usually be implemented without complicated precommissioning at the user's end, and the costs incurred through on-demand solutions are lower than long-term investments, the entry threshold will fall, and the use of indispensable measuring technology will become more appealing. For despite the unequivocal recommendations, continuous measurement of quality of service is still not a universal practice owing to the regrettable lack of suitable QoS/QoE measurement techniques for measuring environments.

2 Impairment Parameters in the IP Environment

Network theory comprises a considerable range of parameters which can and do impair the performance of a network and the quality of any service it might convey. These parameters can be assigned to various logical protocol layers. The more serious impairment parameters in IP networks (and that goes for the Internet of Things as well) include: upload and download speeds, delay, delay variation (jitter), packet loss ratio, and packet error ratio (see CEPT's ECC report 195 [9]). The criteria which CEPT used to choose the relevant standard were primarily outlined in ETSI Guide EG 202,057 [10] and ITU-T Recommendations Y.1541 [11] and G.1010 [12]. Table 1, based on ECC Report 195 [9], illustrates popular services, and the relative importance of the network parameters to the performance or quality of those services, or both. In the following table, their relative importance ranges from '-' (irrelevant) to '+++' (very relevant).

Table 1 shows that the type of service must be considered at all costs in any appraisal of QoS/QoE. Consequently, QoS/QoE measurement techniques must be

Table 1 Relevance of network impairment parameters to various applications

Service	Data transmission speed		Delay	Delay variation (jitter)	Packet loss	Packet error
	Down-stream	Up-stream				
Browse (text)	++	-	++	-	+++	+++
Browse (media)	+++	-	++	+	+++	+++
Download file	+++	-	+	-	+++	+++
Transactions	-	-	++	-	+++	+++
Streaming media	+++	-	+	-	+	+
VoIP	+	+	+++	+++	+	+
Gaming	+	+	+++	++	+++	+++

adjusted to take account of the nature of each service in the light of information contained in Table 1.

3 QoS/QoE Models and Techniques

In order to determine the QoS/QoE in a network, two models are generally used: (a) dual-ended model and (b) single-ended model; cf. Fig. 1 [13]. In the case of the dual-ended model, two signals are used: (a) original signal and (b) degraded signal. These two signals are available uncompressed. For this reason, measurements can be carried out for both Quality of Experience (subjective evaluation) and Quality of Service (objective evaluation). In the case of the single-ended model, only the impaired signal (compressed) is available. This allows only an objective evaluation (QoS) to be made. QoS measurement is referred to as “intrusive measurement” (online) in the case of the dual-ended model, and as “non-intrusive measurement” (offline) in the case of the single-ended model.

Two measurement techniques can be used in the two models cited: (a) signal-based and (b) parameter-based measurement. The dual-ended model uses tailor-made algorithms to compare the input and output signals of signal-based measurements. In the case of the single-ended model, this comparison is made by using a reference signal. In both cases the system to be assessed is treated as a “black box”. When carrying out parameter-based measurements though, a distinction is made between: (a) “glass box” and (b) “black box”. In the first case, both the structure of the system to be assessed and the reaction of the individual system components to the reference signal are known. This knowledge is then taken into consideration in a suitable model. Additionally, the network parameters can be measured, and included in the calculation of the QoS. In the second case, details of the system to be assessed are limited. For this reason, only the measured network

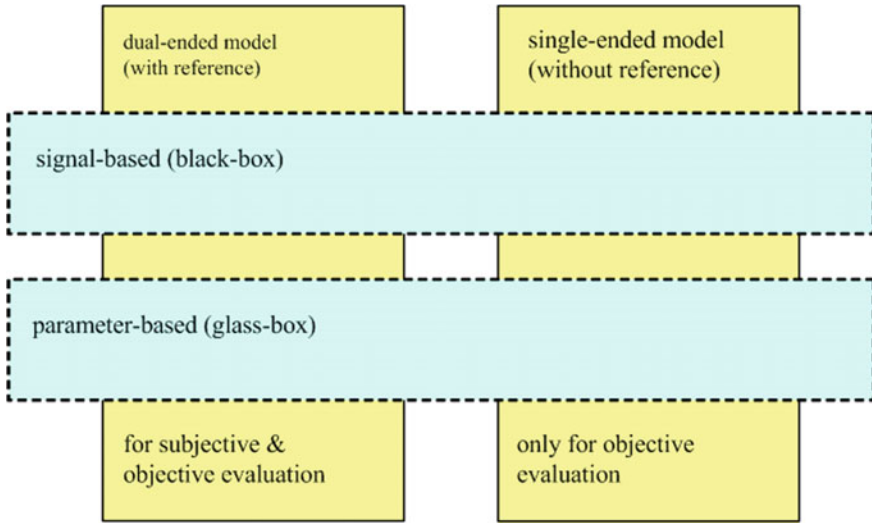


Fig. 1 Overview of QoS and QoE measurement techniques

parameters and the characteristic parameters of a given service are taken into account.

Applications provided in the Internet of Things can generally be divided into three categories: audio, video and data. This has given rise to the popular term Triple Play Services. The most important audio applications are VoIP and audio streaming. The most notable video applications are VToIP, IPTV, and video streaming, and the service WWW is the most prominent example of data-type services. The following section elaborates on the most important techniques for measuring QoS/QoE in VoIP, VToIP/IPTV and WWW applications.

4 QoS/QoE in the VoIP Service

4.1 Introduction

Figure 2 represents current QoS/QoE measurement techniques for the VoIP service. It is noticeable that several international standards touch on this area. The signal-based QoE measurement techniques PESQ and POLQA are very accurate; they are, however, time-consuming and can often only be implemented with a licence. That is why parameter-based QoS measuring methods are usually preferred in practice. The E Model, which was originally developed for circuit-switched telephone networks, has recently been adapted for use in IP networks to produce the modified E(IP) Model, released in 2014 as a patent [14], purchased by the company Nextragen GmbH [15] and implemented in their product Trace_View [16].

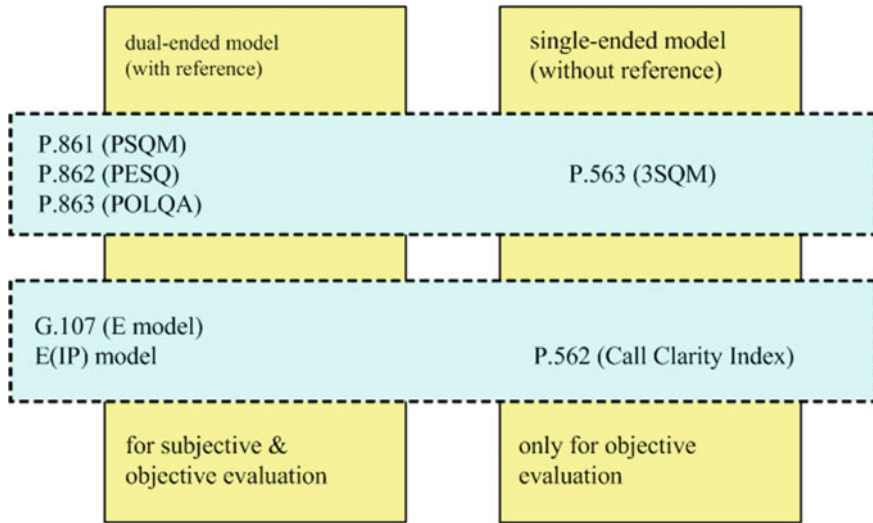


Fig. 2 Overview of QoS and QoE measurement techniques for service VoIP

The following section describes in detail the most important methods of measuring QoS/QoE in the VoIP service. The chapter then goes on to describe a series of analyses aimed to show up their strong points and weaknesses, similarities and differences.

4.2 QoS/QoE Measurement Techniques

4.2.1 PESQ and POLQA

Figure 3 shows a schematic diagram of the PESQ algorithm [17]. PESQ compares an original signal $x(t)$ with a degraded signal $y(t)$ that results when $x(t)$ has passed through a communications system. In the first step of PESQ a series of delays between original input and degraded output is computed, one for each time interval for which the delay is significantly different from the previous time interval. For each of these intervals a corresponding start point and stop point are calculated. Based on the set of delays that are found, PESQ compares the original signal with the aligned degraded output of the device under test using a perceptual model. The key to this process is the transformation of both the original and the degraded signal to an internal representation that is analogous to the psychophysical representation of audio signals in the human auditory system. This is achieved in several stages: time alignment, alignment with a calibrated listening level, time–frequency mapping, frequency warping, and compressive loudness scaling.

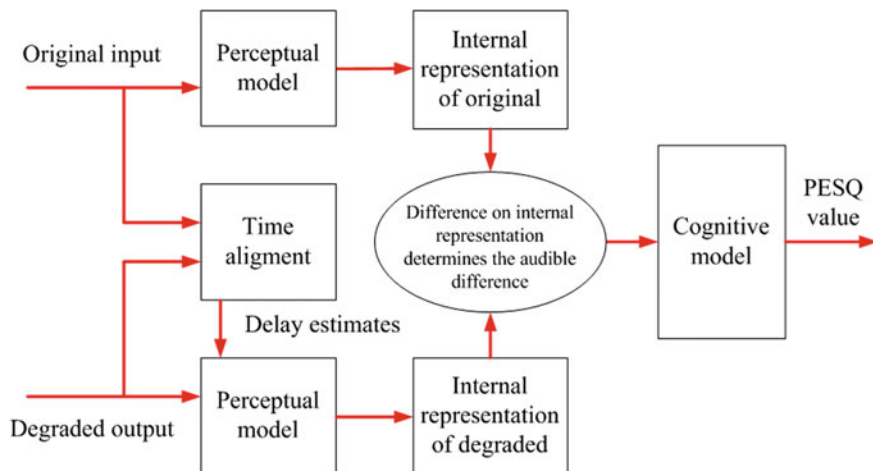


Fig. 3 Schematic diagram of the PESQ algorithm

The internal representation is processed to take account of effects such as local gain variations and linear filtering that may, however, have little perceptual significance. This is achieved by limiting the amount of compensation and making the compensation lag behind the effect. Thus minor, steady-state differences between original and degraded are compensated. More severe effects or rapid variations are only partially offset so that a residual effect will remain and increase the overall perceptual disturbance. At any rate, a relatively small number of quality indicators in the model is sufficient to cover all subjective effects. Going off the result of this comparison, it is possible to make predictions about the audibility of the interferences that have been added by the line under test, and a measure of the speech quality can be calculated. As a result, the PESQ value is determined. PESQ offers an output value for speech quality on a scale from “−0.5” to “4.5”, with values in the region of “−0.5” indicating a very poor speech quality whilst values in the region of “4.5” represent an excellent speech quality. It is possible to map the PESQ scale to the MOS-LQO scale (Mean Opinion Score—Listening Quality Objective; cf. ITU-T Rec. P.862.1 [18] and P.862.2 [19]) from “1” to “5”. The mapping function in the narrowband case is given by Eq. (1).

$$LQO = 0.999 + \frac{4.999 - 0.999}{1 + e^{-1.4945PESQ + 4.6606}} \quad (1)$$

Table 2 shows the evaluation of speech quality according to PESQ and MOS-LQO values for the narrowband case (300–3,400 Hz).

The mapping function for the wideband case is given by Eq. (2).

Table 2 QoS values on the PESQ and MOS-LQO scales (narrowband)

PESQ value	MOS-LQO value	Speech quality
4.5	4.55	Excellent
4	4.15	Good
3	2.82	Fair
2	1.63	Poor
-0.5	1.02	Bad

Table 3 QoS values on the PESQ and MOS-LQO scales (wideband)

PESQ value	MOS-LQO value	Speech quality
4.5	4.64	Excellent
4	4.35	Good
3	3.28	Fair
2	2.01	Poor
-0.5	1.04	Bad

Table 4 Limits of the MOS-LQO values obtained using the POLQA Algorithm

	NB mode		SWB mode	
	min. MOS value	max. MOS value	min. MOS value	max. MOS value
NB signal	1	4.5	1	4
WB signal	-	-	1	4.5
SWB signal	-	-	1	4.75

$$LQO = 0.999 + \frac{4.999 - 0.999}{1 + e^{-1.3669PESQ - 3.8224}} \tag{2}$$

Table 3 shows the evaluation of speech quality according to PESQ and MOS-LQO values for the wideband case (50–7,000 Hz).

Perceptual Objective Listening Quality Analysis (POLQA) is the successor to PESQ and functions in accordance with the specifications laid down in ITU-T Standard P.863 [20]. The algorithm has two modes: NB (NB reference signal) and SWB (SWB reference signal). Using the SWB mode of the POLQA method, an analyst can evaluate both sets of signals on a single scale. The degraded signals associated with a codec that normally has a sampling frequency below 48 kHz are recalculated internally to make a comparison of both input signals possible. In order to conduct a practical measurement using the SWB-Modus of POLQA, it might first be necessary to obtain a suitable sample specimen from the SWB reference signal by reducing the sampling rate (downsampling). The POLQA results obtained will all be MOS-LQO values. Table 4 shows the minimum and maximum values that can be achieved by using a given modus with a given signal.

The broad bandwidth of supported signals (NB, WB, SWB) and the limitation to values on the MOS scale (1.0–4.5 for the NB mode or 1.0–4.75 for the SWB mode)

mean that any MOS-LQO values that are obtained in accordance with Standard P.863 are offset by approx. 0.5 MOS when compared with corresponding MOS-LQO values obtained using the P.862 Standard. This certainly hampers any comparison of the results gained from the two algorithms. That is why a conversion takes place within the POLQA algorithm before the result is displayed. The recalculation is done using Eq. (3) for narrowband and Eq. (4) for broadband signals [21].

$$MOS_{LQO} = 0.79 + 0.0036 \cdot MOS_{P.863} + 0.2117 \cdot MOS_{P.863}^2 - 0.0065 \cdot MOS_{P.863}^3 \quad (3)$$

$$MOS_{LQO} = 0.276 + 0.7203 \cdot MOS_{P.863} + 0.00756 \cdot MOS_{P.863}^2 - 0.01141 \cdot MOS_{P.863}^3 \quad (4)$$

Both mapping functions were developed by the POLQA coalition consisting of the companies Opticom [22], SwissQual [23] and TNO [24] to enable the relative QoS values for signals with different sampling rates to be transferred to a single scale. Only versions of those algorithms are supported whose resulting values can be placed on this scale.

According to ITU-T Recommendation P. 830 [25] all recorded speech material (speech samples) should consist of simple, meaningful utterances. Sentences should be chosen that are readily understandable. In addition, the sentences should be separated by silent pauses into two or three sections and compiled in such a way that there are no obvious trains of thought between the individual sentences. The use of very short and very long utterances should be avoided; ideally, each utterance should be in the range of 2–3 s long. The speech samples themselves should be 8–12 s long, and 40–80 % of this time should be spoken. If longer pauses are desired, it is advisable to make several, separate recordings, each 8–20 s long.

The Web presence of ITU-T offers several reference files for the PESQ algorithm [25]. The Web pages of the company Opticom [22], the licence holder of the PESQ algorithm in Germany, also contain several reference files. Paper [26] contains a detailed study of reference signals suitable for PESQ.

4.2.2 E Model

The E Model is an implementation of ITU-T Recommendation G.107 [27]. It is used widely, both at the planning stage of networks and for monitoring the performance of active connections. The model is based on a parametrised description of the telephone network and also uses the psychological additivity parameter. It was developed using a large corpus of auditory tests. Since it also takes into account such things as quality loss due to echo, the E Model can also be used to evaluate the quality in conversations and similar scenarios. The result of the model's first step in the calculation is an evaluation factor R , which incorporates all transmission

Table 5 Factor R and MOS values

Factor R	MOS value	Speech quality
100	4.5	Excellent
80	4.0	Good
60	3.1	Fair
40	2.0	Poor
20	1.2	Bad

parameters that are relevant for any given connection. The factor R can then be converted into a MOS value using Eq. (5) [27]. The relationship was established through auditory tests conducted on a considerable number of test persons.

$$MOS = \begin{cases} 1 & \text{for } R < 0 \\ 1 + 0.035R + R(R - 60)(100 - R) \times 7 \times 10^{-6} & \text{for } 0 \leq R \leq 100 \\ 4.5 & \text{for } R > 100 \end{cases} \quad (5)$$

Table 5 shows the equivalence of the factor R to corresponding MOS values.

The central parameters of the E Model are Ie (Equipment Impairment Factor) and Bpl (Packet-Loss Robustness Factor). These two parameters can be adapted for any measurement of quality and can therefore have a decisive influence on any QoS values obtained by using the E Model.

4.2.3 E(IP) Model

As was mentioned above, the central parameters of the E Model are: Ie (Equipment Impairment Factor) and Bpl (Packet-Loss Robustness Factor). These parameters in the E Model have been calculated for various speech codecs for line-switched voice networks in exhaustive auditory tests, collated in tables as default values, and then published [27]. If these values are used in an evaluation of QoS in a VoIP environment, there will be large discrepancies between the PESQ values and the values calculated according to the E Model (cf. Point 4.4.), so some form of modification of these parameters is necessary. This is the point of the new E(IP) Model.

Numerous studies of the VoIP environment (see e.g. [28]) have shown that the BSLP (burst sample length product) has a significant influence on QoS values. That is sufficient reason to represent the parameters Bpl and Ie as functions of the BSLP. The nature of the BSLP makes it very useful in practice; it comprises both the character of packet losses in the real environment (block building) and the sample length that is actually used in practice for the speech codec used. Linear dependencies have been deduced between the Bpl , the Ie and the BSLP in several experiments using current speech codecs. These equations form the main modification to the overhauled E Model. Their patent was applied for, and issued in 2014 [14]. Equations (6–7) show as an example the dependencies mentioned above for the very well-known speech codec G.711 a-Law.

$$Bpl = 0.04 \cdot BSLP + 17.83 \quad (6)$$

$$Ie = -0.0155 \cdot BSLP + 10.153 \quad (7)$$

The evaluation of QoS can be conducted using either of two methods: (a) by making measurements in the real environment and (b) by emulation in a suitable numerical tool. The next point will present such a tool in some detail.

4.3 The Tool QoSCalc(VoIP)

Figure 4 shows the block diagram of the numerical tool for analysing QoS in the VoIP service. The tool has been called QoSCalc(VoIP) [29].

The following describes the operation of the tool, starting with a breakdown into its single steps:

- First, the reference file is loaded.
- The reference signal is coded in accordance with the speech codec selected.
- The coded speech samples are segmented according to the selected speech sample lengths and encapsulated in RTP packets.
- Network impairments (i.e. jitter, packet loss) are emulated in the block “Error”.
- The received RTP packets containing speech samples are latched in the jitter buffer where they are processed. Should packet losses occur, “silence insertion” according to Recommendation ITU-T G.113 is used [30].
- After encapsulation, the speech samples are decoded in the next block according to the codec selected.
- Finally, the received and reference signal are passed on to the PESQ or POLQA algorithm for analysis. Either algorithm will then calculate and output the QoE values on the MOS scale. It is also possible to pass the calculated parameters to the E Model or the E(IP) Model as soon as the RTP packets are received because either of these models is capable of determining and outputting QoS values on the MOS scale.

The QoSCalc(VoIP) tool was developed using the programming language C ++. All the steps described above are executed on the computer, so the transmission of RTP packets actually takes place within the tool itself.

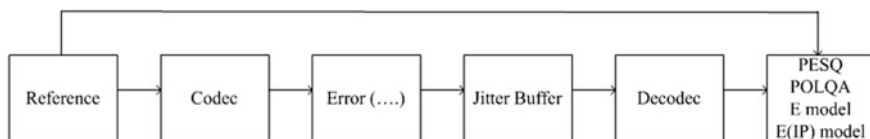


Fig. 4 Block diagram of the QoSCalc(VoIP) tool

The following codecs are currently supported: G.711 a-Law, G.711 μ -Law, G.711.1 (64 kbit/s, 80 kbit/s, 96 kbit/s), G.721, G.723.1, G.726 (16 kbit/s, 24 kbit/s, 32 kbit/s, 40 kbit/s), G.729a, GSM, iLBC, G.722 (64 kbit/s), G.722.2 (6.6 kbit/s, 8.85 kbit/s, 12.65 kbit/s, 14.25 kbit/s, 15.85 kbit/s, 18.25 kbit/s, 19.85 kbit/s, 23.05 kbit/s, 23.85 kbit/s) und MP3 (64 kbit/s, 80 kbit/s, 96 kbit/s, 128 kbit/s).

The tool enables its users to make quick, reproducible evaluations of QoS/QoE as a function of a set of selected impairment parameters. In a real VoIP environment such evaluations would not only involve the use of considerable hardware and software resources but would also be extremely time-consuming to boot. The network would also need to have a real-time emulator of impairment parameters, and the reproducibility of measurement results could never be guaranteed. That is why it pays to use numerical tools to determine QoS.

The numerical tool QoSCalc(VoIP) has been subjected to several series of tests. It delivers irrefutable and reproducible results which match those obtained in a real environment using Nextragen's QoS measurement systems [15]. It was used in the following point as well.

4.4 Comparison Study

The comparison study was conducted using the tool QoSCalc(VoIP) in the following representative scenario:

- Codec iLBC with sample length of 20 ms.
- Nondeterministic distributed packet loss (in accordance with the binomial distribution with the probability P) in range of 0–20 %.
- Exponentially distributed burst size with the average values from 1 to 10.
- Files from the ITU-T P.564 [31] as a reference signal.

$$Bpl = 0.0355 \cdot BSLP + 11.829$$

$$Ie = -0.0032 \cdot BSLP + 14.545$$

- 31 measurements for each determined performance value. In this way, it is possible to attain a confidence interval of less than 10 % of the estimated average.
- "Silence insertion" method whenever packet losses occur.
- PESQ, E Model and E(IP) Model as the QoE/QoS measurement methods.

Figures 5 and 6 show a selection of representative results from a large-scale study.

The curves in Fig. 5 show that QoS/QoE decreases exponentially irrespective of the QoS/QoE measurement technique when packet loss increases. The PESQ curve and the E^{IP} Model curve run very close to each other. The curve of the original

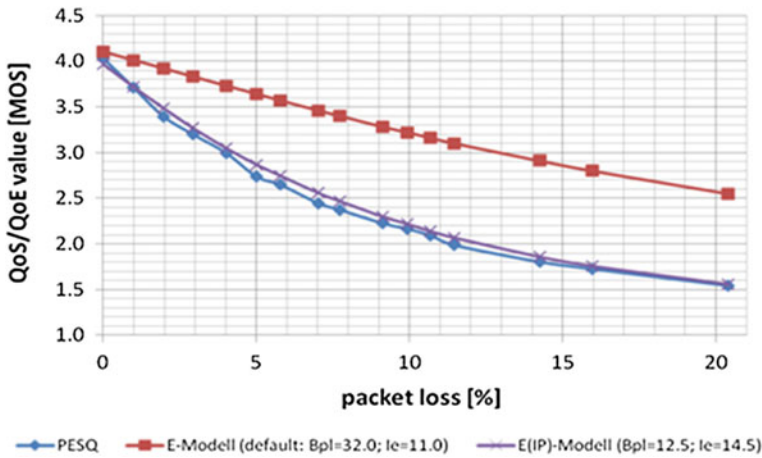


Fig. 5 QoS/QoE values as a function of packet loss with burst size equal to 1

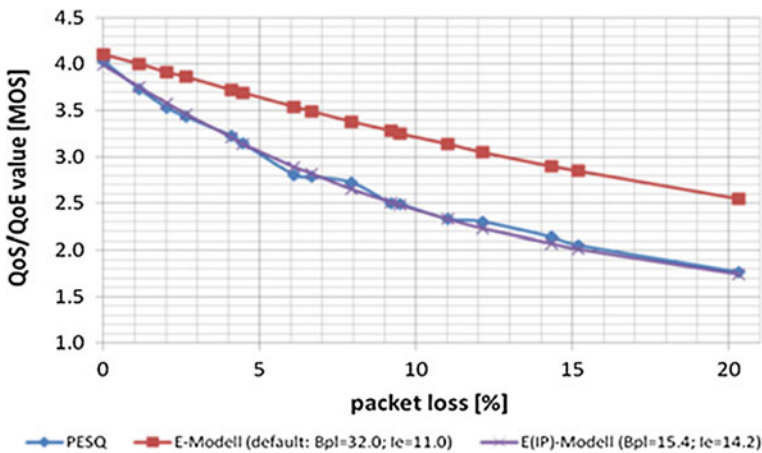


Fig. 6 QoS/QoE values as a function of packet loss with burst size equal to 5

E Model differs considerably. This scenario confirms the usability of the new E(IP) Model.

The curves in Fig. 6 show that burst sizes have noticeable effects on QoS/QoE. The curve for the original E Model deviates considerably from the PESQ curve whereas the E(IP) Model curve is very close to the PESQ curve. The new E(IP) Model has proved itself to be suitable in this scenario. Comparable results were obtained from further analyses using other speech codecs; however, they cannot be included here for lack of space.

In conclusion it can be said that the modification of the old E Model was overdue, and necessary to align the QoE values yielded by PESQ with those yielded by the E(IP) Model (QoS values). This can only be seen as a benefit in practice.

5 QoS/QoE in the VToIP/IPTV

5.1 Introduction

Figure 7 shows established methods for measuring the QoS/QoE of the video component of the VToIP/IPTV service. Alongside several signal-based international standards there are also a few parametrised measurement techniques that work without a reference signal, not the least of these being the parametrised VSoIP Model [32], that is to be classified as a dual-ended model. Its make-up and mode of operation correspond to those of the E(IP) Model for the speech service. The parametrised QoS models are quick and easy to use, which is of great benefit in practice. In contrast, it takes minutes to measure the QoE of a HD video file using, say, the PEVQ method. Besides, it is one of the groups of active (intrusive) methods of measuring QoE, which means that a connection must first be established in an IP environment, and a reference signal must be sent and mirrored at the receiver’s end. That is very time-consuming.

The next section describes and compares the most important QoS/QoE measurement techniques for the VToIP/IPTV service.

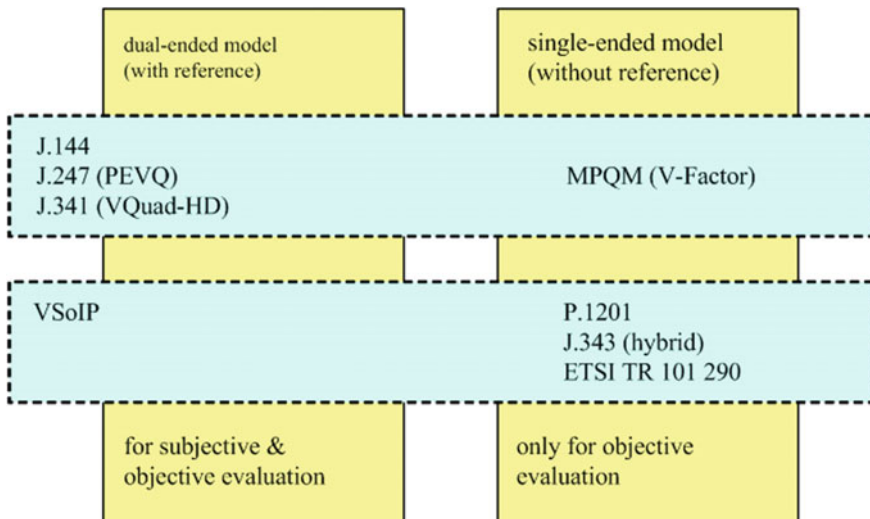


Fig. 7 Overview of QoS/QoE measurement techniques for the VToIP/IPTV service

5.2 QoS/QoE Measurement Techniques

5.2.1 PEVQ and VQuad-HD

PEVQ is designed to predict the effects of transmission impairments on the video quality as perceived by a test person. Its main application areas are mobile multimedia applications and IPTV. It fulfils the ITU-T Recommendation J.247 [33] for full reference quality measurements. The key features of PEVQ are:

- Temporal alignment of the input sequences based on multi-dimensional feature correlation analysis with limits that reach far beyond those tested by the Video Quality Experts Group (VQEG), especially with regard to the amount of time clipping, frame freezing and frame skipping which can be handled.
- Full frame spatial alignment.
- Colour alignment algorithm based on cumulative histograms.
- Enhanced frame rate estimation and rating.
- Detection and perceptually compliant weighting of frame freezes and frame skips.
- Only four indicators are used to detect the video quality. Those indicators operate in different domains (temporal, spatial, chrominance) and are motivated by the Human Visual System (HVS). Perceptual masking properties of the HVS are modelled at several stages of the algorithm. These indicators are integrated using a sophisticated spatial and temporal integration algorithm.

Figure 8 shows the block diagram of the Algorithm PEVQ. It works as follow:

- The first block (pre-processing stage) is responsible for the spatial and temporal alignment of the reference and the impaired signal. This process ensures that only corresponding frames are compared with each other.
- The second block calculates the perceptual difference of the aligned signal. Perceptual means that only those differences are taken into accounts which are actually perceived by a human viewer. Furthermore the activity of the motion in the reference signal provides another indicator that represents the temporal information. This indicator is important as it takes into account that in frame

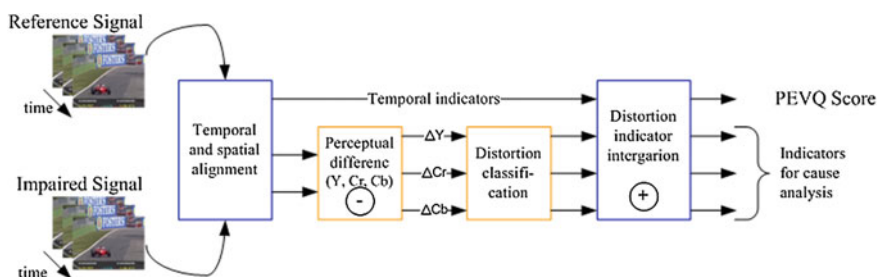


Fig. 8 Sequence diagram of PEVQ score calculation

series with low activity a viewer's perception of details is much higher than in frame series with rapid motion.

- The third block in Fig. 8 classifies the previously calculated indicators and detects certain types of distortions.
- In the fourth block all the appropriate indicators according to the detected distortions are aggregated, forming the final result—the mean opinion score (MOS). The MOS value describes the video quality on a range from 1 for very bad quality to 5 for excellent quality.

Apart from the MOS value, which is the ultimate yardstick of quality, PEVQ offers several other indicators that are used to analyse the reasons for quality impairments such as:

- Distortion,
- Delay,
- Luminance,
- Contrast,
- Peak signal to noise ratio,
- Jerking,
- Blurring,
- Block constriction,
- Frame freezing and frame skipping,
- Effective picture rate,
- Time and areal activity.

The VQuad-HD (Objective perceptual multimedia video quality measurement of HDTV) is the successor of PEVQ for video material in HD format and works according to the guidelines laid out in the ITU-T Standard J.341 [34]. This QoE technique predicts the video quality as it would be perceived by test persons in a sensory test. The prediction model uses psycho-visual and cognitive-inspired modelling to emulate subjective perception. As a full reference approach, the model compares the input or high-quality reference video and the associated degraded video sequence under test. This process is shown in Fig. 9.

The VQuad-HD score estimation is based on the following steps:

- First, the video sequences are preprocessed. In particular, noise is removed by filtering the frames, and the frames are subsampled.
- Temporal frame alignment of reference and processed video sequence is performed.
- Spatial frame alignment of processed video frame and its corresponding reference video frame is performed.
- Local spatial quality features are computed: local similarity and local difference.
- Measure, inspired by visual perception.
- Analysis of the distribution of the local similarity and difference feature is performed.
- Global spatial degradation is measured using a blockiness feature.

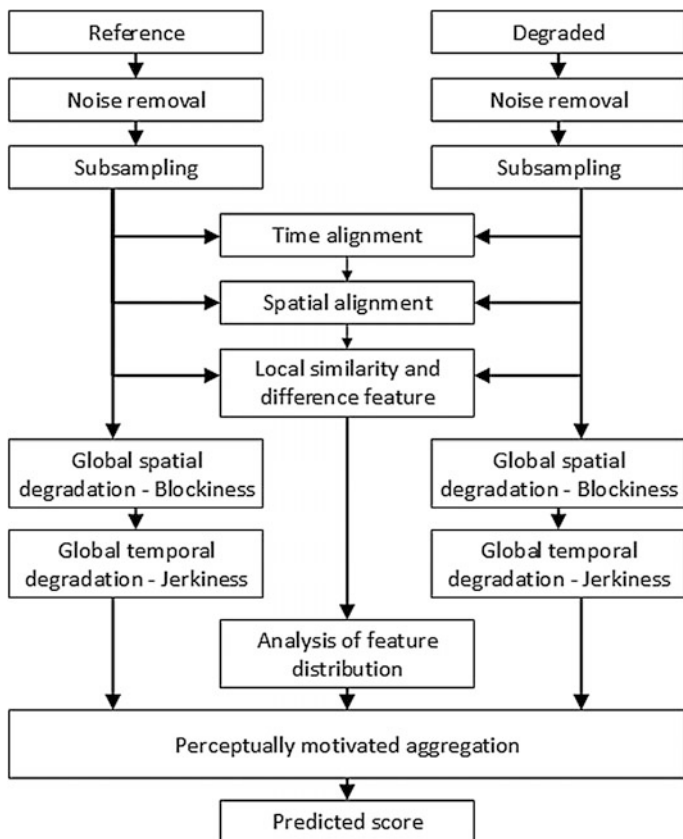


Fig. 9 Sequence diagram of VQuad-HD score prediction

- Global temporal degradation is measured using a jerkiness feature. The jerkiness measure is computed by evaluating local and global motion intensity and frame display time.
- The quality score is estimated based on a non-linear aggregation of the above features.
- To avoid misprediction whenever relatively large spatial misalignments of reference and processed video sequences occur, the above steps are computed for three different horizontal and vertical spatial alignments of the video sequence, the maximum predicted score among all spatial positions being the final estimated quality score.

Many factors need to be taken into consideration when selecting reference signals. These factors can be found in the ITU-T Tutorial [35] and in publications of the Video Quality Experts Group [36]. The video format requirements and recommendations regarding the algorithms and tools used state that the best results

will be obtained if the reference file is an uncompressed AVI (Audio Video Interleaving) file in the YUV 4:2:0 colour space. A short video sequence of around 10 s is ideal since the algorithms would take far too long to process longer sequences whilst the influence of network impairments in shorter sequences would hardly coax sufficiently meaningful responses from the algorithms. In Europe, full HD videos are usually in 1080i50, which means a resolution of 1920×1080 pixels at 25 full frames per second are ideal parameters for the reference signals. The reference signals should of course be free from distortions, errors and coding artefacts to preclude influences above and beyond network impairments.

A selection of reference files can be found in the consumer digital video library [37] or obtained from the license holders of the two measurement algorithms (Opticom [22] and SwissQual [23]). A detailed analysis which covers the selection of suitable reference signals for PEVQ and VQuad-HD can be found in the paper [38].

5.2.2 VSoIP Model

Figure 10 shows the newly established, parametrised VSoIP Model for determining the quality of video streams in the VToIP/IPTV services [32].

The model works on the following principle: All network impairments are collected and processed in the first block of the diagram. The effects of jitter and out-of-order packet delivery are converted into losses, bearing in mind that these errors can be smoothed with the aid of the jitter buffer. The values calculated in this block are passed on, along with the packet losses from the network, to the second block, where total losses and burst size are determined. The VSoIP Model implements the Markov property “memorylessness”, which is widely used in analyses of networks. The ensuing recalculated parameters are passed on to the third and final block. Further inputs for the third block include information about the codec type, encoding rate and TS type. The final output of the model is the IPTV factor [MOS].

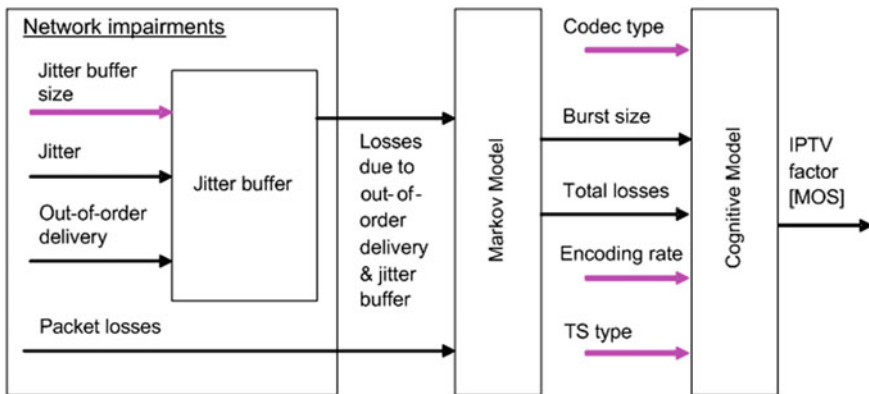


Fig. 10 Block diagram for the VSoIP model

the encoding rate and the type of transport stream that are being used. These data are gained from measuring the multicast streams. In the block called “Cognitive Model” the VSoIP factor is calculated and outputted as a value on the MOS scale. The mathematical dependencies needed to do this are stored in the block in the form of a table.

The VSoIP factor generally has the following form:

$$VSoIP\ factor = P \cdot e^{\frac{a \cdot packetloss}{burstsize}} + Q \cdot e^{\frac{b \cdot packetloss}{burstsize}} \quad (8)$$

All constants (P , Q , a , and b) are now calculated iteratively as best possible values for each encoding rate. The basis for this determination are the PEAQ curves, that were determined as a function of packet losses, burst size and encoding rate for various codecs and types of transport streams. Such curves can be determined using a suitable tool, for example the one described in point 5.3. The paper [32] contains full details on the procedure for determining the constants mentioned above.

In the example of VToIP in CIF format and for the H.263 Codec, the constants mentioned above are:

$$P = 3.54 \times 10^{-8} \times Bitrate^2 - 3.45 \times 10^{-4} \times Bitrate + 2.39 \quad (9)$$

$$Q = -7.02 \times 10^{-15} \times Bitrate^4 + 1.36 \times 10^{-10} \times Bitrate^3 - 9.66 \times 10^{-7} \times Bitrate^2 + 3.02 \times 10^{-3} \times Bitrate - 0.51 \quad (10)$$

$$a = -7.00 \times 10^{-10} \times Bitrate^2 + 8.00 \times 10^{-6} \times Bitrate - 2.39 \times 10^{-2} \quad (11)$$

$$b = 3.68 \times 10^{-11} \times Bitrate^3 - 5.23 \times 10^{-7} \times Bitrate^2 + 1.94 \times 10^{-3} \times Bitrate - 2.80 \quad (12)$$

Here, too, there are two methods of conducting an evaluation of QoS: (a) by making measurements in the real environment and (b) by emulation in a suitable numerical tool. The next point presents such a tool.

5.3 The Tool QoSCalc(VSoIP)

Figure 11 is a block diagram representing the concept behind the tool designed to examine QoS/QoE in the VToIP/IPTV service. To be exact, it determines the quality of the video streams in the VToIP/IPTV service. That is why it came to be called QoSCalc(VSoIP); VS standing for *video streaming* [39, 40].

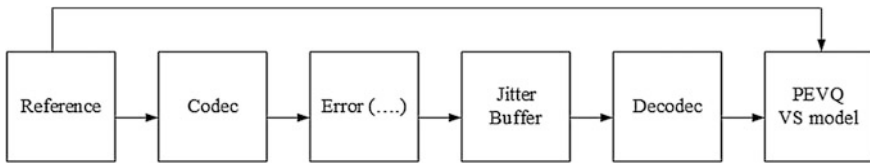


Fig. 11 Block diagram of the tool QoSCalc(VSoIP)

The following explains step for step how the tool works:

- First, the reference file is loaded.
- The reference signal is encoded in conformity with the video codec selected.
- The encoded video samples are segmented and encapsulated in the transport streams.
- The impairment parameters in the network (jitter, packet losses, or both) are emulated in the block “Error”.
- The transport packets containing the video samples are received and buffered in the jitter buffer, where they are processed. Whenever a packet loss occurs, the packet is discarded.
- Once decapsulated, the video samples are decoded in the next block according to the codec selected.
- Finally, the received signal and the reference signal are passed on to the PEVQ algorithm, which then calculates the QoE as values on the MOS scale. It is also possible to pass the calculated parameters to the VSoIP Model as soon as the RTP packets are received because either of this model is capable of determining and outputting QoS values on the MOS scale.

The Tool QoSCalc(VSoIP) was developed using the programming language C Sharp. All of the steps outlined in the previous paragraph are performed on the computer. The transmission of transport packets is done virtually within the tool itself.

The following codecs are currently supported: H.263, H.263+, MPEG-2 and MPEG-4 with different Presets, i.e. ultrafast, superfast, very fast, fast, slow, slower very slow. The tool allows the following formats to be selected: CIF, QCIF, QVGA, QQVGA, 720p, 1080p. The following techniques are provided for the packing and formation of transport streams: RPT, native RPT and MPEG2-TS.

The numerical tool QoSCalc(VSoIP) has been tested in a number of analyses. It delivers reliable and reproducible results. The next point describes how the tool is used.

5.4 Comparison Study

The comparison study was carried out with the aid of the QoSCalc(VSoIP) tool in the following, typical scenario of the VToIP service:

- Nondeterministic distributed packet losses of 0–20 % and constant burst size of 1 at an encoding rate of 1702 kbit/s.
- Nondeterministic distributed packet losses of 0–20 % and nondeterministic burst size of 2 at an encoding rate of 1702 kbps.
- Nondeterministic distributed packet losses of 0–20 % and constant burst size of 1 at an encoding rate of 4978 kbps.
- Nondeterministic distributed packet losses of 0–20 % and nondeterministic burst size of 2 at an encoding rate of 4978 kbps.
- Video codec H.263 (see Eqs. 8–11).
- Image format CIF.
- Image refresh rate of 25 images/s.
- 31 measurements per value of each variable (here: packet loss). This ensures that confidence intervals are achieved that are less than 10 % of the mean values under analysis (with a probability of error of 5 %).
- PEVQ and VSoIP Models as the QoE/QoS measuring techniques.
- An AVI file from the company Opticom [22] was chosen as the reference video. The file is 8 s long and a resolution of 352×288 . Figure 12 shows a screenshot of the reference video.

The results of the comparison study are presented graphically in Figs. 13, 14, 15 and 16. Figures 13, 14, 15 and 16 show that QoS deteriorates exponentially as packet losses increase. This is the case for both QoE/QoS measuring techniques used. Furthermore, the curves fall less steeply as burst size increases. The reason for this is that synchronisation of I/P/B images fails more frequently when small groups of packet losses regularly occur than when large groups of packet losses occur infrequently, and the more numerous the breakdowns in synchronisation are, the more frequently the images will freeze. This will naturally be reflected in a drop in QoE/QoS values.

Figures 13, 14, 15 and 16 also show that the curves of the PEVQ and VSoIP Models progress very closely to each other. This behaviour was also observed in further analyses, which unfortunately cannot be included here for lack of space (cf. results in e.g. [32]). In summary: the numerical comparison study has delivered strong arguments for using VSoIP Model in everyday practice.



Fig. 12 Screenshot of the reference video

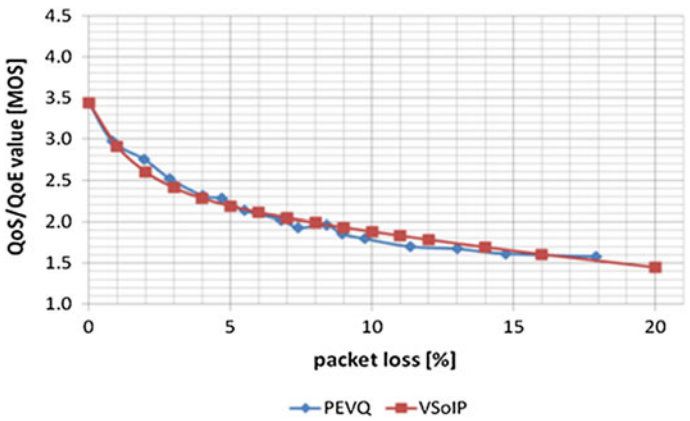


Fig. 13 QoS values as a function of packet losses gained from different measuring methods for the Codec H.263, the image format CIF, burst size 1 and an encoding rate of 765 kbps

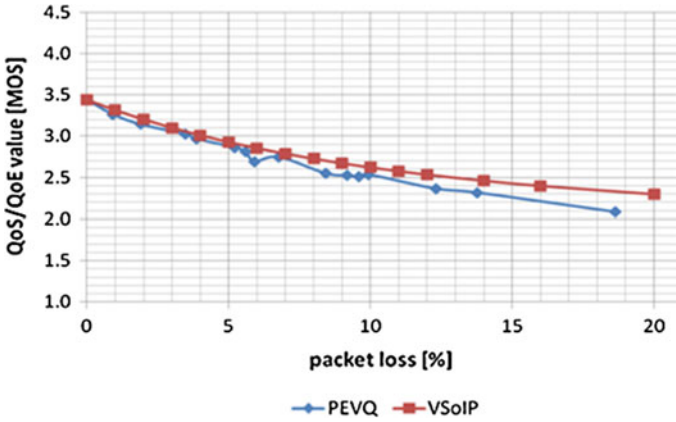


Fig. 14 QoS values as a function of packet losses gained from different measuring methods for the Codec H.263, the image format CIF, burst size 5 and an encoding rate of 765 kbps

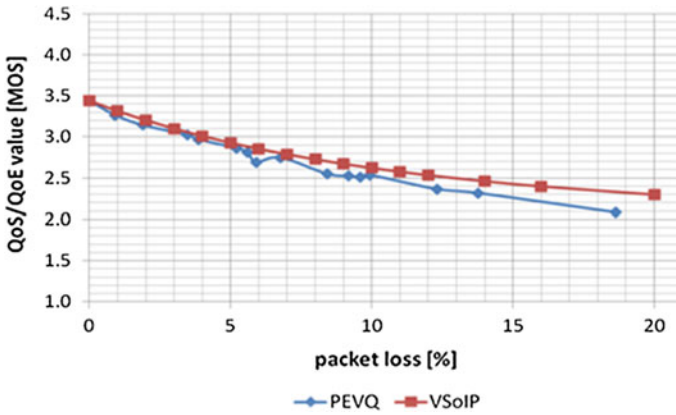


Fig. 15 QoS values as a function of packet losses gained from different measuring methods for the Codec H.263, the image format CIF, burst size 1 and an encoding rate of 2167 kbps

6 QoS/QoE in the WWW Service

6.1 Introduction

Figure 17 shows the most widely known techniques currently used to measure QoS/QoE in the WWW service.

It is immediately clear that there is only one standardised technique for measuring QoE in the WWW service: G.1030 [41]. There are two other QoS techniques in the single-ended model—Apdex Index [42] and Power [43]—but they have not

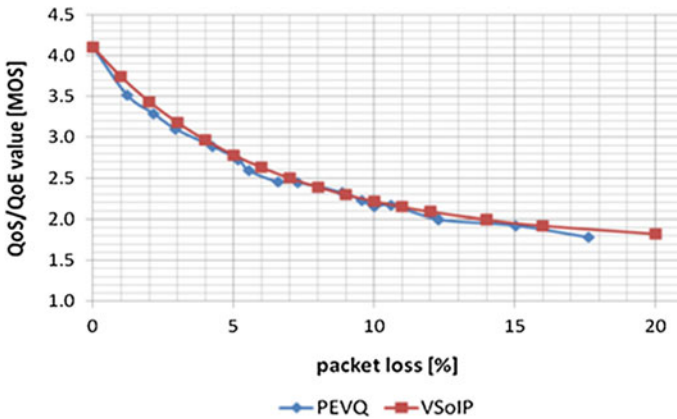


Fig. 16 QoS values as a function of packet losses gained from different measuring methods for the Codec H.263, the image format CIF, burst size 5 and an encoding rate of 2167 kbps

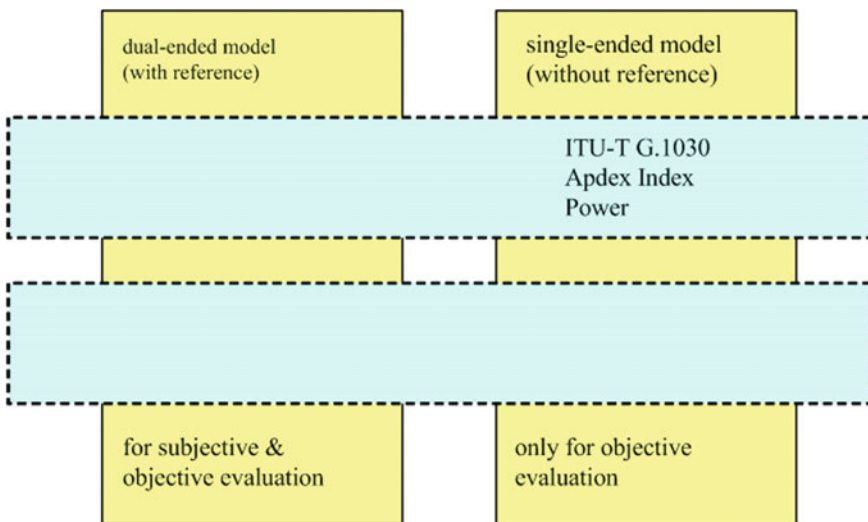


Fig. 17 Overview of QoS and QoE measurement techniques for the WWW service

been standardised, meaning that there is an enormous need for further developments in this area, especially in view of the fact that the WWW service is one of the most widely used applications in the modern Internet of Things and accounts for the lion’s share of traffic.

The three techniques for measuring QoS/QoE in the WWW service shown in Fig. 17 will be described in the next section and compared in an analysis that then follows.

6.2 QoS/QoE Measurement Techniques

6.2.1 Standard ITU-T G.1030

The ITU-T Standard G.1030 [41] covers the relationships between the parameters of a service and subjective appraisal of the quality of that service by the end-user. So it is, in fact, a measurement of the quality of experience (QoE). The chief parameter under consideration is the time taken to open a web page. Various threshold values are used to assess the quality of service subjectively:

- (a) Threshold value 0.1 s—maximum value for the reaction time without any impairment to communication.
- (b) Threshold value 1 s—maximum value of the reaction time without impairment to the smooth operation of the application.
- (c) Threshold value 10 s—maximum value for the time which can elapse during the operation of an application without the user becoming frustrated.

For the measurement of QoE covered in the Recommendation several time slots are taken into account, as Fig. 18 shows.

The G.1030 Standard includes a description of an experiment in which a distinction is made between two groups of test persons: (a) test persons with considerable experience and (b) test persons with little or no experience. The information supplied by G.1030 allows one to make the following observations:

- (a) Both groups reacted similarly, identifying in their subjective judgements an inversely proportional relationship between service parameters and quality of service.
- (b) Persons with little or no experience were more critical than persons familiar with the WWW service.
- (c) As sessions grow longer (in excess of 60 s) the discrepancies between the judgements given by the two groups begins to diminish.

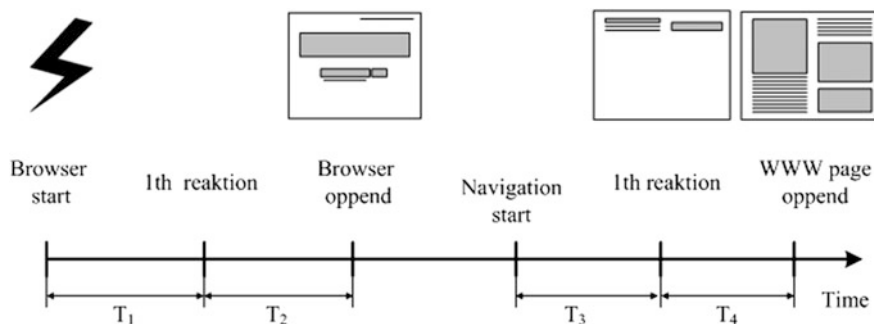


Fig. 18 Time slices in the determination of QoE according to G.1030

The impairment parameter under scrutiny here—whether it is being measured or appraised—is the time taken for a web page to open. No account is made of local conditions and parameters. Moreover, this kind of QoS/QoE analysis is extremely time-consuming and resource-consuming. Consequently, this measurement technique stands little chance of being used successfully in practice.

6.2.2 Apdex Index

The Apdex Index Method [42] is a proprietary solution that has yet to be standardised. It, too, is a method for measuring both QoE and QoS in the WWW service. It, too, is a single-ended measuring model. It can be applied without recourse to a group of test persons, but only if specialised tools are then assigned to determine the parameters necessary for this metric. Its chief criterion of evaluation is also the time taken for a web page to open. The assessment of the quality of service yields a value on a scale from 0 (unacceptable) to 1 (excellent). The assessment depends to a large extent on which threshold time T is used. All relationships (and the metric itself) are clearly shown in Fig. 19.

Steps 1 to 6 have the following meanings:

1. Define threshold T (in seconds) for the application.
2. Real-time expert reporting group—View by Application, Server, or User.
3. Extract data set from existing measurements for viewing.
4. Count the number of samples in the three performance zones.
5. Calculate the Apdex value (see $Apdex_T$ formula).
6. Display Apdex result.

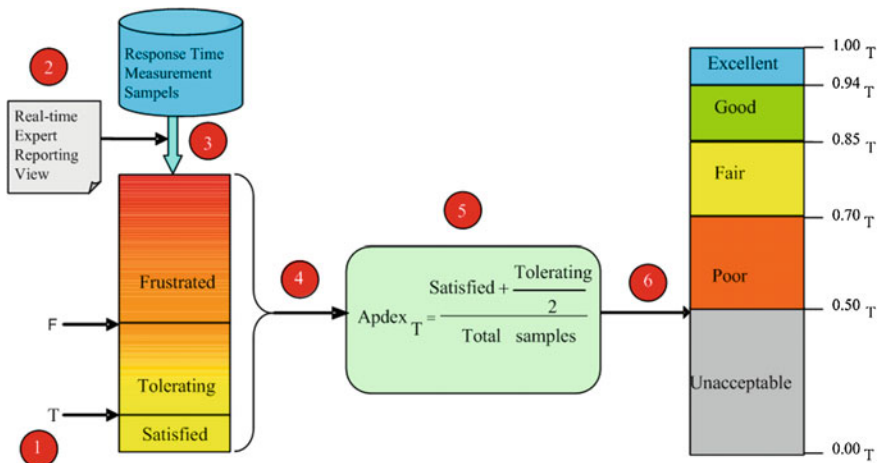


Fig. 19 Overview of the Apdex process

The value “T” is the application target time (threshold between satisfied and tolerating users). The threshold “F” between tolerating and frustrated users is calculated as $F = 4T$. This method can be used in combination with specialised tools in practice.

6.2.3 Metric Power

The metric Power was first described in paper [43]. It is a further method of measuring both QoE and QoS that is to be classed as a single-ended model. Unlike the methods described above this method takes a further parameter into account that can be crucial for the WWW service: data download rate. As a result, the method is more versatile and therefore of much more practical use. Equation (13) describes the metric Power.

$$P = \frac{1}{1 + \frac{\alpha}{ddr}} \quad (13)$$

where:

α delay coefficient

$$\alpha = \begin{cases} 0 & \text{for } td \leq th \\ (td - th) & \text{for } td > th \end{cases}$$

ddr download data rate in (Mbps)

td total delay in (s)

th threshold of delay in (s)

It is obvious that both the download data rate and the total delay (time lapse between the point in time at which the session begins and the point in time at which the Web page is completely built up) depend on the throughput of the transmission canal used. But they also depend on the locality of the WWW server, its level of activity, on the content of the Web pages being accessed, and whether that content is static or dynamic. Total delay can therefore vary within a considerable range. The new metric Power takes that into account.

Actual practical application of the QoS measurement techniques described above requires high-performance tools that could feasibly also be implemented in the real environment. The next section presents such a tool in quite some detail.

6.3 The Tool QoSCalc(WWW)

The tool QoSCalc(WWW) was first presented in paper [44] published in 2014. The arrows in Fig. 20 explain how the new tool works, the single steps being:

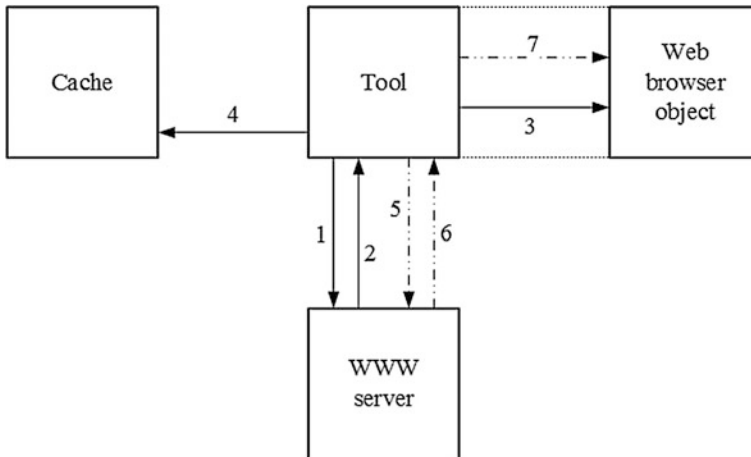


Fig. 20 Principle of operation of the QoSCalc(www) tool

- (1) First navigation phase
- (2) First data download phase
- (3) Page set-up in web browser object
- (4) Storage in computer cache
- (5) Start of a complete page refresh
- (6) Refresh data download phase
- (7) Refresh page set-up in web browser object

The first navigation phase and the first data download phase assist in timing the start of the measurement. During these phases the tool checks whether outsourcing is being used for the selected ASP, or not. If outsourcing is being used, the tool then looks to see how many WWW servers are being used to store elements of the contents that are being called up. The fact that the called page must always be built up from scratch in the web browser object can be used to good effect when it comes to determining QoE. When the measurement has been completed, the tool displays—without any further action on the part of the user—the QoS values calculated in accordance with the metrics Power and Apdex.

The reliability of the tool QoSCalc(www) was tested in exhaustive trials. Its operability was then further tested in a real-life environment. This environment and the measurement scenarios operating within it are described in the next chapter.

6.4 Comparison Study

QoSCalc(www) was implemented in a real IP environment. To emulate the effects of network impairment parameters a wanulator [45] was switched between the tool and the Internet. The tool was used in two scenarios [44]:

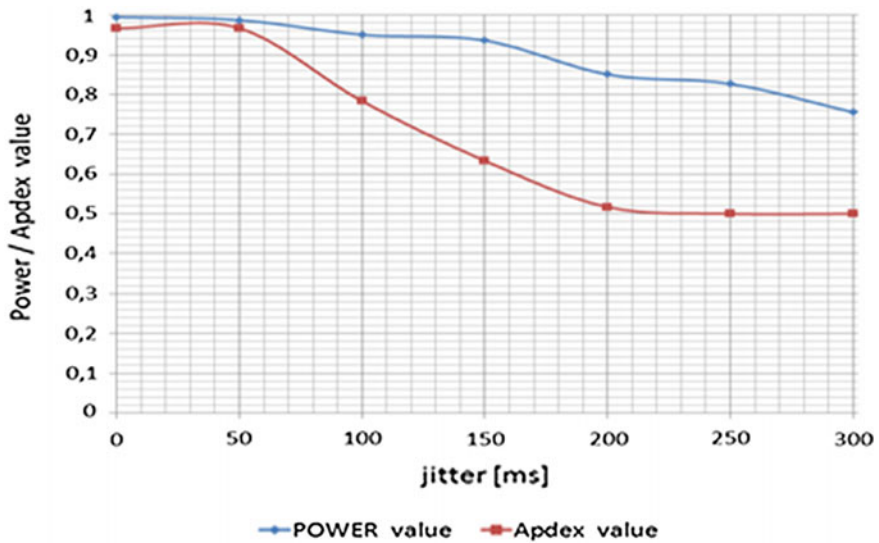


Fig. 21 POWER and APDEX values as a function of jitter

Scenario 1:

- Threshold “POWER”: 2 s.
- Threshold “APDEX”: 2 s.
- Interval between measurements: 3 s.
- Number of measurements: 30.
- Jitter in ms [0, 50, 100, 150, 200, 250, 300].

Scenario 2:

- Threshold “POWER”: 5 s.
- Threshold “APDEX”: 5 s.
- Interval between measurements: 3 s.
- Number of measurements: 30.
- Packet Loss in % [1–4, 6, 8, 10].

Representative results from Scenario 1 (with <http://www.google.de>) are presented graphically in Fig. 21.

A comparison between Apdex and Power shows that the two QoS curves develop exponentially but with differing gradients: at a jitter of 50 ms the Apdex curve starts to fall rapidly. The metric Power exhibits much more elasticity, with the subjective appraisal corresponding more to the objective result yielded by Power. While the measurements were being made, it also became evident that the ASP for the webpage <http://www.google.de> uses no outsourcing, which means that the call-up time of approx. 2 s is relatively short.

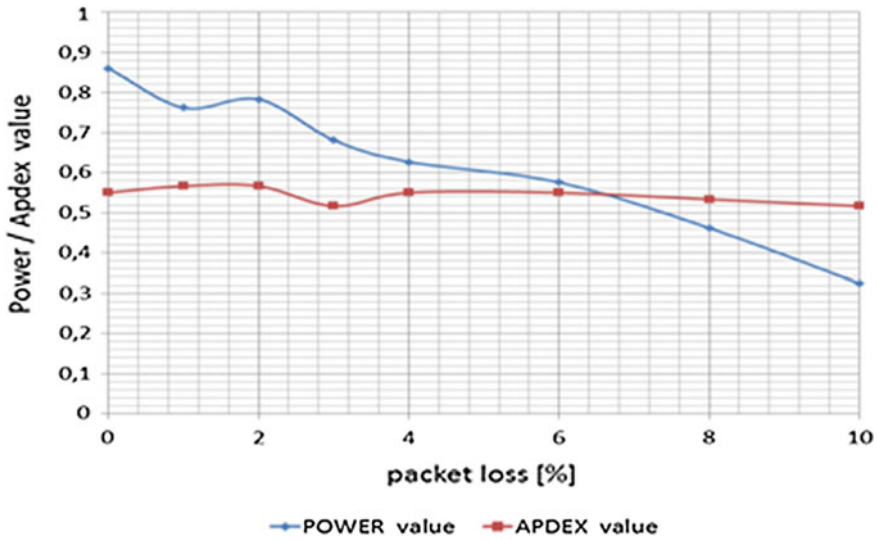


Fig. 22 POWER and APDEX values as a function of packet loss

Representative results from Scenario 2 (with <http://www.spiegel.de>) are presented graphically in Fig. 22.

From Fig. 22 it is evident that even in a loss-free environment the times taken for a page to open were above the threshold set at 5 s, which resulted in values of less than 1. The time taken is so excessive because the ASP for the Web site <http://www.spiegel.de> uses outsourcing, which is known to take time. The Apdex values remain almost constant at approx. 0.55. The metric Power demonstrates favourable elasticity, too.

The results obtained here show that the goal of creating an innovative, efficient tool for evaluating the quality of the WWW service has been achieved. The tool operates dependably and can be implemented practically anywhere. Consequently, the new tool is of immense practical importance for the communications industry.

7 Summary and Conclusion

This chapter has reviewed and discussed the application of common QoS/QoE measuring methods in Triple Play Services (audio/video/data). Examples of the technologies in VoIP, VToIP/IPTV, WWW applications have been examined to demonstrate how they work. The results have been presented graphically, and interpreted. The strengths and weakness of the individual QoS/QoE measurement techniques have been spelt out. In practice it is highly beneficial to work with parametrised QoS models. They, too, have been presented in this chapter.

In practice, QoS/QoE measurements should be made at regular intervals and, best of all, completely automatically. Measurement Management Systems (MMS) could take care of that. They are currently undergoing intensive development. Late 2015 saw the appearance of the monitoring measurement systems operated by the Polish network regulatory agency UKE [46] and by the German network regulatory agency BNetzA [47]. Both measurement systems were designed to monitor the network parameters (cf. Table 1) at the access interface to the broadband Internet, and that is where they have been implemented. The measurements are carried out completely independently of the network provider. Ultimately, the measurement results will be published on purpose-designed web pages of the two network regulatory agencies, thus ensuring transparency and neutrality of the telecommunications market.

The next step will be to extend monitoring to the QoS of the various applications available in the Internet of Things. That will present a further challenge to measurement technology. Paper [48] describes a preliminary, deliberately generally formulated concept for an MMS that will ultimately achieve this. The latest EU research projects (e.g. Leone [5] and mPlane [6]) and the IETF's Framework for Large-Scale Measurement of Broadband Performance (LMBP [7]) pursue that goal.

Recently it has become possible to shift the measurement of QoS/QoE into the cloud. There are various approaches to establishing a measurement system in the cloud, depending on where the measurements are actually to be made and where the results are to be collected. There are three different approaches: (a) evaluation in the cloud, (b) measurements with the cloud and (c) measurements in the cloud. These three approaches are discussed in length in paper [49].

To summarise: a great deal of development work and practical implementation remain to be done in the field of QoS/QoE. New scientific concepts for QoS/QoE measurement techniques and systems are needed. Designers and engineers are faced with a mighty challenge!

References

1. Definition of NGN. Information available at (access: May 2016): <http://www.itu.int/rec/T-REC-Y.2001>.
2. Report TD 109rev2. Information available at (access: May 2016): <http://www.itu.int/en/ITU-T/studygroups/2013-2016/12/Pages/default.aspx>.
3. DIRECTIVE 2009/140/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 25 November 2009 amending Directives 2002/21/EC (Official Journal EU L.337/37).
4. DIRECTIVE 2009/136/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 25 November 2009 amending Directive 2002/22/EC (Official Journal EU L 337/11).
5. Leone project: From global measurements to local management. Information available at (access: May 2016): <http://www.leone-project.eu/> leone.
6. mPlane: An Intelligent Measurement Plane for Future Network and Application Management. Information available at (access: May 2016): <http://www.ict-mplane.eu/> mPlane.

7. Large-Scale Measurement of Broadband Performance. Information available at (access: Mai 2016): <https://tools.ietf.org/wg/lmap>.
8. D. Felsmann, Cloud Computing—Basistechnologie, Architektur, Erfolgsfaktoren, Herausforderungen und die aktuelle Marktsituation (in German). GRIN Editor, 2010.
9. CEPT's ECC report 195. Information available at (access May 2016): <http://www.erodocdb.dk/Docs/doc98/official/pdf/ECCREP195.PDF>.
10. ETSI Guide EG 202 057. Information available at (access May 2016): <https://www.google.de/#q=ETSI+Guide+EG+202+057+>.
11. ITU-T Recommendation Y.1541. Information available at (access May 2016): <https://www.google.de/#q=ITU-T+Recommendation+Y.1541++>.
12. ITU-T Recommendation G.1010. Information available at (access May 2016): <https://www.google.de/#q=ITU-T+Recommendation+G.1010+>;
13. A. Raake, Speech Quality of VoIP. John Wiley&Sons, Chichester, 2006.
14. Patent no. 102010044727. Information available at (access May 2016): <http://www.dpma.de>.
15. Company Nextragen. Information available at (access May 2016): <http://www.nextragen.de>.
16. Tool Trace_View. Information available at (access May 2016): <http://www.nextragen.de/produkte/traceview/?L=%3FL%3D>.
17. ITU-T Recommendation P.862. Information available at (access May 2016): <http://www.itu.int/rec/T-REC-P.862-200102-I-P.862>.
18. ITU-T Recommendation P.862.1. Information available at (access May 2016): <http://www.itu.int/rec/T-REC-P.862.1/en>.
19. ITU-T Recommendation P.862.2. Information available at (access May 2016): <http://www.itu.int/rec/T-REC-P.862.2/en>.
20. ITU-T Recommendation P.863. Information available at (access May 2016): <http://www.itu.int/rec/T-REC-P.863-201101-P-P.863>.
21. Nextragen GmbH, White Paper: Einsatz von PESQ und POLQA in VoIP-Umgebungen: Ein Vergleich (in German), 2011.
22. Company Opticom. Information available at (access May 2016): <http://www.opticom.de>.
23. Company SwissQual. Information available at (access May 2016): <http://www.swissqual.com>.
24. Company TNO. Information available at (access May 2016): <http://www.tno.nl>.
25. ITU-T Recommendation P.830. Information available at (access May 2016): <http://www.itu.int/rec/T-REC-P.830/en>.
26. S. Paulsen and T. Uhl, Quantifying the Suitability of Reference Signals for the PESQ Algorithm. Proceedings of the CTRQ Conference 2010 (CD form), Athens/Greece, Juno 2010.
27. ITU-T Recommendation G.107. Information available at (access May 2016): <http://www.itu.int/rec/T-REC-G.107/en>.
28. B. Kolbe and T. Uhl, Messung und quantitative Bewertung der Qualität des Dienstes VoIP (in German). Journal PIK (Praxis Informationsverarbeitung und Kommunikation), 32 (3), 2009, pp. 183–187.
29. S. Paulsen and T. Uhl, Numerisches Tool zur Untersuchung der QoS bei VoIP (in German). Proceeding of the 7th Workshops MMBnet2013, Hamburg, September 2013, pp. 85–90.
30. ITU-T Recommendation G.113. Information available at (access May 2016): <http://www.itu.int/rec/T-REC-G.113-200711-I>.
31. ITU-T Recommendation P.564. Information available at (access May 2016): <https://www.itu.int/rec/T-REC-P.564>.
32. T. Uhl and H. Jürgensen, The new, parameterized IPTV Model for Determining Quality of Video Streaming in the IPTV Service. Bulletin of the Polish Academy of Sciences Technical Sciences, 63(2), 2015, pp. 495–500.
33. ITU-T Recommendation J.247. Information available at (access May 2016): <http://www.itu.int/rec/T-REC-J.247-200808-I>.
34. ITU-T Recommendation J.341. Information available at (access May 2016): <http://www.itu.int/rec/T-REC-J.341-201101-II>.

35. ITU-T Tutorial about reference signals. Information available at (access May 2016): http://www.itu.int/ITU-T/studygroups/com09/docs/tutorial_opavc.pdf.
36. Video Quality Experts Group. Information available at (access May 2016): <http://www.its.bldrdoc.gov/vqeg/projects/hdtv/hdtv.aspx>.
37. Consumer digital video library. Information available at (access May 2016): <http://www.cdvlib.org/login.php>.
38. Ch. Hoppe, R. Manzke, M. Rompf and T. Uhl, Quantifying the suitability of reference signals for the video streaming analysis for IPTV. *Journal of Telecommunications and Information Technology*, no. 1/2016, ISSN 1509-4553, pp. 29–36.
39. T. Uhl, S. Paulsen and K. Nowicki, New Tool for Examining QoS in the VToIP Service. *Journal of Telecommunications and Information Technology*, no. 1/2014, ISSN 1509-4553, pp. 10–15.
40. T. Uhl and H. Jürgensen, New Tool for Examining QoS in the IPTV Service. *Proceeding of the World Telecommunications Congress*, VDE Editor GmbH, ISBN 978-3-807-3602-7, Berlin/Germany, June 2014.
41. ITU-T Recommendation G.1030. Information available at (access May 2016): <http://www.itu.int/rec/T-REC-G.1030/en>.
42. Apdex Index. Information available at (access May 2016): <http://www.apdex.org/specs.html>.
43. T. Uhl, J. Klink and P. Bardowski, New Metric for World Wide Web Service Quality. *Journal of Telecommunications and Information Technology*, no. 2/2014, ISSN 1509-4553, pp. 50–58.
44. T. Uhl and M. Rompf, New Tool for Investigating QoS in the WWW Service. *Journal of Telecommunications and Information Technology*, no. 1/2015, ISSN 1509-4553, pp. 1–7.
45. The tool Wanulator. Information available at (access May 2016): <http://wanulator.de>.
46. Office UKE. Information available at (access May 2016): <https://www.uke.gov.pl>.
47. Office BNetzA. Information available at (access May 2016): http://www.bundesnetzagentur.de/DE/Home/home_node.html.
48. J. Klink, J. Podolska and T. Uhl, Concept for a Measurement Management System for Access Service to the Internet. Paper accepted to publish in the *Journal of Telecommunications and Information Technology*, 2016.
49. H. Jürgensen and T. Uhl, Auswertung von QoS-Messdaten in einer Cloud-Umgebung (in German). *Proceeding of the 8th Workshops MMBnet2015*, Hamburg, September 2015, pp. 75–82.

Part III
Applicability of Interconnecting Everything

Integration of Internet of Everything (IoE) with Cloud

Sarbani Roy and Chandreyee Chowdhury

Abstract This chapter presents a roadmap of key developments in IoT-Cloud research in the context of different application domains and its applicability to IoE. IoT can be extended to IoE, if it is possible to connect everything to the Internet. Different layers of IoT protocol stack are discussed in this chapter. In IoT, physical objects connected to the Internet generate huge amount of data. Here, one of the main challenges is to move these data from the underlying IoT to the cloud. Cloud technologies are appealing due to the fact that the requirements for developing such IoE environment match very closely what cloud can offer in terms of computational and storage resources. Various application areas and existing works are also discussed.

1 Introduction

The Internet of Things (IoT) is considered as an expansion of the Internet to the physical world objects. In the IoT paradigm, these objects are provided with unique identifiers and connected into networks to transfer data for the purpose of information exchange [1]. Through unique addressing schemes variety of things can be able to interact and cooperate with each other to achieve a common goal. For example, mobile phones, sensors, actuators, embedded devices, RFID tags, etc., together can form an IoT environment [2]. The real world and the virtual world are converging to create smart environments that can make anything smart i.e., smart X e.g., energy, transport, cities and many other areas more intelligent. The devices in such applications need to show a huge variety of sensing and actuation capabilities as well as information processing functionalities. Figure 1 shows an IoT application road-map where smart X signifies IoT enabled smart services.

S. Roy (✉) · C. Chowdhury

Department of Computer Science and Engineering, Jadavpur University, Kolkata, India
e-mail: sarbani.roy@cse.jdvu.ac.in

C. Chowdhury
e-mail: chandreyee@cse.jdvu.ac.in

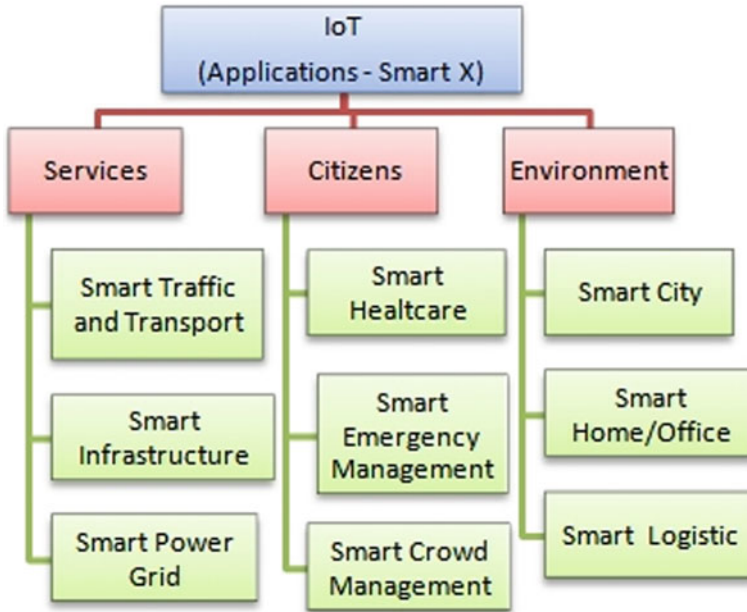


Fig. 1 IoT-an application roadmap

In IoT, the main objective is to provide full interoperability of interconnected devices. This is possible, only by enabling adaptation and autonomous behavior of participant devices, such that the devices appear with a higher degree of smartness in the underlying environment.

On the other hand, from the networking point of view, the IoT concept poses several new challenges. In reality, the things composing the IoT are generally resource constrained devices. These participant devices may be characterized by low resources in terms of both computation and energy capacity. Nowadays, smart handhelds have come up that are quite efficient in sensing tasks as well as computing. Thus, instead of providing complex infrastructure, smart devices from citizens could be utilized for their own benefit. However, to operate successfully, not only things like the sensors need to be connected to the Internet, but the data and services need also to be associated with the Internet. Consequently, it is mentioned in [3] that *forecasters predict the advent of an Internet of Everything (IoE)*. Processes and data, in addition to things and people, will all be part of this greatly expanded paradigm. However, the main challenge is to make everything connected to the Internet. If it is possible, then IoT can be easily extended to IoE. The pervasiveness of IoE can be utilized for the benefit of many people [4]. Cloud based mobile crowd-sensing is one aspect of it [5]. The larger user base can generate huge amount of data, making its storage and timely retrieval a challenging task. Redundant data and energy starvation are other issues related to scalability of such applications. Thus, cloud services can be utilized here to store and effectively control tasks. In [6],

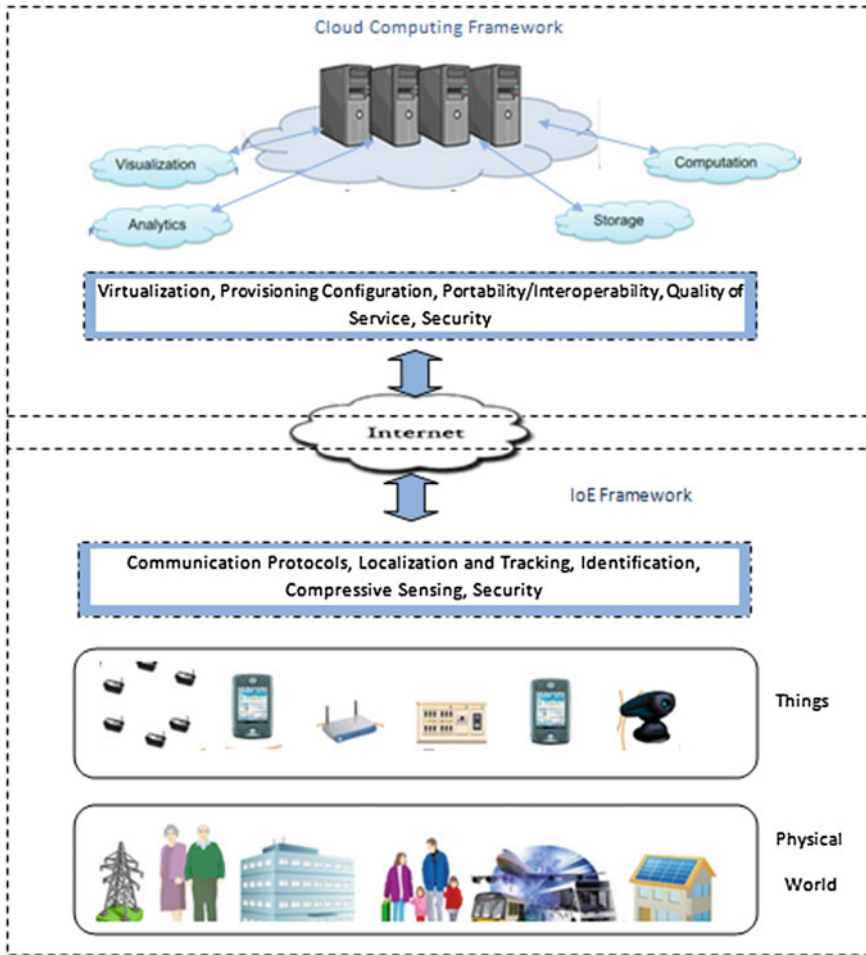


Fig. 2 Integration of IoT and cloud-conceptual view

a cloud compatible open source messaging system is introduced along with APIs that allows developers to write high performance IoT applications. Information about the devices could be provided to the applications that help in selecting the sensors based on application purpose and device status (available energy, location etc.). In [7], a publish/subscribe middleware is proposed based on cloud that filters out redundant data and/or selects minimum no. of devices for sensing/monitoring a particular area. Authors in [8] present network architecture and algorithms for optimal selection of multimedia content delivery methods in cloud.

Thus the integration of IoT and cloud is very essential to provide computational and storage infrastructure and support the development of services and applications beyond the limits of conventional IoT. One of the main objectives is to facilitate the

shift of data from IoT to the cloud computing environment so that the scientifically and economically valuable data may be fully utilized and properly analyzed. Thus, developers using IoT-Cloud can easily develop, compute intensive services and applications for data analysis and decision reporting. IoT-Cloud could be a strong pillar to extend smart applications to people at affordable cost. Figure 2 depicts the conceptual view of the IoT and cloud integration.

The architecture of IoT-Cloud encapsulates devices inside IoT and services in the cloud which satisfy the demands of a smart application. In this chapter, we assume that devices can be connected and accessed through the Internet and have sensing and actuation (whenever possible) capabilities.

The remainder of this chapter is organized as follows. An overview of IoT is presented in Sect. 2. The protocols mainly related to application layer are discussed in this section. The concept of cloud computing and the enabling technologies are discussed in Sect. 3. Section 4 reviews different applications of IoT-Cloud. Section 5 presents some research challenges in IoT-Cloud. Concluding remarks are given in Sect. 6.

2 Internet of Things (IoT)

In recent years, IoT has become an extension of the Internet. This presents a concept and a paradigm that considers the pervasive presence by allowing a variety of things or objects to connect anytime, anywhere, with anything and anyone using any network service. This is possible as the Internet is evolving into the communication medium for different types of things that are attached to the physical world. To create new applications or services, things of the physical world can interact and cooperate with each other through wireless or wired connections and unique addressing schemes [2]. These things can make them recognizable and also obtain intelligence by making or enabling context related decisions. Moreover, they can share knowledge and also access information that has been aggregated by other things in the environment. The coupling between such things and a worldwide standard-based communication infrastructure constitutes the concept of IoT and is characterized by machine-to-machine (M2M), machine-to-server (M2S) and server-to-server (S2S) communications [1]. Figure 3 depicts the communications at different layers of IoT.

This world stands to benefit the most from connecting everything like people, objects or things, data, etc. Nowadays, most of the times people are connected to each other through the social networks like Facebook, Twitter, LinkedIn via the Internet connection from their personal devices like smart-phones, tablets, laptops, PCs, etc. By default, any physical things which are equipped with sensors and/or actuators and connected to the Internet are part of the IoT. In IoT, these things such as Radio-Frequency IDentification (RFID) tags, sensors, actuators, mobile phones etc. sense the environment, generate data, become context-aware and finally after analysis can provide more experimental information to help human and machines to

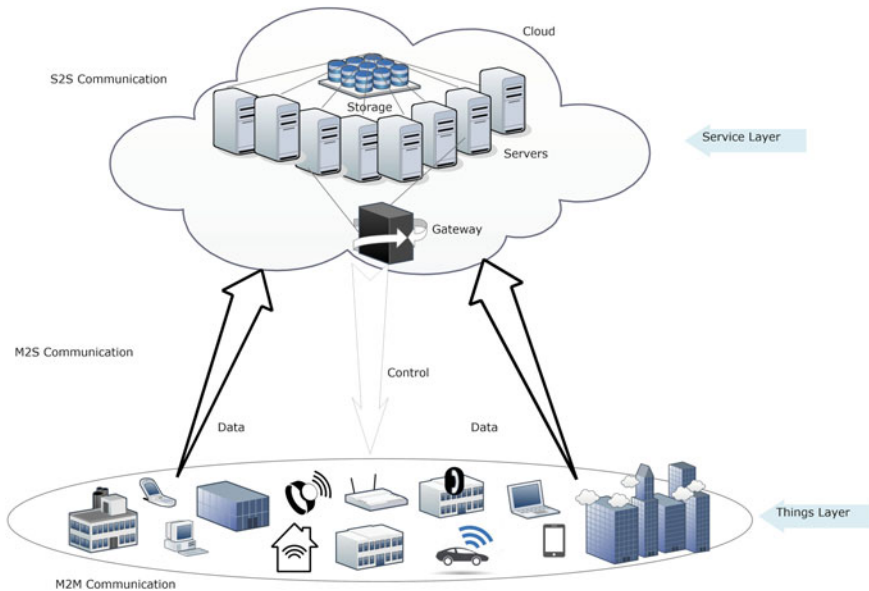


Fig. 3 Communications at different layers of IoT

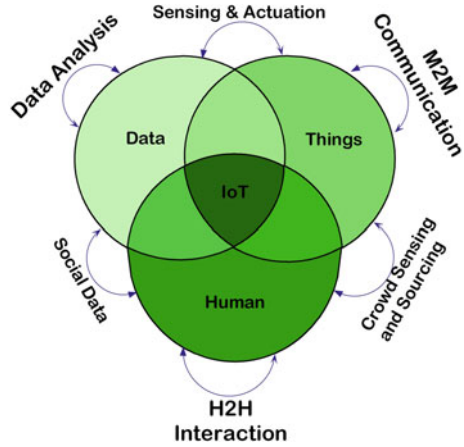
make more appropriate decisions. In the future, with the advancements in IoT, people and everything else may have the potential to become nodes on the Internet (i.e., IoE) and that will be very relevant and valuable for the evolving society.

Figure 4 depicts how the components of IoT—humans, things and data—interact with each other to enable smart environment. As depicted in Fig. 4, humans as well as things typically gather data and stream it over the Internet. Moreover, this process is not only limited to the reporting of raw data, connected things and people can also receive higher-level information back for further evaluation and decision making. Gathered data are then analyzed and as everything is connected to the Internet continue to advance and get the benefit of knowledge and information sharing. Thus the environment will become more intelligent by combining data into more useful information. This transformation from data to information in IoT is important because it will help the whole system to act fast as well as to take more intelligent decisions to control the underlying environment more effectively.

In IoT, things or smart objects augmented with sensing, processing, and network capabilities. Usually, they contain application logic that lets them make sense of their local situation and interpret the situation occurring within the underlying environment. These smart objects can act on their own and also intercommunicate with other machines (M2M) and exchange information with human users. Three main system-level characteristics of anything in IoT are:

- *Communication*: Smart objects or things must have the ability to communicate (wired or wireless) among themselves as well as with servers. These objects can form ad hoc networks of interconnected devices. WiFi, Bluetooth, IEEE

Fig. 4 IoT brings together human, things and data



802.15.4, Z-wave, and LTE-Advanced are some communication protocols used in IoT. Some other specific communication technologies like RFID, near field communication (NFC) and ultra-wide bandwidth (UWB) are also in use.

- *Identification*: In the network, these objects must be uniquely identified. Identification methods are used to provide a clear identity for each object within the network. This could be done using any tag mechanism or self description approach. The relationships among things can also be specified in the digital domain in the absence of physical interconnection. It is very important to distinguish between object's identification and address. Identification techniques along with the addressing schemes are used to uniquely identify objects in IoT. In [9], authors present architecture and mechanisms of Identifier (ID) layer for IoT.
- *Interaction*: These things can sense the surrounding environment and can also interact with the environment through their actuation capabilities whenever present.

Today's smart device infrastructure, mainly the applications are typically all about human interaction. Moreover, crowdsensing services are designed to provide specific information for its end-users. Also crowdsourcing activity is popular to foster peoples' participation. Due to the obvious reasons, human interaction can enhance data collection, analysis, and system behavior. Thus, human users' judgement (H2H) and input will make the IoT smarter.

The most important feature of the vision of IoT is that by observing the behavior of data collected from things and humans it will be possible to gain important insights. In IoT, connected and communicating devices (things) and interconnected human and their virtual social interactions are the big data source. Usually, things and humans in IoT are independent, and thus that collected data need to be aggregated together.

2.1 Overview of IoT Protocol Stack

This field is based on the paradigm of supporting the Internet protocol to all edges of the Internet. In reality, it is quite a complicated task to extend the support of IP stack to the small devices at the edge of the network. The IP stack is the center of the Internet and it can encapsulate many protocols. In IoT, the use of IP technology is fundamental as it allows for systems interoperability. It is also possible to build an IoT system with existing web technologies. However, it may not be very efficient. The protocol selection for the application critically depends on what type of things are part of that IoT application. The low-power devices in IoT may run entirely on battery and frequently exchange data over lossy networks. Thus, the major challenge is to manage these devices efficiently such that the system acts properly. Most of the existing protocols in the TCP/IP protocol stack such as HTTP and TCP are not good choices for such low-power communication due to overheads involved. However, devices which are directly connected to the Internet must use the IP suite to be able to exchange data with other devices (i.e., M2M communication) and servers (i.e., M2S communication) over the Internet. On the other hand, devices that form a local network must connect to a gateway. The local network can be based on any one of many different technologies, but the gateway can translate that communication protocol to Internet protocol.

This is an application layer gateway because it handles the data coming in from the local network and restructures it with a TCP/IP stack to enable communication with a device or service in the Internet. Figure 5 shows the comparison of Internet and IoT protocol stacks.

As compared to traditional networks, the IoT poses new challenges. Due to resource constraints and limited capability of the devices, communication and computation decisions must be taken judiciously. While developing any application on such infrastructure, it is necessary to understand which protocol or scheme will be suited best for the application. A deeper understanding of protocols and the application requirements is necessary to properly select which protocol is most suitable for the application at hand.

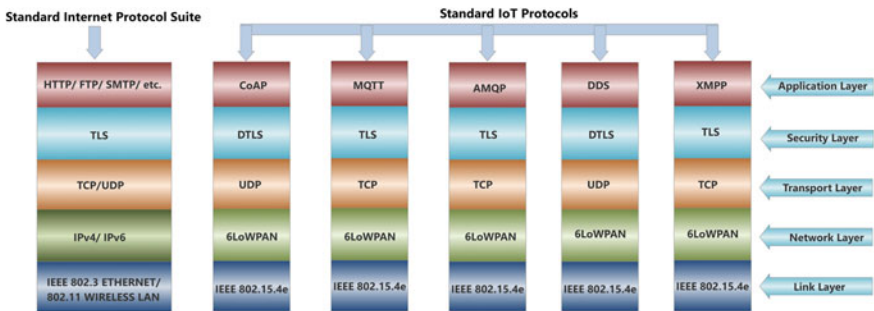


Fig. 5 Comparison of Internet and IoT protocol stacks

The network layer addresses and routes (multi-hop) data through the network. Networking protocols used to provide IP address to devices (things in IoT) to transport packets from one device to another. 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) is an open standard defined in RFC 6282 by the Internet Engineering Task Force (IETF) [10, 11]. 6LoWPAN is a very popular technology in IoT-Cloud for connecting low-power and IP-driven things (of IoT) to the cloud. It is a networking adaptation layer that allows IPv6 packets to be carried efficiently within small link layer frames. The main challenge in the IPv6 packet transmission in 6LoWPAN is the low capabilities of the 802.15.4 devices in terms of processing, bandwidth and memory [10]. 6LoWPAN is also adapted and used over different networking media like low-power RF, Bluetooth, low-power Wi-Fi, etc. IPv6 communication over IEEE 802.15.4 frames are defined by 6LoWPAN format [11].

In general, standalone IP networks are mainly based on IPv4, which uses 32-bit addresses. In IPv4, address space is limited to 2^{32} and that stimulated the development of IPv6. IPv6 solves the problem associated with IP address space by changing the addresses from 32 bits to 128 bits i.e., address space of 2^{128} . Minimum MTU (Maximum Transferable Unit) of IPv6 is 1,280 bytes that are ten times more than the one specified for 802.15.4 networks. As a result the IPv6 adoption as the network layer protocol does not fit with its MTU specifications [12]. An adaptation layer is always used when sending data over MAC and PHY layers. Moreover, the IPv6 header (40 bytes length) also creates a huge overhead. Therefore, to overcome the MTU requirements of IPv6 and header overhead, an adaptation layer is incorporated between network and data link layers in 6LoWPAN [13].

The transport layer generates communication sessions between application modules running on end devices. Multiple applications can run on a device and the transport layer allows them to have separate communication channel. TCP and UDP are the transport layer protocols on the Internet. Among them, UDP is a low overhead, connectionless protocol, and could be a better option for IoT. To make them secure, TLS (transport layer security) running on top of TCP and DTLS are paired with UDP.

Finally, the application layer is responsible for data formatting and also ensures that data is transported using application-optimal schemes. Different application layer protocols are briefly discussed here.

2.2 Constrained Application Protocol (CoAP)

CoAP is designed by The IETF Constrained RESTful Environments (CoRE) working group [14]. This protocol is mainly designed for machine-to-machine (M2M) communications. The CoAP is a generic web transfer protocol for use with resource constrained nodes [15]. The nodes often have 8-bit microcontrollers with small amounts of ROM and RAM. The CoAP defines a web transfer protocol based on REpresentational State Transfer (REST) on top of HTTP functionalities. REST

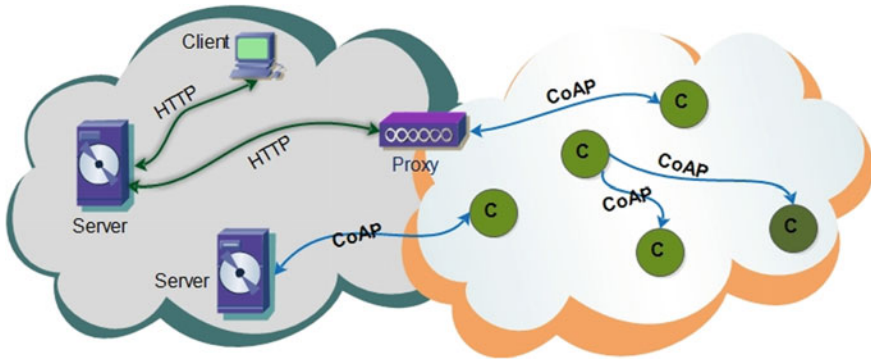


Fig. 6 CoAP architecture [15]

represents a simpler way to exchange data between clients and servers over HTTP. REST does not require XML for message exchanges. CoAP is designed to interoperate with HTTP and the RESTful web at large through simple proxies as shown in Fig. 6.

The CoAP messaging model is based on the exchange of messages over UDP and it also modifies some HTTP functionalities to handle resource constrained devices as well as lossy and noisy links. CoAP provides a request/response interaction model between application endpoints. Clients make requests to servers, servers send back responses. Clients may GET, PUT, POST and DELETE resources. Four types of messages are used in CoAP: confirmable, non-confirmable, reset and acknowledgement (ACK). Request and response semantics are carried in CoAP messages, which include either a method code or response code, respectively. A Token is used to match responses to requests independently from the underlying messages. In confirmable response mode, if a request is immediately available, then the response to a request is carried in the resulting ACK message. On the other hand, in non-confirmable response mode, the client sends data without waiting for an ACK message. To detect duplicates, message IDs are used. The server side responds with a RST message when messages are missed or communication issues occur. As CoAP is datagram based, it may be used on top of SMS and other packet based communications protocols.

2.3 Advanced Message Queuing Protocol (AMQP)

The AMQP is an open standard application layer protocol for message-oriented environments [16]. AMQP is comprised of several layers. It requires a reliable transport protocol like TCP to exchange messages. The most attractive feature of AMQP is that it ensures reliability by guaranteeing message delivery in terms of— (a) *at most once*: a message is sent once even if it is delivered or not; (b) *at least*

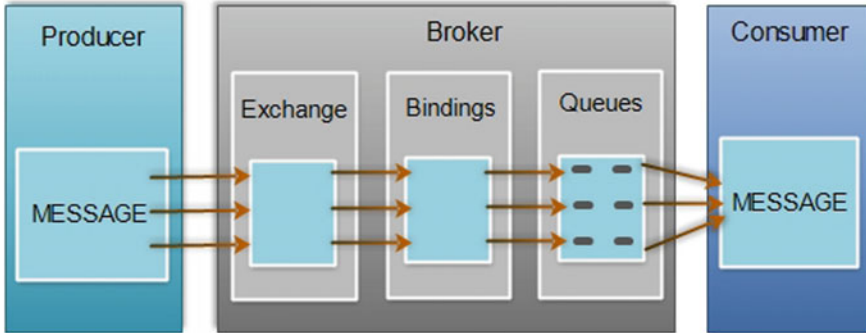


Fig. 7 AMQP architecture [16]

once: a message will be definitely delivered one time, possibly more; (c) *exactly once*: a message will be delivered only once. AMQP implementations are able to interoperate with each other by defining a wire-level protocol. An efficient peer-to-peer protocol is used in the AMQP transport specification for transporting messages between two devices over a network. In the messaging layer, an abstract message format is defined by the standard encoding scheme, which is used by AMQP compliant devices to send and receive messages. When a message travels through an AMQP network, its safe storage and delivery is transferred between the devices it encounters. A collection of devices in such network can form a container. Three types of roles are used in AMQP namely, producer, consumer, and queue. Producers and consumers are the part of an application that generate and process messages. Messages can be stored in message queues and then be sent to consumers as shown in Fig. 7.

Two types of messages are defined in AMQP: bare messages and annotated messages. Bare messages are supplied by the sender. The message as seen at the receiver is called annotated message. An annotated message consists of the bare message and sections for annotation at the head and the tail of the bare message. Two classes of annotations are used here, one that travels with the message till it is alive and the other is consumed by the next node.

2.4 Extensible Messaging and Presence Protocol (XMPP)

The eXtensible Messaging and Presence Protocol (XMPP) [17], is a standard specified by the IETF for instant messaging (IM) service that is used for multi-client chat, voice and video calling. It is an open XML protocol for messaging, presence, and request/response services. XMPP was first proposed and developed by Jabber open-source community to support an open, secure, spam free and decentralized messaging protocol. XMPP allows users to communicate with each other by sending instant messages on the Internet no matter which operating system they are

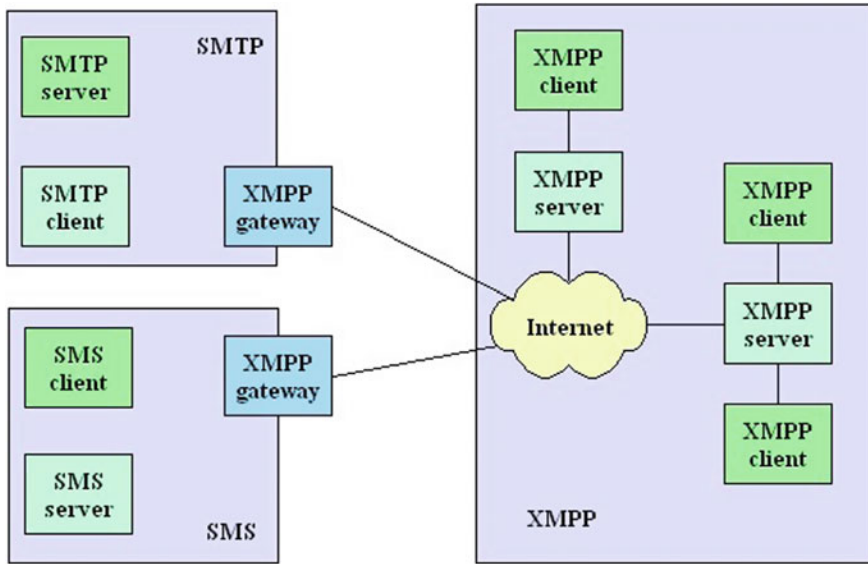


Fig. 8 XMPP architecture [18]

using. As an extensible protocol, XMPP is an ideal backbone protocol to provide universal connectivity among different endpoint protocols [18]. XMPP is usually implemented as client-server architecture and composed of three elements, XMPP client, XMPP server and gateways to foreign networks. Figure 8 shows an XMPP network with gateways to a Short Message Service (SMS) domain and an SMTP domain.

The XMPP server is responsible for connection management and message routing. The gateway serves a bridge between different networks and has to manipulate at least two different protocols. Any two devices in XMPP communicate with each other over TCP connections. XMPP has TLS/SSL security built on the core of the specification. However, it does not provide QoS options that make it impractical for M2 M communications. Only the inherited mechanisms of TCP ensure reliability. An XMPP client connects to an XMPP server using a stream of XML stanzas. XML stanza is the actual payload message in XML format that can be exchanged over the XML stream. Three key stanzas are: message, presence, and iq (info/query). Message stanzas generated by the client or server are used to “push” information to another entity. The source and destination addresses, types, and IDs of XMPP entities are first revealed by the message stanzas and a push method is called. A message stanza fills the subject and body fields with the message title and contents. Presence stanzas are used to express an entity’s current network status (e.g., availability) and then also notifies other entities of that status. IQ stanzas provide a structured request-response mechanism by pairing message senders and receivers.

2.5 Message Queue Telemetry Transport (MQTT)

MQTT was developed by Andy Stanford-Clark (IBM) and Arlen Nipper (Eurotech; now Cirrus Link) in 1999 and was standardized in 2013 at OASIS [19]. They were working on a project for the monitoring of an oil pipeline through the desert. A bandwidth and energy efficient protocol was required as the devices were connected via satellite link. MQTT is a M2M connectivity protocol used in IoT. The protocol uses an extremely lightweight publish/subscribe messaging transport architecture in contrast to HTTP with its request/response paradigm. Publish/Subscribe is event-driven and enables messages to be pushed to clients. It is useful for connections between devices with limited resources (like memory, storage, and network bandwidth) in remote locations. MQTT consists of three components, namely subscriber, publisher and broker. The broker in MQTT architecture is controlling all messages passing between the publishers and the subscribers as shown in Fig. 9.

Each device that wants to generate data can register as a subscriber for specific topics. Here, the publisher is a generator of data. If an event occurs, the publisher transmits that data to the broker first. To receive messages, a device needs to subscribe a certain topic. The broker first performs a matching and then delivers messages accordingly. In this scenario, nodes don't have to know each other, they only communicate over the topic. This architecture enables highly scalable solutions without dependencies between the data producers and the data consumers. MQTT is popular for mobile applications because of its small footprint, low power usage and less overhead minimized data packets, and efficient distribution of information to one or many receivers. MQTT ensures reliability by providing three

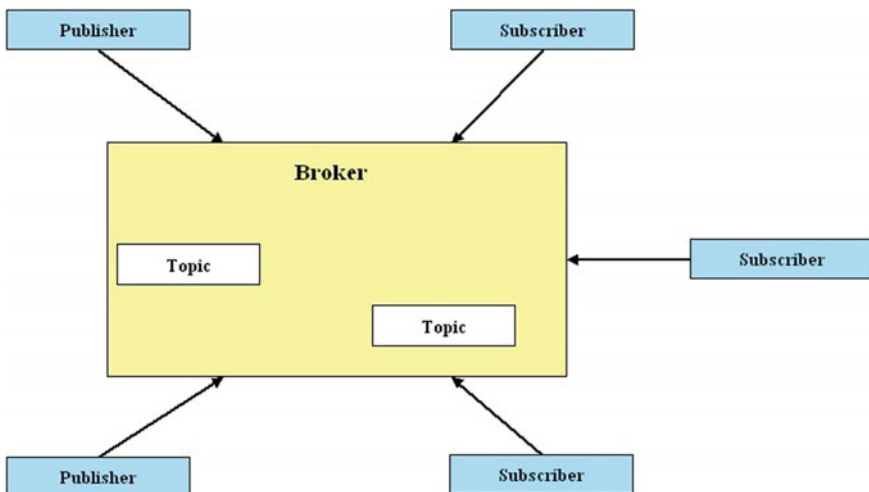


Fig. 9 MQTT architecture [19]

types of QoS levels—(a) *fire and forget*: a message is sent once and no acknowledgement is required; (b) *delivered at least once*: a message is sent at least once and an acknowledgement is required; (c) *delivered exactly once*: a four-way handshake mechanism is used to ensure the message is delivered exactly once.

2.6 Data Distribution Service (DDS)

The Data Distribution Service (DDS) was proposed by the Object Management Group (OMG) for communication between real-time systems in IoT [20]. It is a standard for data centric publish-subscribe model introduced in 2004. The main objective was to address the challenges faced by real time systems or mission-critical systems in a systems-of-systems environment. The two main components of DDS are DDS v1.2 API and the Data Distribution Service Interoperability (DDSI) wire protocol. Figure 10 shows the layered architecture of

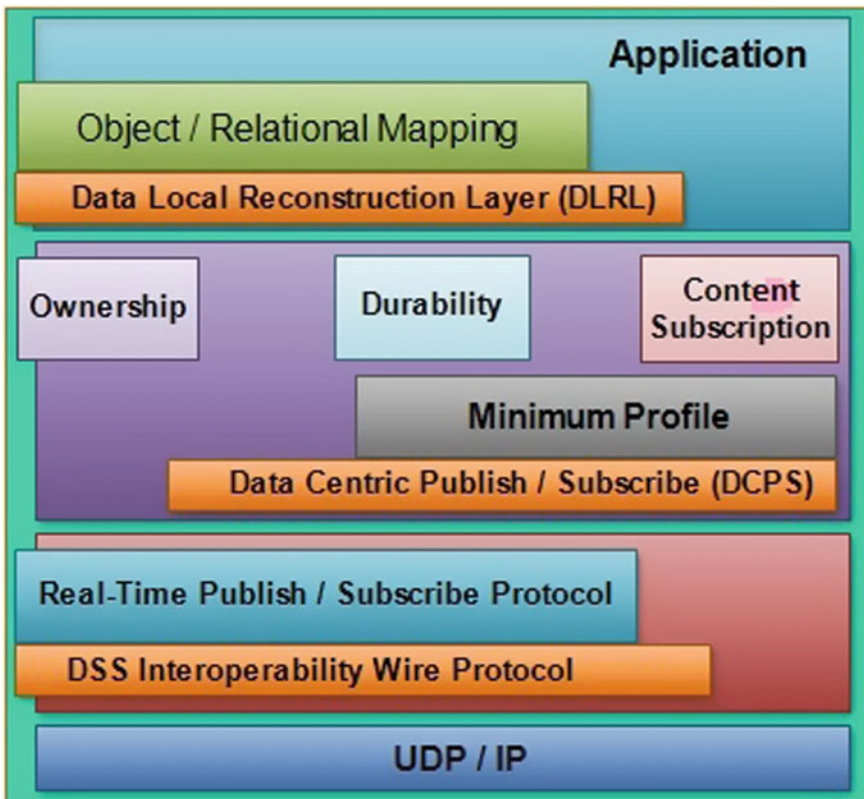


Fig. 10 DDS architecture [20]

DDS. DCPS is the standard API for data-centric, topic based, real-time publish/subscribe methodology. Standard API for creating object views out of collection of topics is DLRL (Data Local Reconstruction Layer). DDS API standard ensures source code portability across different type of devices, while the DDSI Standard ensures on the wire interoperability across different types DDS implementations. The key abstraction at the foundation of DDS is a fully distributed Global Data Space (GDS). To avoid single points of failure or single points of contention, the DDS specification requires a fully distributed implementation of the GDS. Publishers and subscribers are dynamically discovered and can join or leave the GDS at any point of time. The dynamic discovery of publisher and subscribers is performed by the GDS and does not rely on any kind of centralized registry services.

3 Cloud Environment

Cloud environment provides compute and storage infrastructural support for it based applications. National Institute of Standard and Technologies (NIST) defines the cloud as [21]: *Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.* This new computing model is enabled due to the availability of virtually unlimited storage and processing capabilities. Significant innovations in virtualization and distributed computing, as well as improved access to high-speed Internet and a weak economy have accelerated interest in cloud computing. In cloud, these virtualized resources can be leased in an on-demand fashion or as general utilities. Also, it can host services to be delivered over the Internet. A cloud service has three distinct characteristics that differentiate it from traditional hosting.

- *On demand:* Services in the cloud are available on demand.
- *Elastic:* Quality and quantity of resources and services in the cloud can be provided according to the demand or requirement.
- *Low (No) headache:* Services and resources are mainly managed by the cloud service provider and the service consumer only needs a computing device and Internet to access those resources.

In IoT, large number of physical objects equipped with sensors like smart phones, smart home appliances, etc. are connected to the Internet and generate huge amounts of data what is called *big data*. The real challenge is to analyze these data to build the knowledge base and ultimately the ability to respond to the world with greater intelligence. Hadoop is an open source cloud computing environment created and maintained by the Apache project. In the following subsection, we provide an overview of some of the important modules of the Hadoop framework.

3.1 Enabling Technologies

Hadoop is one of the most popular open source frameworks for distributed programming on commodity hardware. Figure 11 shows the Hadoop ecosystem. This framework allows the distributed processing of large data sets stored across clusters of computers. Hadoop common contains libraries and utilities needed by other Hadoop modules. The Hadoop framework consists of two main components—Hadoop Distributed File System (HDFS) [22, 23] and MapReduce programming framework [24, 25]. Basically HDFS is an open source variant of the Google File System (GFS) and Hadoop MapReduce is the open source variant of Google MapReduce. To process data in Hadoop framework, we first need to move the data to HDFS. HDFS is designed to reliably store very large files across nodes in a large cluster. It stores each file as a sequence of blocks; all blocks in a file except the last block are the same size. The blocks of a file are replicated for fault tolerance. The block size and replication factor are configurable per file. An application can specify the number of replicas of a file. HDFS follows the master/slave architecture as shown in Fig. 12. An HDFS cluster consists of a single namenode and one or multiple datanode(s). Namenode acts as the master of a Hadoop cluster and maintains the namespace of the HDFS. It also provides access to the files, by users. Data in HDFS does not flow through namenode, it only keeps track of the data in the Hadoop cluster. The JobTracker is the service within Hadoop that assigns MapReduce tasks to specific nodes in the cluster, ideally the nodes that have the data. A TaskTracker is a node in the cluster that accepts tasks (like map, reduce and shuffle) from a JobTracker.

ZooKeeper is used to coordinate the cluster in hadoop framework [26]. The high availability feature of the framework is due to the coordination functionality of the Zookeeper. The namenode was a single point of failure (SPOF) in a cluster. Now it has the option of running two redundant namenodes in the same cluster in an

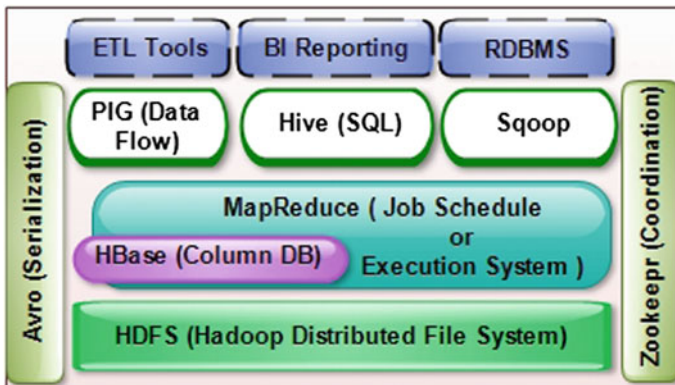


Fig. 11 Hadoop ecosystem (schematic diagram by Cloudera)

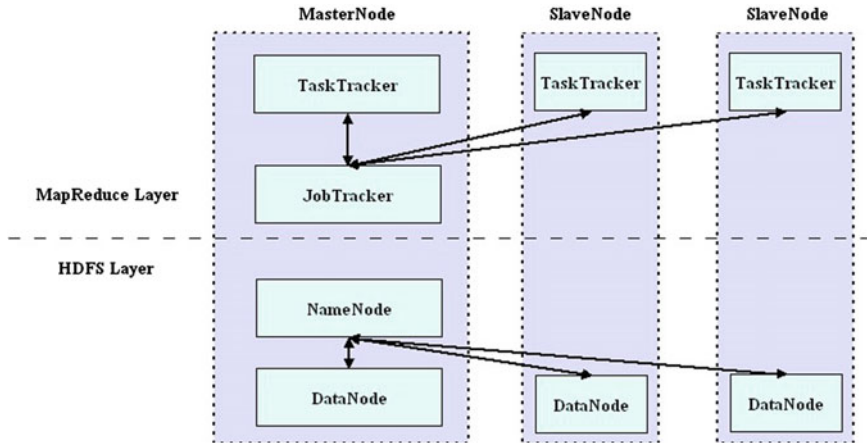


Fig. 12 Hadoop high level architecture [22]

active/passive configuration. ZooKeeper was developed at Yahoo! Research. Several Hadoop projects are already using ZooKeeper to coordinate the cluster and provide highly-available distributed services.

Apache Avro [27] is used for Remote Procedure Calls (RPC) in the system. It supports self describing schema for the data, where data is described by a schema and also stored in the same file as the data it describes.

MapReduce is the programming part of Hadoop framework. This programming paradigm consists of two phases. One is Map phase and the other one is the Reduce phase. A user generally needs to implement these two phases by defining mapper and a reducer method/procedure. The input to a MapReduce job is a file. The input file is split into 64 MB blocks (64 MB is the default block size. The block size can change according to the configuration setup). The last block size can be less than 64 MB. Each of these blocks is fed to a mapper method via a record reader. Record reader creates a key-value pair upon each iteration with the input block and passes that key-value pair to the mapper method as input. The mapper method in turn performs the user specified task and produces results as key-value pair. Reducer method called exactly once, for each key appeared in the shuffled intermediate data and performs user specified task for all the values of a key at a time. Finally, it produces output as <key, value> pair and writes this to an output file. Figure 13 depicts the working principle of MapReduce.

Beside HDFS and MapReduce, the other related projects in the Apache Hadoop platform are: Pig, Hive, HBase, Sqoop, Spark and others. Pig [28] provides an engine for executing data flows in parallel on Hadoop. It uses MapReduce to execute all of its data processing. Hive [29] is the data warehouse infrastructure developed by Facebook. It is used for data summarization, query, and analysis. HiveQL is a SQL-like language. HBase [30] is a Hadoop database inspired from Google BigTable and non-relational distributed database. It is used as a storage

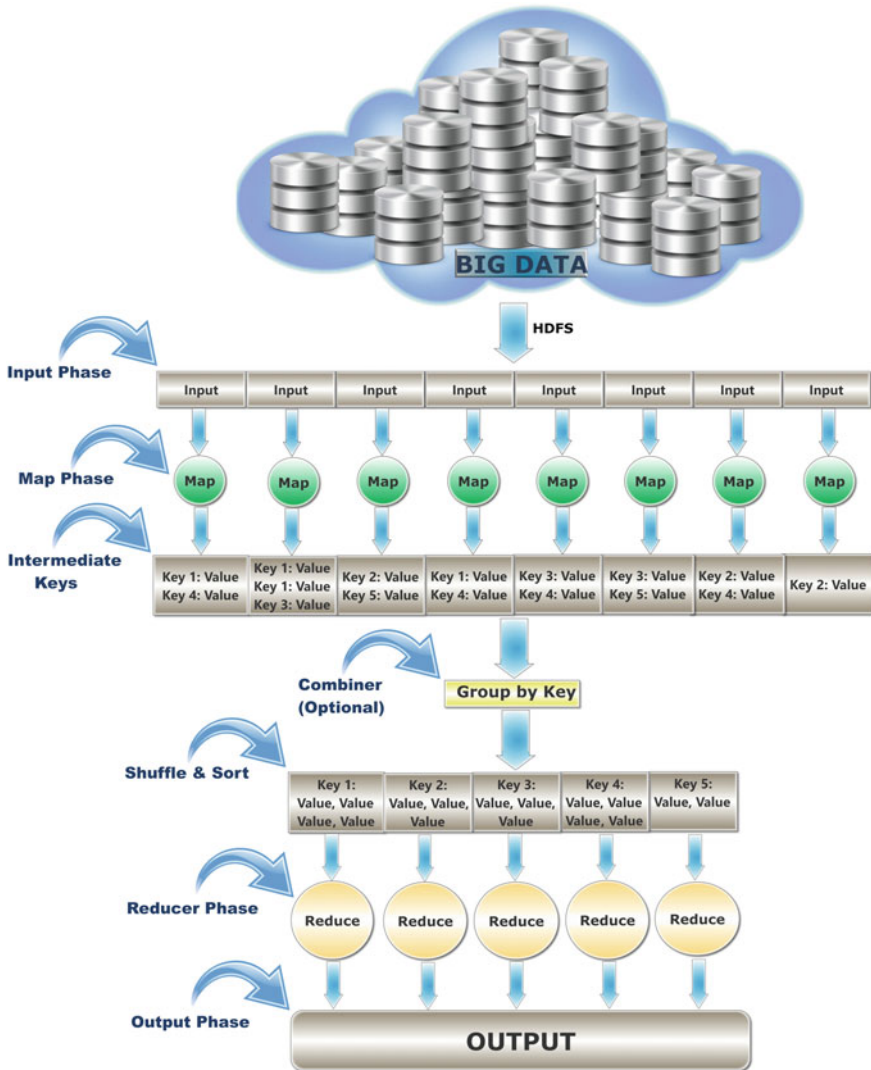


Fig. 13 MapReduce architecture [25]

system for MapReduce jobs outputs. Most useful to store column-oriented, very large tables for random, real-time read/write operations. Sqoop [31] is a system for bulk data transfer between HDFS and structured data-stores as RDBMS. Spark is a framework for writing fast, distributed programs. Spark [32] solves similar problems as Hadoop MapReduce does, but with a fast in-memory approach and a clean functional style API [33].

4 Applications

There are several application areas which will be impacted by the integration of IoT-Cloud. There is a huge crossover in applications and the use of data between different domains as the Internet enables sharing of data between different service providers in a seamless manner. In a smart city environment, various devices related to our daily life are interconnected to provide required services like homes and buildings, health, transportation, industry etc. Figure 14 shows a schematic diagram of the involvement of things and people in some typical IoT-Cloud applications, which are briefly discussed here.

- Smartest human: Today’s technology will aid us in every task, from learning to work to entertainment to health to distant worlds. With wearable sensors, human



Fig. 14 IoT-things, people and application areas

activity recognition is not a very complicated task nowadays. Smart gadgets can easily impact our day to day lives by capturing accurate and opportune information of our activities. Such information is also helpful to support decision making in different scenarios. For example, precise information on human beings' activities along with their locations and physical conditions is highly beneficial for their day to day health monitoring. Moreover, constant connectivity between people comes through social network sites, which also sometimes influences their lives for the betterment. For example, crowd-sensing is used in applications like BikeNet [34] where individuals measure location and bike route quality based on parameters like CO₂ content on route, the bumpiness of ride etc. Data collected is analyzed to obtain the "most" bikeable routes. Common Sense [35] provides a prototype deployment for pollution monitoring based on participatory sensing. In this work, specialized handheld air quality sensing devices that communicate with mobile phones (using Bluetooth) are utilized to measure various air pollutants (e.g., CO₂; NO_x). One can utilize microphones on mobile phones to monitor noise levels in an area as well. The devices when used by crowd can help monitor pollution levels across an entire region.

- **Smart Medical:** Today's healthcare applications could be benefited hugely with the help of IoT, from clinical care to remote health monitoring to early intervention or prevention. Physiological status of patients can be constantly monitored remotely using IoT. In Masimo Radical-7 [36], patient's health status is monitored remotely and then the system reports that to a caregiver. Another interesting example is discussed in [37], where IBM utilized RFID technology to track hand washing after checking each patient at one of OhioHealth's hospitals. In this type of application, sensors attached to body forming body area network (BAN) which collects comprehensive physiological information and uses gateways to forward that data to cloud. Cloud environment can store the information and then send the analyzed data to caregivers for further analysis and review.
- **Smart Home:** In general, smart home is designed to deliver a number of services through a range of networked devices that can be accessed and controlled within and outside the home [38]. The devices in the home are all interconnected and have the potential to share information with one another and they have become part of building automation system (BAS). High-speed Internet access is not essential for all the devices deployed in the home to form the IoT. Though the full functionality of a smart home depends on the availability of a permanently accessible Internet connection where multiple devices might be connected in the same instance to a central hub. IoT based BAS allows to control and manage different home devices using sensors and actuators such as lighting and shading, security and safety, entertainment, etc.
- **Smart Transport:** Many works are done on smart transportation as in [39]. In CityPulse [40], 101 smart city application scenarios have been identified, including facilitating transportation such as a real time travel planner or a service predicting public parking space availability. In [41], Singapore's bike sharing

system is proposed. Here the idea was to replace short train routes (maximum three stops in the popular train network of the city) by bicycles that may be taken for a rent and parked near the destination [42]. The authors propose that if an individual starts a ride in the system s/he is asked to give his/her destination. Built-in GPS sensors can be utilized for tracking the bicycles and predicting its availability. Tracking public transport vehicles can also be utilized for predicting arrival times of buses at a bus stop and also availability of seats. This is particularly important for harsh weather conditions. This kind of application mostly relies on accelerometer readings of the smart devices and it uses a progressive localization technique comparing Wi-Fi SSIDs sensed at different stopping places as in [43]. Applications like Tranquilien [44], Moovit [45] and Tiramisu [46] are also built on a similar idea of route planning by predicting the conditions of public vehicles, for instance crowdedness, arrival times, cleanliness, availability of air conditioning etc. In Tranquilien [44], citizens can predict well in advance the comfort of trains in France. It uses optimization algorithms to predict if a person (in a compartment) should be able to find a seat; some chance of obtaining a seat; and standing room only up to three days in advance. Historical data of passengers are fed into the system. There is also an option for crowdsourcing where the passengers can share their experience so that corrections may be done for any wrong prediction. On the contrary applications like Moovit [45] uses data to plan future infrastructure and service provision based on demand. Alternatively, opportunistic sensing may be utilized to identify the crowded routes of the city, predict pollution levels and design better public transport service as in projects like Istanbul in Motion initiated by Vodafone and IBM [47].

- **Smart Industry:** With the advancements in IoT and modern industrial technology, an organization can make intelligent operating decisions. Using IoT, it is possible and viable to establish connectivity among the main assets in industrial facilities. Moreover, industrial automation focused on minimizing human involvement by computerizing robotic devices to complete manufacturing tasks. Four main pillars of smart industry are: sensing, processing, communication and transportation. Communications and process technology advancements continue to enable more data collection with increasing number of sensors monitoring. Finally, that big data is stored in cloud infrastructure to increase productivity by analyzing production data, timing and causes of production issues. Thus, IoT-Cloud [48] is utilized in industrial automation to control and monitor M2M and M2H interactions. For example, if a particular device encounters a sudden failure, the system can send a maintenance request immediately to the maintenance department to handle the issue. Another popular example is smart grids [49, 50]. IoT-Cloud based smart grids can improve and enhance the energy consumption of buildings. On the other hand, it reduces the potential failures and thus increases efficiency and improves quality of services.

- **Smart Surveillance:** Nowadays a smart surveillance system becomes a critical component in security infrastructures of different high risk areas. A smart surveillance in an airport environment is presented in [51]. Today's video surveillance systems can completely take over existing systems with the increasing availability of the IoT enabled video infrastructure and better video analysis technologies with the help of cloud.

5 Research Challenges in IoT-Cloud

The fields of IoT and cloud have gone through an independent evolution. IoT based applications can get benefit from the virtually unlimited capabilities and resources of cloud to compensate the technological constraints of IoT devices. Similarly, cloud can also extend its scope to deal with the real world things in a more distributed and dynamic manner, and for delivering new services in a large number of real life scenarios. In this chapter, we have discussed the complementary characteristics of cloud and IoT. Due to several mutual advantages, IoT-Cloud can offer an effective solution to the real world. However, there are some issues and challenges that need to be tackled in IoT-Cloud. Some of the prominent challenges in this field are discussed below.

- **Networking and Communication:** IoT-Cloud is centered on data fusion as the fundamental network purpose. It involves M2M communications among many heterogeneous devices with different protocols. This is a direct consequence of the integration of an increasing number of sensing devices with the Internet and the increasing need to elevate information abstractions to bridge the human-machine gap. It is very challenging to manage these heterogeneous things in a uniform fashion without compromising the performance. Data fusion as the fundamental network's purpose will likely require a different protocol stack. A fundamental challenge is therefore to redefine network architecture in a way that optimizes it for distributed information fusion and retrieval.
- **Decentralized nature:** There is no central control, as the sensor nodes or devices in IoT are mobile and typically scattered over a vast area. There is no single unit that can monitor and coordinate the behavior of all sensors or devices. As a result, nodes need to coordinate their transmissions in a decentralized i.e., distributed manner.
- **Distributed computing:** New models and paradigms are needed for distributed IoT-Cloud environment. A service framework must be developed for sensor networks to make SaaS platforms a viable option for target applications. Both PaaS and IaaS are also viable architectures.
- **Data management:** In general, the data in IoT are un-structured or semi-structured. Moreover, these massive amounts of data are coming from distributed IoT sources. IoT-Cloud has to provide real-time data processing and service provisioning techniques considering such Big Data. Data mining and

machine learning techniques will play an increasingly important role in IoT-Cloud to identify data patterns, learn the context, detect complex distributed events of interest, and generally act without human assistance. This has important implications on the design of network protocols and programming abstractions. Reusable tools will be needed to deal with data management functions in a lossy, possibly mobile, poorly-structured environments.

- Knowledge sharing: Due to the mobility and limited transmission and sensing range, nodes or devices have only local information and lack any global. Moreover, sensors cannot directly observe the actions of others, but only the effect of their own actions. Again, communicating such local information comes at a certain cost.

6 Conclusion

Today we are in the era of IoT and with the advancement of technologies eventually will be part of IoE. The vision of the IoT-Cloud is to utilize the increased sophistication in sensing and actuation, communication, and exploring knowledge from big data to improve our lifestyles. The main intention of this chapter is to draw up a research agenda for an exploitation of IoT and cloud computing in different application fields. We firstly presents an overview of the IoT and cloud environment and then discuss the background of why integration is so essential. Different protocols and paradigms important in the integration of IoT-Cloud are also discussed. Next, the most relevant smart applications of IoT-Cloud have been presented. A number of research issues associated with IoT-Cloud has been identified and analyzed. IoT-Cloud provides functionalities to manage resources in the dynamic, distributed, heterogeneous environment. These are also defining characteristics of the IoT-Cloud environment.

References

1. Atzori L., Iera A., and Morabito G., The internet of things: A survey, *Computer Network*, vol. 54, no. 15, pp. 2787–2805, 2010.
2. Zanella A., Bui N., Castellani A., Vangelista L., Zorzi M., Internet of Things for smart cities, *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–33, February 2014.
3. Bojanova I., Hurlburt G., Voas J., Imagineering an Internet of Anything, *IEEE Computer Journal*, pp. 72–77, June 2014.
4. Balfour R.E., Building the “Internet of Everything” (IoE) for first responders, *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pp. 1–6, 2015.
5. Ganti R.K., Fan Y., and Hui L., Mobile Crowdsensing: Current State and Future Challenges, In *IEEE Communications Magazine*, pp. 32–39, 2011.

6. Geoffrey C. F., Kamburugamuve S., Hartman R.D., Architecture and Measured Characteristics of a Cloud Based Internet of Things API, International Conference on Collaboration Technologies and Systems (CTS), pp. 6–12, 2012.
7. Antonic A., Roankovic K., Marjanovic M., Pripucic K., Zarko I.P., A Mobile Crowdsensing Ecosystem Enabled by a Cloud-Based Publish/Subscribe Middleware, International Conference on Future Internet of Things and Cloud (FiCloud), pp. 107–114, 2014.
8. Kryftis Y., Mavromoustakis C.X., Mastorakis G., Pallis E., Batalla J. M., Rodrigues J. P.C., Dobre C., Kormentzas G., Resource Usage Prediction Algorithms for Optimal Selection of Multimedia Content Delivery Methods. IEEE International Conference on Communications ICC. London, UK, 2015.
9. Batalla J. M. and Krawiec P., Conception of ID layer performance at the network level for Internet of Things. Springer Journal Personal and Ubiquitous Computing, vol.18, Issue 2, pp. 465–480, 2014.
10. Montenegro G., Kushalnagar N., Culler d., Transmission of IPv6 Packets over IEEE 802.15.4 Networks, IETF RFC 4944, 2007. <http://tools.ietf.org/html/rfc4944> (Accessed on May 2, 2016).
11. Kim E., Kaspar D., Gomez C., Bormann, C.; Problem Statement and Requirements for 6LoWPAN Routing, draft-ietf-6lowpan-routingrequirements-10, November 2011. <http://tools.ietf.org/html/draft-ietf-6lowpan-routing-requirements-10> (Accessed on May 2, 2016).
12. Bhunia S. S., Sikder D., Roy S., Mukherjee N., A Comparative Study on Routing schemes of IP based Wireless Sensor Network, in the proceedings of the IEEE 9th International Conference on Wireless and Optical Communications Networks (WOCN), IEEE, Indore, India, pp. 1–5, 20–22 September 2012.
13. Mulligan G., The 6LoWPAN Architecture, ACM EmNets2007, Cork Ireland, 2007.
14. Shelby Z., Hartke K., Bormann C., Frank B., Constrained Application Protocol (CoAP), draft-ietf-core-coap-18, Internet Engineering Task Force (IETF), Fremont, CA, USA, 2013.
15. Bormann C., Castellani A. P., Shelby Z., CoAP: An application protocol for billions of tiny Internet nodes, IEEE Internet Computing, Volume 16, Issue 2, pp. 62–67, 2012.
16. OASIS Advanced Message Queuing Protocol (AMQP) Version 1.0, Advanced Open Standard for the Information Society (OASIS), Burlington, MA, USA, 2012.
17. Saint-Andre P., Extensible messaging and presence protocol (XMPP): Core, Internet Engineering Task Force (IETF), Fremont, CA, USA, 2011.
18. Jones M. T., Meet the Extensible Messaging and Presence Protocol (XMPP), IBM Developer Works, Markham, ON, Canada, 2009.
19. Locke D., MQ telemetry transport (MQTT) v3. 1 protocol specification, IBM Developer Works, Markham, ON, Canada, 2010.
20. Corsaro A., Schmidt D. C., The Data Distribution Service – The Communication Middleware Fabric for Scalable and Extensible Systems-of-Systems, System of Systems, Dr. Adrian V. Gheorghe (Ed.), ISBN: 978-953-51-0101-7, InTech, 2012.
21. P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” US Nat’l Inst. of Science and Technology, 2011; <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf> (Accessed on May 2, 2016).
22. HDFS Architecture Guide https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html (Accessed on May 2, 2016).
23. Shvachko K., Hairong Kuang, Radia S., Chansler R., The Hadoop Distributed File System, In IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), pp. 1–10, 2010.
24. MapReduce Tutorial https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html (Accessed on May 2, 2016).
25. Thusoo A., Sen Sarma J., Jain N., Shao Z., Chakka P., Anthony S., Liu H., Wyckoff P., Murthy R., Hive: a warehousing solution over a map-reduce framework, In Proceedings of VLDB Endow. 2, pp. 1626–1629, August 2009.
26. Zookeeper Tutorial <https://zookeeper.apache.org/> (Accessed on May 2, 2016).
27. Avro Tutorial <https://avro.apache.org/> (Accessed on May 2, 2016).
28. Pig Tutorial <https://pig.apache.org/> (Accessed on May 2, 2016).

29. Hive Tutorial <https://hive.apache.org/> (Accessed on May 2, 2016).
30. HBase Tutorial <https://hbase.apache.org/> (Accessed on May 2, 2016).
31. Sqoop Tutorial <http://sqoop.apache.org/> (Accessed on May 2, 2016).
32. Spark Tutorial <http://spark.apache.org/> (Accessed on May 2, 2016).
33. Badcock G, Googles' Big Query vs Hadoop: Complimentors or Competitors? February 2013. <https://gavinbadcock.wordpress.com/2013/02/06/googles-bigquery-vs-hadoop-complimentors-or-competitors/> (Accessed on May 2, 2016).
34. Eisenman S.B., Miluzzo E., Lane N.D., Peterson R.A., Ahn G-S., Campbell A.T., The BikeNet mobile sensing system for cyclist experience mapping. In: Proc of the 5th international conference on Embedded networked sensor systems (SenSys'07), pp. 87–101, 2007.
35. Dutta P., Paul M. A., Kumar N., Mainwaring A., Myers C., Willett W., and Woodruff A., Demo abstract: Common sense: Participatory urban sensing using a network of handheld air quality monitors, in Proc. of ACM SenSys, pp. 349–350, 2009.
36. Masimo Corporation, Radical monitor, Radical-7 breakthrough measurements, Irvine, CA, USA, Data Sheet Radical-7, 2013.
37. Nay C., Sensors remind doctors to wash up, IBM Research, Armonk, NY, USA, 2013.
38. Liu Y., Study on Smart Home System Based on Internet of Things Technology, Book Chapter in Informatics and Management Science, Lecture Notes in Electrical Engineering, Series Volume 207, pp. 73–81, 2013.
39. Farkas K., Feh'er G., Bencz'ur A., Sidl'ó C., Crowdsensing Based Public Transport Information Service in Smart Cities, In IEEE Communications Magazine, pp. 158–165, August 2015.
40. Presser M., Vestergaard L., Ganea S., Smart City Use Cases and Requirements, CityPulse Project Deliverable D2.1, May 2014.
41. Shu J., Chou M., Liu Q., Teo C. P., Wang I. L., Bicycle-sharing system: deployment, utilization and the value of re-distribution. Technical report. National University of Singapore-NUS Business School, Singapore, 2010.
42. Fricker C., Gast N., Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. EURO Journal on Transportation and Logistics, pp. 1–31, 2014.
43. Petkovics Á., Farkas K., Efficient event detection in public transport tracking, In International Conference on Telecommunications and Multimedia (TEMU), pp. 74–79, July 2014.
44. Tranquilien, mobile application, <http://www.tranquilien.com> (Accessed on May 2, 2016).
45. Moovit, mobile application, online: <http://www.moovitapp.com/> (Accessed on May 2, 2016).
46. Tiramisu, the real-time bus tracker, <http://www.tiramisutransit.com/> (Accessed on May 2, 2016).
47. Guide to Smart Cities, The opportunity for mobile operators, GSMA report, February 2013.
48. Wang C., Bi Z., Xu L. D., IoT and cloud computing in automation of assembly modeling systems, IEEE Transaction for Industrial Information, Volume 10, Issue 2, pp. 1426–1434, May 2014.
49. Yan Y., Qian Y., Sharif H., Tipper D., A survey on smart grid communication infrastructures: Motivations, requirements and challenges, IEEE Communications Surveys and Tutorials, Volume 15, Issue 1, pp. 5–20, 2013.
50. Komninos N., Philippou E., Pitsillides A., Survey in smart grid and smart home security: Issues, challenges and countermeasures, IEEE Communications Surveys and Tutorials, Volume 16, Issue 4, pp. 1933–1954, 2014.
51. Shu C. F., Hampapur A., Max L., Brown L., Connell J., Senior A., YingLi T., IBM smart surveillance system (S3): a open and extensible framework for event based surveillance, IEEE Conference on Advanced Video and Signal Based Surveillance, pp. 318–323, 15–16 September 2005.

Multimodal Low-Invasive System for Sleep Quality Monitoring and Improvement

**Fábio Manoel Franca Lobato, Damares Crystina Oliveira de Resende,
Roberto Pereira do Nascimento, André Luis Carvalho Siqueira,
Antonio Fernando Lavareda Jacob, Jr. and Ádamo Lima de Santana**

Abstract The attention on sleep disorders has grown in recent years. Mainly because of the changes imposed by the information and communication Era. These changes have impacted in a hazardous way on our sleep quality. The current mainstream sleep disorder detection and assessment method, the laboratory polysomnography, is very expensive and inconvenient for patients who are extracted from their own sleep-environment. Aiming to avoid the high costs and to perform an assessment in loco, we present in this chapter a multimodal low-invasive system for sleep quality monitoring and its improvement by Internet of Things paradigm. A stand-alone device was designed to provide robustness, scalability and usability to a completely built-in sleep assessment system. The main goal of this in-home device is to give more accurate information to physicians and technical staff, assisting in the screening process, reducing costs and helping to improve the wellbeing of people with sleep disorders.

F.M.F. Lobato (✉) · R.P. do Nascimento
Universidade Federal do Oeste do Pará, Santarém, PA, Brazil
e-mail: fabio.lobato@ufopa.edu.br

R.P. do Nascimento
e-mail: robertotpd@gmail.com

D.C.O. de Resende · A.L.C. Siqueira · A.F.L. Jacob, Jr. · Á.L. de Santana
Universidade Federal do Pará, Belém, PA, Brazil
e-mail: d.oliveiraresende@gmail.com

A.L.C. Siqueira
e-mail: alcas.andre@gmail.com

A.F.L. Jacob, Jr.
e-mail: jacobjr@engcomp.uema.br

Á.L. de Santana
e-mail: adamo@ufpa.br

A.F.L. Jacob, Jr.
Universidade do Estado do Maranhão, São Luís, MA, Brazil

1 Introduction

Sleep plays an important role in wellbeing. Humans, in fact, spend roughly one-third of their lives asleep [1]. In this sense, sleep studies aim to develop and apply standardized tests to record body activity during sleep, in order to identify or refute the existence of a sleep disorder in patients that report these kinds of problems. A sleep disorder, also known as somniphobia, is a medical condition in which sleep is abnormal. Formerly, it was considered a problem that affected mainly the elderly. Nowadays it affects anyone from children to retired people.

There are more than 50 sleep disorders listed in the International Classification of Sleep Disorders (ICSD), third edition [2]. The prevalence of insomnia, for instance, depends on the criteria used to define itself and the people studied. Although, a consensus on population-based studies shows that approximately 30 % of adults report one or more insomnia symptoms [3]. Despite this fact, approximately 7 % of the general population meet the criteria for episodic insomnia, which is defined as a difficulty that lasts at least 1 month to initiate and/or maintain sleep, which makes it hard to diagnose [4]. Other sleep disorder symptoms are: short sleep duration, nonrestorative sleep sensation and daytime sleepiness.

Information and communication technologies are imposing lifestyle changes, especially the introduction of laptops, tablets and smartphones in the bedroom environment, which has had a detrimental impact on this sleep-friendly ambient [5]. As a consequence, according to the National US surveys, there has been a reduction of 1.5–2 hours in self-reported sleep duration in the past 50 years.

Beyond to the aforementioned medical conditions associated with sleep disturbances, other negative consequences should be highlighted. In children and teens, this drop in sleep quality has caused mood disturbances like behavioral problems of attention deficit, hyperactivity disorder and memorization issues, which all have a huge impact in learning performance [6]. In adolescents attending college, which are suffering increasing exposure to the information bombardment, added to the consequences that affect children and teens, there is also a raise in risk-taking behaviour, depression and impaired social relationships [7]. In adults, we observe that the symptoms above also reduce productivity and increase absenteeism and occupational accidents [5].

Based on these issues, sleep disorders can be seen as a public health problem, increasing costs of education and health services and reducing the work force productivity. Thus, the diagnosis of sleep disturbances becomes a fundamental step for the society's wellbeing improvement. The current mainstream sleep disorder detection and assessment method, the laboratory polysomnography, is very expensive and inconvenient for patients as they are extracted from their normal sleep-environment. Moreover, this unusual environment negatively impacts sleep physiology, and the typical single-night paradigm does not measure variability across multiple nights.

In this context, the Internet of Things (IoT) through pervasive technologies such as wearable devices, environmental sensors, cameras, mobile phones, self-applied

cognitive tests and even social network data, can support the sleep assessment. For instance, IoT allows a global-scale quantification of sleep schedules using smartphone data [8]. So, it could be said that these technologies, combined with sleep monitoring systems, can provide enough information about people's sleep patterns, guiding the diagnosis and treatment processes in a more accurate way.

The IoT is not restricted to sleep monitoring only [9], it can be used to improve sleep quality as well. Through intelligent control systems it is possible to develop, among other things, (1) adaptive and natural alarm clocks—using natural lights by means of controlled blinds and flexible timetables; (2) more comfortable sleep-environments—controlling some parameters such as temperature, humidity and lights, making the bedroom even more sleep-friendly; and (3) give some tips along the day with health behaviours that can improve the sleep quality, such as: avoid caffeine beverages and high calorie foods after six o'clock, reminders to practice physical exercises etc.

In this sense, it is described in this chapter a low-invasive sleep monitoring system using multimodal technologies, including a non-invasive sleep-environment monitoring device [10]. The objective of this project is to improve the wellbeing of people with sleep disorders through some smart home facilities. Such system encompasses, but is not restricted to:

- An in-home device for sleep-environment monitoring;
- Investigation of multimodal technologies to assess sleep quality (e.g. EEG headsets; cardiac, oxygen saturation and sugar level monitors; cognitive tests presented in mobile games; social network data etc.);
- Development of strategies to improve sleep quality using smart home facilities.

The exploitation of this study is manifold. For instance, it is known the high impact of sleep deprivation in human emotions; therefore, the data acquired by the proposed system can be used not only to perform emotion recognition, but also to help to predict it. Regarding to patients with chronic diseases that negatively impact in sleep quality, this system can improve others e/m health systems, which do not cover sleep assessment, for example. It is important to stress that the main contribution of this in-home device is to give more accurate information to physicians and technical staff, assisting in the screening process, reducing costs and helping to improve the wellbeing of people with sleep disorders.

2 Internet of Things in e-Health Context

IoT is a novel paradigm of wireless telecommunications that connects “everything” to the Internet. The term “Internet of Things” was firstly proposed by Kevin Ashton in the article “That Internet of Things Thing” to the RFID Journal that describes a system where physical objects can be connected to the Internet through several types of information sensor devices. In other words, it is a global and dynamic network infrastructure with capacity to self-configure based on standard and interoperable

communication protocols, where virtual and physical things have identity (electronic product code—EPC), physical attributes and virtual personality, and also use intelligent interfaces and are perfectly integrated in the information network [11, 12].

The IoT allows a large number of physical devices connections to each other aiming to interact and perform wireless data communication, using the Internet as a global communication environment [13]. Under this vision and by making intelligent use of the network infrastructure support, things will be capable to provide autonomous transport, implement automatic process and, as a consequence, optimize logistic; They must be able to collect the energy they need, to configure themselves when exposed to a new environment and to show an “intelligent/cognitive” behaviour when dealing with other things so that they would be able to continually deal with unexpected circumstances. Finally, these things will manage their own disassembly and recycling, helping to preserve the environment at the end of their life cycle [14].

By combining several technological elements, the International Telecommunications Union (ITU) described a new dimension to the telecommunications ambient: from anytime, anyplace and anyone connectivity, we will be able to connect anything [14–16]. Figure 1 shows this new dimension.

There are two distinct modes of communication in the IoT [17]:

- Thing-to-person (and person-to-thing—communications covers a series of technologies and applications where people interact with things and vice versa, including the remote access to objects by humans and objects that continuously report their status, location and sensor data;

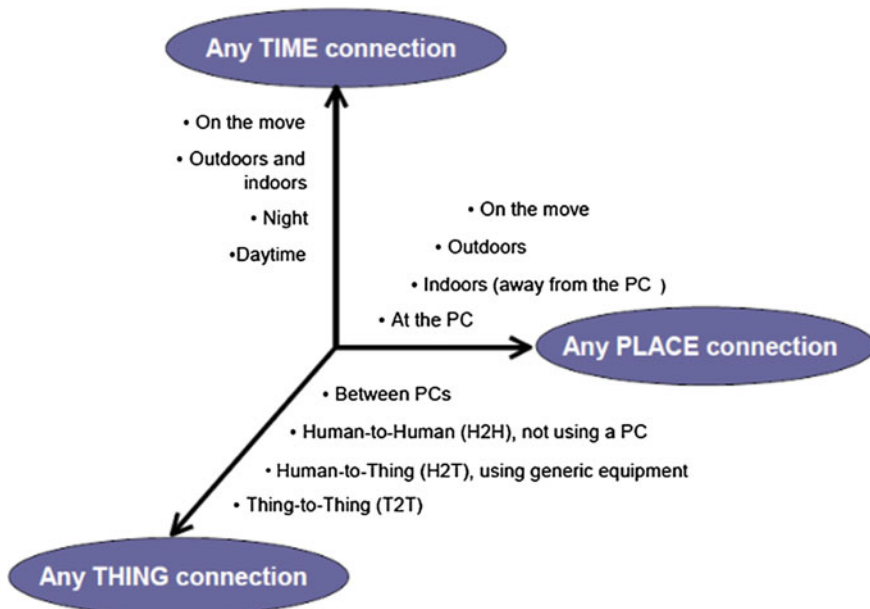


Fig. 1 Dimensions of Internet of Things [15]

- Thing-to-thing—communications include technologies and applications where daily objects and infrastructure do not have human interaction at any time of the process. In machine-to-machine scenario, the communication is a subset of the communication thing-to-thing, however this communication usually exists in large-scale IT systems and include things that cannot be classified as “daily objects”.

This new paradigm provides solutions for a wide range of applications such as: intelligent cities, traffic congestion, waste management, safety, emergency services, logistics, retail, industrial control and health care [18]. Among the variety of applications for IoT, service in human health represents one of the most attractive area of application [19]. According to the authors, a notable use case is Health-IoT, what may supply health services in the cloud and therefore improve the lives of many people.

In this way, IoT has potential to originate countless medical applications such as: remote health, fitness programs and chronic diseases monitoring. Another important application is adherence to treatment and medication at home by health professionals. For this, various medical devices, sensors and medical diagnostic devices should be developed as smart devices or objects that are an essential part of IoT [18].

E-Health concept describes a “customer-centered model” of health systems, which combines information and communication technologies to provide benefits to health from individual to public context [11]. Health services based on IoT cover from hospitals, clinics and diagnostic centers to home and portable facilities, reducing costs and improving patients’ wellbeing. Figure 2 illustrates health services in IoT context.

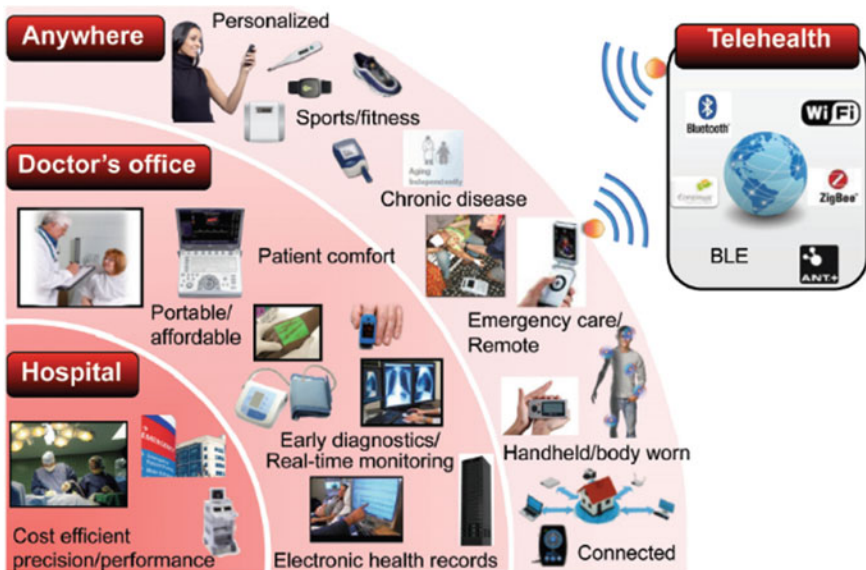


Fig. 2 Healthcare trends [20]

Table 1 IoT devices

Appliance	Health data
Micro oven	Food consumption data
Smart TV	Physical inactivity of user
Mimo	Temperature, sleep and breathing
Listener	Baby's cry detection
Sproutling	Heart rate, temperature
Owlet baby care	Oxygen level and heart rate
Sensible baby	Movement, temperature and breathing
Withings home	Analyzes local sound for signs of distress
Pacif-i	Temperature and boundary check for kid
Emospark	Emotion text and content analysis
EAR-IT	Acoustic event detection

Moreover, IoT architectures are accelerating data sharing among applications and are enabling the inference of derived attributes related to health. The Micro oven, for example, may provide data related to food consumption that can be used to identify a person's eating habits. Table 1 shows others devices and their association to health services [21].

Finally, information management in e-health context supports a wide range of clinical procedures, reducing costs and improving service availability [11]. In a technological perspective, Health-IoT is an emerging and promising research topic, due to the increasing number of applications that are raising challenges to be surpassed, especially [22]:

- Robustness in connectivity;
- Interoperability and standardization;
- Naming and identity management;
- Safety and security of objects;
- Data confidentiality and encryption;
- Data integrity and storage.

3 IoT and Sleep Medicine

The International Classification of Sleep Disorders, Diagnosis and Coding Manual, third edition (ICSD-3), lists more than 50 sleep disorders [2] organized in 7 major categories: (i) insomnias; (ii) sleep-related breathing disorders; (iii) central disorders of hypersomnolence; (iv) circadian rhythm sleep-wake disorders; (v) parasomnias; (vi) sleep-related movement disorders; and (vii) other sleep disorders. Some medical conditions are correlated with sleep disturbances, including: obesity, diabetes, cardiovascular disease, hyperactivity disorder and early mortality [23], making sleep disorders an important public health issue.

The phenomenological experience of sleep as a cessation of waking activity is misleading. On the other hand, it constitutes, like a switch, of a simple mechanism by which are shut off all neurophysiological processes associated with an active and costly wake state of vigilance [24]. In a simple observational approach, sleep is characterized by physical quiescence, closure of eyes, reduced responsiveness to external stimulation and a stereotypic body posture. However, it is not sufficient, just by means of behavioral observation, to assert that the individual under investigation is sleeping.

Two main processes are involved in timing and duration of sleep: homeostatic and circadian [25]. In a normal day-night cycle, the pressure for sleep continuously accumulates in the awake time and then dissipates during the night of sleep. This is the homeostatic regulation of sleep, which is counterbalanced by the circadian process—the endogenous oscillatory variation in alertness, sleep in alertness and sleep propensity during the 24 h. The circadian signal for alertness is high in early hours and is low in the evening, the reverse period is true to sleep signal. The interaction between these two processes explains various aspects of human behavior, due to its influence on variations in cognitive performance, especially in the attentional domain [26, 27].

Sleep is composed by two distinct physiological stages: non-rapid eye movement (NREM) and paradoxical or rapid eye movement (REM). In humans, NREM sleep is further subdivided into three stages according to sleep depth [28]. The first stage is characterized by drowsiness and sleep onset, which is followed by the light sleep stage; the third one is known as slow-wave sleep (SWS). Under normal nocturnal conditions, NREM precedes REM sleep within an ultradian cycle lasting on average 90 min and repeating itself. At the end of a typical night, five cycles should be completed [29], as shown in Fig. 3.

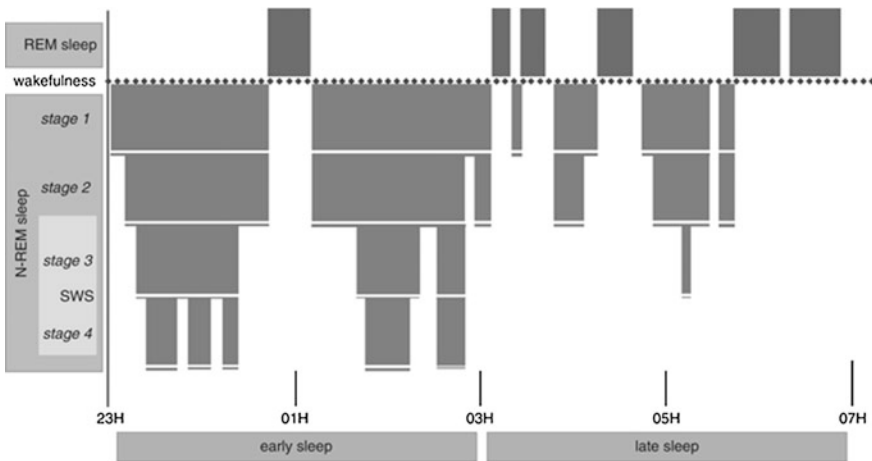


Fig. 3 Five sleep cycles [29]

REM and NREM stages exhibit different oscillation patterns of the electrical activity of the brain, as can be observed through electroencephalographic (EEG), electrooculographic (EOG), and electromyographic (EMG) signals. The classification and codification of sleep patterns still represent a proficuous research field, mainly due to the development of new technologies for acquisition and processing of biomedical signals. There are international consensus guidelines adopted to analyze data from polysomnographic laboratories, described in the “Sleep Scoring Manual” [30], in which the different stages of sleep and wakefulness are delineated. This manual is constantly updated in order to go along with technological and methodological advances.

In Health-IoT, some pilot studies have developed solutions in sleep monitoring through the identification of sleep stages and their duration. In this way, [31] proposed a simplified polysomnography system, focusing on data processing methods and sleep stages classification. Han et al. [32] describes a Smart Sleep Care System that controls each sleep period, from the sleep induction to full awakening, taking into consideration biosignal measurements and sleep analysis engines.

Although it is required a certain accuracy of the electroencephalographic (EEG), electrooculographic (EOG), and electromyographic (EMG) signals, to perform sleep disorders diagnosis these data are inconvenient to acquire during sleep, since electrodes should be attached to patient body. For this reason, low invasive approaches are more suitable for continuous monitoring. For instance, it is possible to use EEG tests before and after sleep sessions, since it is expected brain changes due to the physiology of sleep.

Some parameters can be used to assess sleep quality; the following variables are frequently reported and considerably most important regarding sleep patterns [33–35]

- Sleep onset latency: it is obtained by estimating the length of time that the individual under scrutiny takes to accomplish the transition from full wakefulness to sleep;
- Total sleep time: it is equal to the total sleep episode deducting the awake time;
- Wake after sleep onset: it is defined as total amount of time awake excluding the sleep onset latency;
- Total wake time: it is the time that the patient is awake during the observation;
- Sleep efficiency: it is the ratio between the total sleep time and the total amount of time spent in bed.

Other parameters are also relevant to sleep disorders diagnosis, like total time in bed, number of naps during the day and number of times that the patient wakes up during the night, for instance [36, 37]. In addition to the laboratory polysomnography, actigraphy devices are also used to support sleep assessment. Usually, actigraphy devices are worn on the non-dominant wrist to measure gross motor activity. Actigraphy has been well validated for the estimation of nighttime sleep parameters across age groups [38].

Furthermore, self-applied questionnaires and sleep diaries are also useful to sleep assessment [39]. There are plenty of them such as: Pittsburgh Sleep Quality Index; Epworth Sleepiness Scale; Sleep Intervention Acceptability Scale; MI Acceptability Questionnaire; Stanford Sleepiness Scale (SSS) and sleep diaries. These questionnaires have different sensitivity and specificity for sleep disorders categories. For this reason, the integration between them should be taken into consideration in the first investigations.

As mentioned previously, sleep disorders usually lead to behavioral changes. As consequence, it impacts the cognitive performance. Timmers et al. [40] demonstrates that smartphones can be used to assess cognitive functions outside a laboratory setting. Thus, it has been proved that cognitive self-applied tests using smartphones can give important clues about attention deficit and anxiety, which are usually associated with sleep deprivation. Smartphones usage data can also be used with this purpose, once that depression, anxiety, and sleep quality may be associated with smartphone overuse [41].

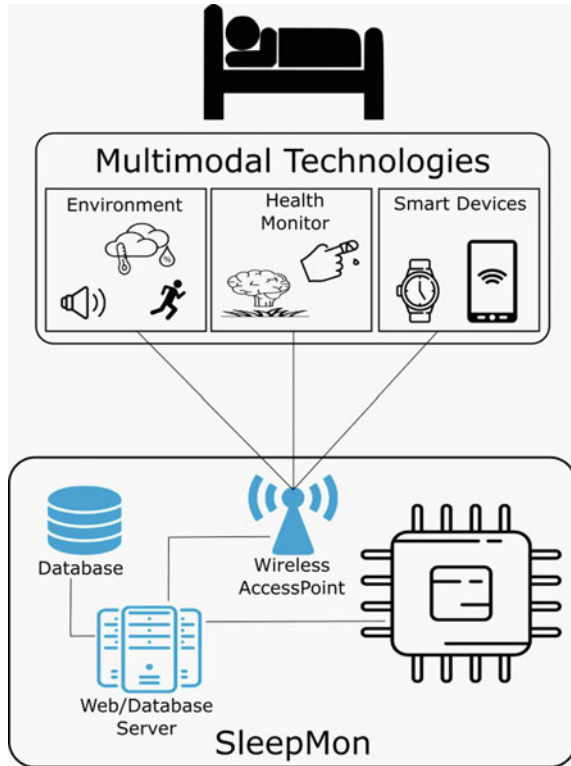
In this section, several methods for sleep assessment were outlined. As discussed in [42], each method has advantages and limitations, and provides different vantage points into the complexity of sleep physiology. For instance, the multi-channel data obtained by a polysomnogram laboratory are useful to study the mechanisms of sleep and their relationships. However, the single-night paradigm does not provide enough information about longitudinal variations from night to night. On the other hand, in home monitors including actigraphy, longitudinal variation can be observed while the nuances of sleep physiology remain hidden.

In view of these facts, the adoption of multiple methods makes sleep diagnosis more accurate, once more truthfulness information are provided to physicians and technical staff, assisting in the screening process, reducing costs and helping to improve the wellbeing of people with sleep disorders. In the next section it is presented a non-invasive sleep-environment monitoring system developed to the detection of environmental factors that may be contributing to poor sleep. In addition to this device, multimodal technologies are also expected to compose the sleep assessment system, such as smart devices (e.g. phones, bands and watches) and e-health monitors. These things interconnected are able to provide enough information to guide sleep disorder diagnosis and improvement.

4 Monitoring Sleep in Multimodal IoT Context

As explored in Sect. 3, new monitoring technologies made possible the data acquisition correlated to sleep quality with non-invasive devices. In this context, various mechanisms were developed to improve the wellbeing of people by monitoring sleep disorders. In this session we present a device developed by [10] that makes use of some of these technologies and other sensors to monitor sleep. The proposed architecture is shown in Fig. 4 and it is composed by:

Fig. 4 Proposed system architecture



- Multimodal technologies: a layer formed by devices and sensors responsible for capturing parameters related to the environment and health data. Health data can be obtained either by specific health monitors (e.g. sugar level and EEG) or smart devices applications;
- SleepMon: represents the middleware responsible for fusion, analysis and storage of data acquired in the bottom layers. The results are displayed in a web-based system, as in several other IoT systems.

As delimited by [43, 44], in IoT scenario, a middleware provides the infrastructure required by services such as data acquisition and management. Usually, it is based on the Service-Oriented Architecture (SOA) pattern. [43] also state that data privacy and information security are important topics. This fact becomes critical in health data so it is noteworthy the importance of having standard security architecture support for SOA-based IoT middleware. Some services could be used to ensure the protection of the entire middleware regardless of the specific requirements of any application domain [43].

4.1 Non-invasive Sleep-Environment Monitoring System

In this section the sleep-environment monitoring systems is described. It is important to note that all aspects were studied with the aim to make a scalable, robust, built-in and stand-alone system, providing more accurate information to technical staff and physicians involved in sleep assessment, and also to compare self-perception (usually the first information that physicians have) with real facts. Moreover, it can be integrated to other “things”, such as wearable devices, cognitive tests available in smartphone applications, social network data etc. Following this, the variables monitored by the system will be presented for further description of their function.

Parameters

Some environmental parameters directly impact sleep quality, such as temperature, humidity noise-level, noise-frequency and luminosity. Others variables, as motion, give important clues about the person’s rest time. These parameters are discussed below:

- Temperature:** the ambient temperature exerts a prominent influence on sleep. In rats and humans, low ambient temperatures generally impair sleep, whereas higher temperatures tend to promote sleep. Additionally, it is important to note that in humans, a lower core temperature coupled with a higher distal (hands and feet) temperature before sleep are associated with shorter sleep latency and better sleep quality [45]. Figure 5 shows the curve of body temperature variation during the circadian cycle. Thus, it is ideal that the temperature keeps stable and avoids low/high ambient temperatures that can cause discomfort and, consequently, harm the sleep;

Fig. 5 Typical curve of body temperature variation during the day

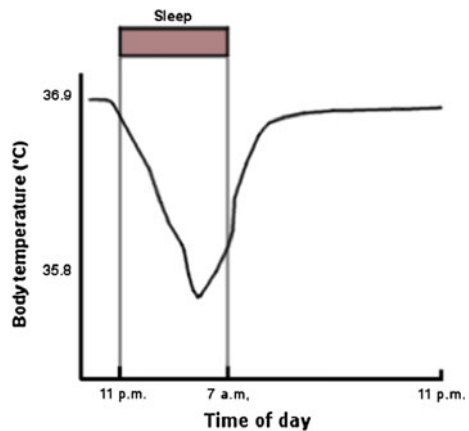
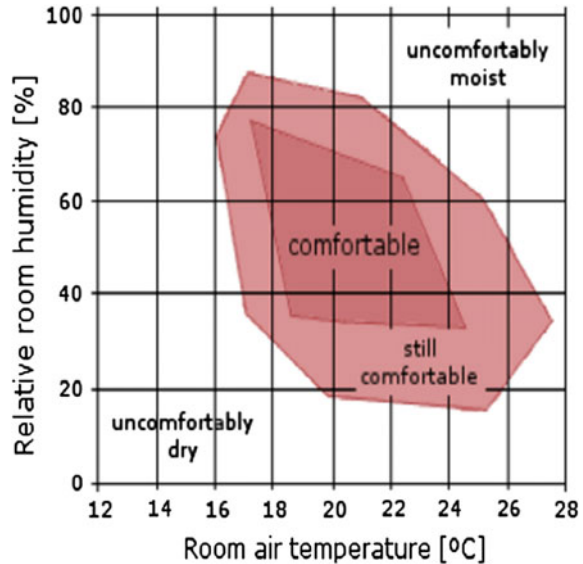


Fig. 6 Relationship between humidity and temperature and its impact in the environmental comfort



- Humidity:** this parameter has an effect on heat transfer and, consequently, on sleep quality. As pointed by [46], a number of hypotheses have been proposed to explain the sleep structure regulation. The circadian-homeostatic interaction model is the most accepted hypothesis; it suggests that humid and heat may affect sleep through the homeostatic pathway, possibly interfering with adenosine accumulation in the basal forebrain and thereby affecting the non-rapid eye movement (NREM) sleep switch point. The association between extreme humidity and respiratory diseases, which has deleterious effects on sleep, is also well established. Figure 6 shows the relationship between humidity and temperature and its impact in the environmental comfort;
- Noise:** temperature and humidity are not considered to be as significant to the assessment of sleep quality as noise, thus, the third and fourth parameter (noise-level and noise-frequency) are related with this feature. This fact is corroborated by a large number of public health studies involving this theme, specially evaluating road traffic and aircraft noise [47]. A consensus regarding noise is that it has harmful effects on sleep structure, especially in terms of sleep fragmentation [48]. Under normal conditions, peaks of noise trigger partial arousal or micro-awakenings, often not consciously recognized or remembered, which makes it hard to map the causes. The third and fourth parameters are therefore related to noise characteristics because both can provide important clues about sleep health. High noise-levels and specific noise-frequency signatures can be correlated with other parameters such as bed-vibration, motion and the simple sleep anamnesis can be used to detect obstructive sleep apnea by means of snoring sounds variability signature [49];

- **Luminosity:** the impact of artificial lighting on the circadian cycle is clear. It occurs because bright artificial light suppresses the nocturnal secretion of melatonin, the “sleep hormone” [50]. Furthermore, humans show the strongest response of melatonin suppression in the short-wavelength portion of the light spectrum, in other words, the blue lights. Other interesting fact pointed out by [51] is that blue monochromatic light has also been shown to be more effective than longer-wavelength light for enhancing alertness and disturbing circadian rhythm. Thus, measuring the light behaviour in the bedroom, in terms of white and blue luminous intensity, can provide evidence for problems in sleep latency;
- **Motion:** this is another important feature to record in order to evaluate how calm was the rest. By storing information about the sleeper movements, it is possible to correlate it with other information, such as: noise-levels, temperature and humidity and a simple sleep anamnesis to better investigate the triggers of partial arousals and micro-awakenings and sleep self-perception. It is important to point out that wrist actigraphy records the same information, but does not take into account legs movement as ultrasonic-based sensors, or even camera, as proposed by [52–54] for instance.

4.2 Installation and Usage

The non-invasive sleep environment monitoring system was designed to be completely built-in and to work as a stand-alone in-home device, as shown in Fig. 7a. Its installation and use are ubiquitous, since the user does not have to give any inputs. The position should be on bedside, preferably placed behind the bed as illustrated in Fig. 7b.

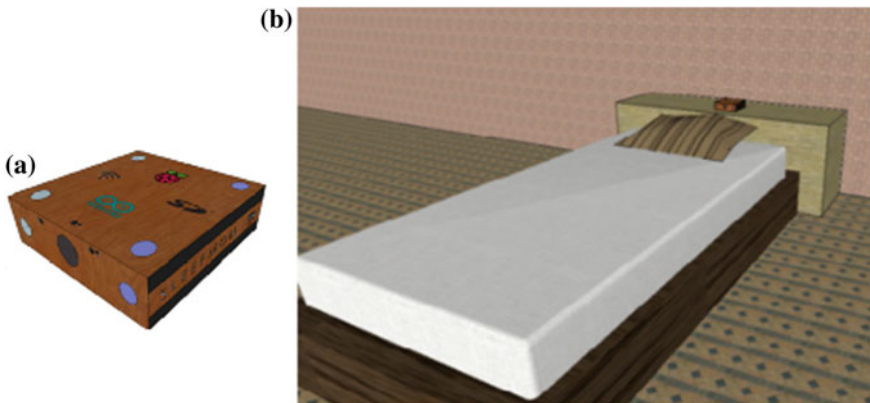


Fig. 7 Example of device installation on the bedside

The data acquisition process is triggered by a combination of factors, like movement and presence in the bed, sleep schedules/diaries and other test scores.

4.3 *Smart Devices and Health Monitors*

There are several sleep monitors available for smartphone users; however, their utility is not restricted to this. Smartphones can be used to apply cognitive tests as well. This assertion is reinforced by the emphasis that behavioral sleep medicine has gained in recent years. In addition to that, candidate models and mechanisms associated with insomnia development and persistence, which are being proposed and validated [55].

Moreover, these findings, combined with the intense adoption and use of smartphones, are unleashing many studies associated to the usage of these technologies and health problems such as stress and anxiety [56–59]. According to [57], the Problematic Mobile Phone Use is associated with chronic stress, low emotional stability, female gender, young age, depression, and extraversion. Among the causes, the smartphone use after sleep onset has the greatest impact on sleep quality. These facts, combined with the sleep monitoring system proposed, make the smartphones a valuable data source for building behavioral sleep models.

In addition to smartphone usage data, ad-hoc apps can also be used to provide self-applicable questionnaires, such as the one shown in Fig. 8. The choice of the questions asked can be controlled by a specialized system according to the problems, behavioral patterns and personality described by the patient [60].

Smartphones and smartbands/watches are also useful for sleep monitoring systems, since the behavioural data acquired by these devices can be used in sleep assessment [61]. This monitoring can identify sedentarism for example, which affects many sleep disorders, among them the obstructive sleep apnea [62].

As stated previously, biological signals monitors as electromyography clinic and electroencephalogram are becoming more accessible to the general public, through electrodes and helmets of simple installation and use [63]. Despite its use during sleep be a bit uncomfortable, the EEG can be used to help the sleep efficiency measurement via tests before and after sleep since the signature of cerebral waves change substantially after resting.

Other monitors can be used to identify secondary factors that may affect the quality of sleep such as blood glucose and blood pressure. It is known that diabetes and hypertension, when not controlled, contribute to the decrease of sleep quality. Considering that most devices used to monitor these factors do not have connectivity, it is possible to provide an app for the patient, together with a questionnaire as the one presented in Fig. 8, to acquire this data.

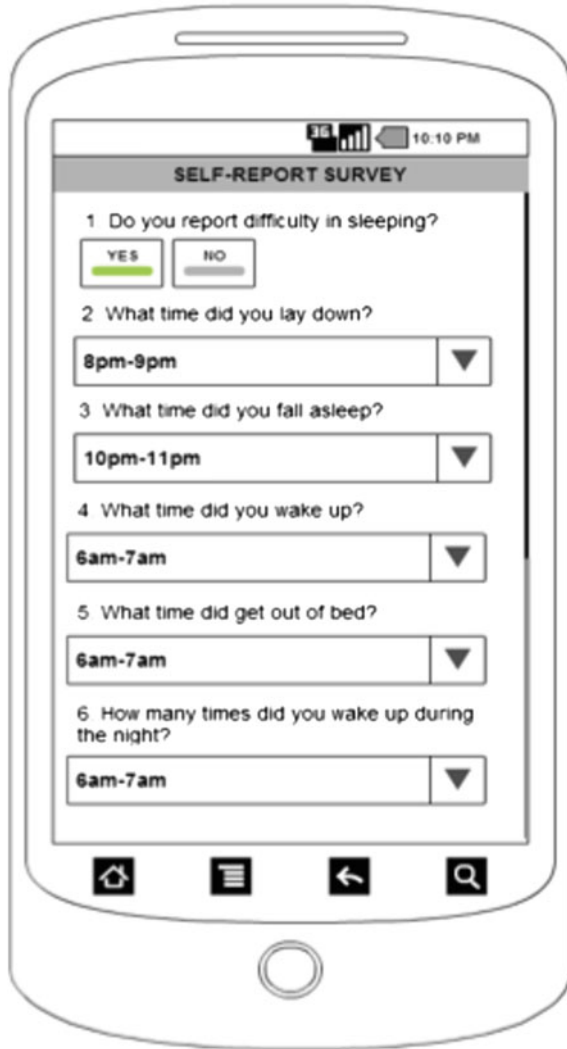


Fig. 8 Example of a self-applicable questionnaire for a sleep diary

5 Final Remarks

The attention on sleep disorders has grown in recent years, mainly because of the changes imposed by the information and communication Era. The data wave and indiscriminate usage of mobile devices impacted in a hazardous way our sleep quality. Thus, it represents a public health issue because sleep disorders are

positively related with some medical conditions, such as: obesity, diabetes, hypertension, cardiovascular disease, stroke, hyperactivity disorder, mental issues and early mortality. The current mainstream sleep disorder detection, the laboratory polysomnography, is very expensive and inconvenient for patients who are extracted from their own sleep-environment. Other sleep assessment devices have been proposed in order to avoid the high costs, the single-night paradigm and to make the observation more comfortable to the patient. Examples of these devices are wristbands, smartwatches and smartphones with specific applications. However, these devices are not interconnected, consequently, a fair amount of data is neglected.

In this context, the Internet of Things paradigm provides models, architectures and methods to interconnect the most diverse types of things, which offers potentially useful information for sleep disorders diagnosis. There are many devices that can give clues about sleep quality. Through their interconnection, it is possible to provide more accurate information to physicians and technical staff; assisting in the screening process, reducing costs and helping to improve the wellbeing of people with sleep disorders.

In this chapter, it was presented an overview of IoT use in healthcare context to later present a theoretical background on sleep medicine. At this point, several variables which are crucial to the monitoring and evaluation of sleep quality were analyzed. Parameters such as sleep onset latency, total sleep time, wake after sleep onset, total awake time and sleep efficiency were explored, highlighting their importance to the application domain studied. Next, it was analyzed a few methodologies and technologies. The combination of them brings great potential to diagnose sleep disorders. In order to do that, multimodal technologies were categorized according to three classes: sleep environment monitoring, health monitors and intelligent devices—where multimodality confers more robustness to further analysis.

As result, we expect that the literature review presented, together with the architecture described, can guide future research to tackle some challenges observed, such as the interconnection between proprietary sleep care devices which in IoT scenario may provide valuable data. This enables a precise and low cost sleep monitoring system that can help physicians to identify and diagnose sleep disorders. This chapter also presents an example of a multimodal low-invasive device that works based on the interconnection of several smart sensors and makes use of IoT principles to be able to acquire enough information to physicians by making an accurate observation of the patient.

Besides helping the sleep monitoring, aiming to provide a more accurate information to the technical staff involved in the diagnosis and treatment of sleep disorders, the proposed system can be also used to improve the sleep quality of a person in three ways: (i) through the use of intelligent devices for a softer awakening replacing alarm clocks that cause an abrupt awakening; (ii) the construction of a sleep routine adaptable to the patient life; and (iii) the improvement of self-knowledge on health harmful behaviours.

References

1. F. A. Roshan, K. Radecka, and Z. Zeljko, "Design and Evaluation of an Intelligent Remote Tidal Volume Variability Monitoring System in E-Health Applications," *Biomed. Heal. Informatics, IEEE J.*, vol. 19, no. 5, pp. 1532–1548, 2015.
2. M. J. Sateia, "International classification of sleep disorders-third edition: highlights and modifications.," *Chest*, vol. 146, no. 5, pp. 1387–94, Nov. 2014.
3. T. Roth, "Insomnia: Definition, Prevalence, Etiology, and Consequences," *J. Clin. Sleep Med.*, vol. 3, no. 5 Suppl, pp. S7–S10, Aug. 2007.
4. C. W. Karlson, M. W. Gallagher, C. A. Olson, and N. A. Hamilton, "Insomnia symptoms and well-being: Longitudinal follow-up.," *Heal. Psychol.*, vol. 32, no. 3, pp. 311–319, Mar. 2013.
5. L. Korpinen and R. Pääkkönen, "Self-reported sleep disorders/disturbances associated with physical symptoms and usage of computers," *Int. J. Ind. Ergon.*, vol. 43, no. 4, pp. 257–263, 2013.
6. J. F. Pagel and C. F. Kwiatkowski, "Sleep Complaints Affecting School Performance at Different Educational Levels," *Front. Neurol.*, vol. 1, p. 125, Nov. 2010.
7. C. E. Carney, J. D. Edinger, B. Meyer, L. Lindman, and T. Istre, "Daily activities and sleep quality in college students," *Chronobiol. Int.*, vol. 23, no. 3, pp. 623–637, Jan. 2006.
8. O. J. Walch, A. Cochran, and D. B. Forger, "A global quantification of 'normal' sleep schedules using smartphone data," *Sci. Adv.*, vol. 2, no. 5, May 2016.
9. L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
10. F. Lobato, B. Silva, R. Bem, and D. Miranda, "Non-invasive sleep-environment monitoring system," in *PETRA'15: Proceedings of the 8th International Conference on Pervasive Technologies Related to Assistive Environments*, 2015.
11. I. Chiuchisan, H.-N. Costin, and O. Geman, "Adopting the internet of things technologies in health care systems," in *Electrical and Power Engineering (EPE), 2014 International Conference and Exposition on*, 2014, pp. 532–535.
12. O. Vermesan and P. Friess, *Internet of Things - From research and innovation to Market Deployment*. River Publishers, 2014.
13. A. Zimmermann, R. Schmidt, K. Sandkuhl, M. Wißotzki, D. Jugel, and M. Mohring, "Digital Enterprise Architecture-Transformation for the Internet of Things," in *Enterprise Distributed Object Computing Workshop (EDOCW), 2015 IEEE 19th International*, 2015, pp. 130–138.
14. M. Y. Suraki, M. Y. Suraki, and O. Nejati, "Benefit of internet of things to improve business interaction with depression prevention and treatment," in *e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on*, 2012, pp. 403–406.
15. ITU, "ITU Internet Reports 2005: The internet of things," 2005.
16. L. Tan and N. Wang, "Future internet: The internet of things," in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, 2010.
17. S. C. B. Intelligence, "Appendix F: The Internet of things (background)," 2008.
18. S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, "The internet of things for health care: a comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.
19. C. Seales, T. Do, E. Belyi, and S. Kumar, "PHINet: A Plug-n-Play Content-centric Testbed Framework for Health-Internet of Things," in *Mobile Services (MS), 2015 IEEE International Conference on*, 2015, pp. 368–375.
20. K. Vasanth and J. Sbert, "Creating solutions for health through technology innovation," *Texas Instruments*, 2013.
21. H. Anumala and S. M. Busetty, "Distributed Device Health Platform Using Internet of Things devices," in *2015 IEEE International Conference on Data Science and Data Intensive Systems*, 2015, pp. 525–531.
22. G. S. Matharu, P. Upadhyay, and L. Chaudhary, "The Internet of Things: Challenges & security issues," in *Emerging Technologies (ICET), 2014 International Conference on*, 2014, pp. 54–59.

23. O. M. Buxton and E. Marcelli, "Short and long sleep are positively associated with obesity, diabetes, hypertension, and cardiovascular disease among adults in the United States," *Soc. Sci. Med.*, vol. 71, no. 5, pp. 1027–1036, 2010.
24. C. M. Morin, C. A. Espie, P. Peigneux, C. Urbain, and R. Schmitz, "Sleep and the Brain," in *The Oxford Handbook of Sleep and Sleep Disorders*, "Oxford University Press," 2012.
25. S. Daan, D. G. Beersma, and A. A. Borbely, "Timing of human sleep: recovery process gated by a circadian pacemaker," *Am. J. Physiol. - Regul. Integr. Comp. Physiol.*, vol. 246, no. 2, pp. R161–R183, Feb. 1984.
26. C. Schmidt, F. Collette, C. Cajochen, and P. Peigneux, "A time to think: Circadian rhythms in human cognition," *Cogn. Neuropsychol.*, vol. 24, no. 7, pp. 755–789, Oct. 2007.
27. C. Schmidt, F. Collette, Y. Leclercq, V. Sterpenich, G. Vandewalle, P. Berthomier, C. Berthomier, C. Phillips, G. Tinguely, A. Darsaud, S. Gais, M. Schabus, M. Desseilles, T. T. Dang-Vu, E. Salmon, E. Baletau, C. Degueldre, A. Luxen, P. Maquet, C. Cajochen, and P. Peigneux, "Homeostatic Sleep Pressure and Responses to Sustained Attention in the Suprachiasmatic Area," *Science (80-.)*, vol. 324, no. 5926, pp. 516–519, Apr. 2009.
28. M. H. Silber, S. Ancoli-Israel, M. H. Bonnet, S. Chokroverty, M. M. Grigg-Damberger, M. Hirshkowitz, S. Kapen, S. A. Keenan, M. H. Kryger, T. Penzel, M. R. Pressman, and C. Iber, "The visual scoring of sleep in adults," *J. Clin. Sleep Med.*, vol. 3, no. 2, pp. 121–131, 2007.
29. C. D. Clemente, "Some Must Watch While Some Must Sleep. William C. Dement," *Q. Rev. Biol.*, vol. 50, no. 3, pp. 358–359, Sep. 1975.
30. S. F. Iber, C., Ancoli-Israel, S., Chesson, A., & Quan, *The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specification*. 2007.
31. O. R. Velicu, N. Mart, and R. Seepold, "Experimental sleep phases monitoring," in *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, 2016, pp. 625–628.
32. H. Han, J. Jo, Y. Son, and J. Park, "Smart sleep care system for quality sleep," in *n Information and Communication Technology Convergence (ICTC), 2015 International Conference on*, 2015, pp. 393–398.
33. A. G. Harvey, K. Stinson, K. L. Whitaker, D. Moskovitz, and H. Virk, "The Subjective Meaning of Sleep Quality: A Comparison of Individuals with and without Insomnia," *Sleep*, vol. 31, no. 3, pp. 383–393, Mar. 2008.
34. A. D. Krystal and J. D. Edinger, "Measuring sleep quality.," *Sleep Med.*, vol. 9 Suppl 1, pp. S10–7, Sep. 2008.
35. N. C. Van Wouwe, P. J. L. Valk, and B. J. Veenstra, "Sleep Monitoring: A Comparison Between Three Wearable Instruments.," *Mil. Med.*, vol. 176, no. 7, pp. 811–816, 2011.
36. C. E. MILNER and K. A. COTE, "Benefits of napping in healthy adults: impact of nap length, time of day, age, and experience with napping," *J. Sleep Res.*, vol. 18, no. 2, pp. 272–281, Jun. 2009.
37. M. M. Ohayon, A. Krystal, T. A. Roehrs, T. Roth, and M. V Vitiello, "Using difficulty resuming sleep to define nocturnal awakenings," *Sleep Med.*, vol. 11, no. 3, pp. 236–241, Mar. 2010.
38. J. L. Martin and A. D. Hakim, "Wrist actigraphy.," *Chest*, vol. 139, no. 6, pp. 1514–27, Jun. 2011.
39. L. G. Sylvia, S. Salcedo, M. T. Bianchi, A. Urdahl, A. A. Nierenberg, and T. Deckersbach, "A Novel Home Sleep Monitoring Device and Brief Sleep Intervention for Bipolar Disorder: Feasibility, Tolerability, and Preliminary Effectiveness," *Cognit. Ther. Res.*, vol. 38, no. 1, pp. 55–61, Feb. 2014.
40. C. Timmers, A. Maeghs, M. Vestjens, C. Bonnemayer, H. Hamers, and A. Blokland, "Ambulant cognitive assessment using a smartphone," *Appl. Neuropsychol. Adult*, vol. 21, no. 2, pp. 136–142, 2014.
41. K. Demirci, M. Akgönül, and A. Akpinar, "Relationship of smartphone use severity with sleep quality, depression, and anxiety in university students," *J. Behav. Addict.*, vol. 4, no. 2, pp. 85–92, 2015.

42. M. T. Bianchi and R. J. Thomas, "Technical advances in the characterization of the complexity of sleep and sleep disorders," *Prog. Neuro-Psychopharmacology Biol. Psychiatry*, vol. 45, pp. 277–286, 2013.
43. R. T. Tiburski, L. A. Amaral, E. de Matos, and F. Hessel, "The Importance of a Standard Security Architecture for SOA -Based IoT Middleware," *IEEE Commun. Mag. — Commun. Stand. Suppl.*, vol. 53, no. 12, pp. 20–26, 2015.
44. J. M. Batalla and P. Krawiec, "Conception of ID layer performance at the network level for Internet of Things," *Springer J. Pers. Ubiquitous Comput.*, vol. 18, no. 2, pp. 465–480, 2014.
45. W.-C. Liao, L. Wang, C.-P. Kuo, C. Lo, M.-J. Chiu, and H. Ting, "Effect of a warm footbath before bedtime on body temperature and sleep in older adults with good and poor sleep: An experimental crossover trial," *Int. J. Nurs. Stud.*, vol. 50, no. 12, pp. 1607–1616, Dec. 2013.
46. M. D. Manzar, M. Sethi, and M. E. Hussain, "Humidity and sleep: a review on thermal aspect.," *Biol. Rhythm Res.*, vol. 43, no. 4, pp. 439–457, 2012.
47. M. Kim, S. I. Chang, J. C. Seong, J. B. Holt, T. H. Park, J. H. Ko, and J. B. Croft, "Road traffic noise: annoyance, sleep disturbance, and public health implications.," *Am. J. Prev. Med.*, vol. 43, no. 4, pp. 353–60, Oct. 2012.
48. M. Saremi, J. Grenèche, A. Bonnefond, O. Rohmer, A. Eschenlauer, and P. Tassi, "Effects of nocturnal railway noise on sleep fragmentation in young and middle-aged subjects as a function of type of train and sound level.," *Int. J. Psychophysiol.*, vol. 70, no. 3, pp. 184–91, Dec. 2008.
49. A. Azarbarzin and Z. Moussavi, "Snoring sounds variability as a signature of obstructive sleep apnea.," *Med. Eng. Phys.*, vol. 35, no. 4, pp. 479–85, Apr. 2013.
50. A. J. Lewy, T. A. Wehr, F. K. Goodwin, D. A. Newsome, and S. P. Markey, "Light suppresses melatonin secretion in humans," *Sci.*, vol. 210, no. 4475, pp. 1267–1269, Dec. 1980.
51. K. E. West, M. R. Jablonski, B. Warfield, K. S. Cecil, M. James, M. A. Ayers, J. Maida, C. Bowen, D. H. Sliney, and M. D. Rollag, "Blue light from light-emitting diodes elicits a dose dependent suppression of melatonin in humans," *J. Appl. Physiol.*, vol. 110, no. 3, pp. 619–626, 2011.
52. L. C.-L. Chen, K.-W. Chen, and Y.-P. Hung, "A sleep monitoring system based on audio, video and depth information for detecting sleep events," in *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, 2014, pp. 1–6.
53. A. Heinrich, D. Geng, D. Znamenskiy, J. P. Vink, and G. de Haan, "Robust and Sensitive Video Motion Detection for Sleep Analysis," *Biomed. Heal. Informatics, IEEE J.*, vol. 18, no. 3, pp. 790–798, May 2014.
54. W.-H. Liao and J.-H. Kuo, "Sleep monitoring system in real bedroom environment using texture-based background modeling approaches.," *J. Ambient Intell. Humaniz. Comput.*, vol. 4, no. 1, pp. 57–66, 2013.
55. C. A. Espie and C. M. Morin, "Introduction : Historical Landmarks and Current Status of Sleep Research and Practice : An Introduction to the Timeliness, Aims, and Scope of this Handbook," no. April 2016, 2012, pp. 1–13.
56. S. Thomée, A. Härenstam, and M. Hagberg, "Mobile phone use and stress, sleep disturbances, and symptoms of depression among young adults - a prospective cohort study," *BMC Public Health*, vol. 11, no. 1, p. 66, 2011.
57. C. Augner and G. W. Hacker, "Associations between problematic mobile phone use and psychological parameters in young adults," *Int. J. Public Health*, vol. 57, no. 2, pp. 437–441, 2011.
58. S. K. Adams and T. S. Kisler, "Sleep Quality as a Mediator Between Technology-Related Sleep Quality, Depression, and Anxiety," *Cyberpsychology, Behav. Soc. Netw.*, vol. 16, no. 1, pp. 25–30, 2013.
59. J. Billieux, P. Philippot, C. Schmid, P. Maurage, J. De Mol, and M. Van der Linden, "Is Dysfunctional Use of the Mobile Phone a Behavioural Addiction? Confronting Symptom-Based Versus Process-Based Approaches," *Clin. Psychol. Psychother.*, vol. 22, no. 5, pp. 460–468, 2015.

60. C. G. DeYoung, "Cybernetic Big Five Theory," *J. Res. Pers.*, vol. 56, pp. 33–58, 2015.
61. K. R. King, L. P. Grazette, D. N. Paltoo, J. T. McDevitt, S. K. Sia, P. M. Barrett, F. S. Apple, P. A. Gurbel, R. Wissleder, H. Leeds, E. J. Iturriaga, A. K. Rao, B. Adhikari, P. Desvigne-Nickens, Z. S. Galis, and P. Libby, "Point-of-Care Technologies for Precision Cardiovascular Care and Clinical Research," *JACC Basic to Transl. Sci.*, vol. 1, no. 1–2, pp. 73–86, 2016.
62. K. M. C. Kumar and C. Engineering, "A New Methodology for Monitoring OSA Patients Based on IoT," vol. 5, no. 2, pp. 298–302, 2016.
63. M. Swan, "Sensor Mania! The Internet of Things, Wearable Computing, Objective Metrics, and the Quantified Self 2.0," *J. Sens. Actuator Networks*, vol. 1, no. 3, pp. 217–253, 2012.

On Real Time Implementation of Emotion Detection Algorithms in Internet of Things

Sorin Zoican

Abstract This chapter describes the methods for detecting the human emotion using signal processing techniques and their implementation in real time. The first sections present the basic approaches both for emotion detection using face images and speech signals. This work highlights the tradeoff between detection performance and algorithm complexity and describes the architectures of microcontrollers used to implement the algorithms for emotion detection (including preprocessing and basic tasks) and methods for code optimization. These optimizations are made for a real time realization on mobile devices (e.g. smart phones, tablets). Preprocessing tasks run on mobile device and the basic tasks may be run on a server or on device. Finally, the chapter estimates computational effort and memory requirements for image and speech processing involved in emotion detection.

1 Motivation of Detecting the Emotion in Internet of Things

Nowadays many smart devices and appliances are connected one to another using high technology with cognitive intelligence but no emotional intelligence. If the technology could detect the human emotion, these devices will bring positive behavior change, suggesting how our actions become better.

Possible applications of the emotion-aware wearable are: intelligent home, automotive, telemedicine, on-line education, shopping recommendation based on your emotion, social robots and interactive games.

An intelligent home has various sensors (video and acoustic) that may capture the human face image and voice signals that will use to discover the mood and

S. Zoican (✉)

Electronics, Telecommunications and Information Technology Faculty,
Telecommunication Department, POLITEHNICA University of Bucharest,
Bucharest, Romania
e-mail: sorin@elcom.pub.ro

create an adequate ambiance. The devices interact with the person to make his or her to take the best action (such as adjust lights according to your mood).

In automotive, detecting the driver emotion will be useful because the road rage can be managed. In this situation, the car engine can be adequately controlled despite the manual controls of the driver. In such application, other signals may detect the driver emotion (electrocardiogram, sweating).

Monitoring the mental health is another important application of the emotion detection. The personal smart phone can detect an ill condition of the person and alarm the medical center to manage the situation.

In on-line education, the educational content can be adapted to the mood of the students, so the educational process will have a greater performance.

The last three possible applications are related to the social behavior—we would be advised to shop things, interact with machines, robots or games.

An emotion-aware Internet of Things will bring these applications in the “smart society”, and has the potential to transform major industries.

This chapter intends to describe how the human emotion may be detected using signal processing techniques and how these techniques can be implemented in a real time system.

The emotion detection can be achieved by face image processing (eyes, lips) and speech processing. For both processing methods specific features are extracted from image or audio signal and mapped by a classifier during a training session. The collected data are preprocessing in the device (e.g. windowing, pre-filtering, conditioning, edge detection, segmentation) then the pre-processed data are sent to a server (or a cloud) for complex processing to decide what are the emotional state sent back to the device that will suggest a change in our behavior or will propose to take a proper action. The detection process has many complex tasks to be completed, so the detection has not 100 % accuracy.

Important problems to tackle are what an emotion is, and how the emotions are measured. The following issues should be analyzed [1]: (a) the event that triggers an emotion, (b) what are the physiological symptoms, (c) what are the motor expression of the person, (d) what are the action tendencies and (e) what are the feelings triggered by that event.

For a given event that will trigger an event the following items should be considered: the frequency and the suddenness of the event occurrence, the importance of the event, and the cause of the event (natural or artificial) and how the event determines an emotion. The knowledge of the event that determines a specific emotion is very important because it confirms the detected emotion.

The emotion will trigger physiological symptoms (such as feeling cold or warm, weak limbs, pale face, heartbeats slowing or faster, muscles relaxing or tensing, breathing slowing or faster, sweating). Most of these symptoms can be measured by observing the corresponding motor expression (face changes—mouth, eyes and eyebrow expressions, voice volume changes, body movements). The measurements of the motor expression underlie the emotion detection algorithms. The action tendencies (moving towards or away the event) may confirm the detected emotion. After measuring the motor expression, a training process is performed to detect

correctly the emotion. Based on this training process the emotions are classified in neutral, joy, sad, angry, happiness, pride, hostility etc. The feelings such as intensity and duration should be determined to classify better the emotion. The device must respond to our emotion fast enough and should have an architecture based on a powerful microcontroller and a quick network communication.

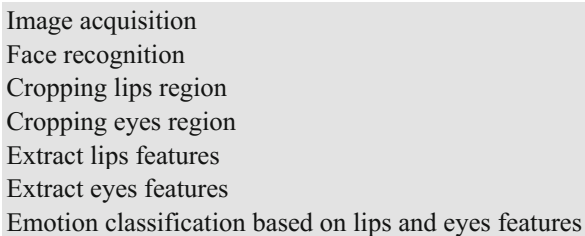
Algorithms for detecting emotions may have high complexity, especially those based on image analysis. It is desirable that emotions to be detect in real time (in seconds or less) using mobile devices that have relatively limited memory amount and use fixed point processors. It is therefore necessary to use specific hardware and software techniques such as hardware loops, multifunction instructions, multi-core processors, and to adopt simplified but effective emotion detection algorithms.

There are a lot of algorithms involved in detection of emotions [2]. In this chapter we focus on the algorithms possible to be implemented on a wearable device (smart phone or tablet) without sensors to be attached to the body (e.g. electrodes). The only input devices that will be used in emotion detection will be the microphone and video camera existing on the portable device so the emotion detection algorithms will use face and voice changes. We focus on the possibility to detect the emotion, based on the best algorithms described in literature, in real time. These algorithms may be simplified, without degrade the performance, to meet this goal.

The methods, techniques and microcontrollers' architectures presented and evaluated in this chapter meet the requirements of a real time implementation of emotion detection algorithms.

2 Emotion Detection Using Frontal Face Image Processing

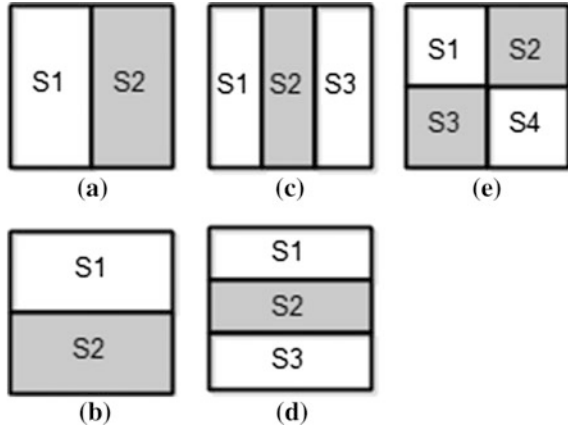
Most of human emotions can be visualized by face expression [3, 4, and 5]. The human face gives clues very useful to detect our mood. Especially the eyes and the mouth will express the best the emotion such as joy, sadness, angry, fear, disgust etc. The emotion detection process phases from image of frontal face are depicted in Fig. 1.



- Image acquisition
- Face recognition
- Cropping lips region
- Cropping eyes region
- Extract lips features
- Extract eyes features
- Emotion classification based on lips and eyes features

Fig. 1 The emotion detection process phases

Fig. 2 The five patterns used in face detection



The challenging tasks in the above process are: face recognition, extract lips and eye features and emotion classification. Many methods to detect a face in an image exist. The most known algorithm is Viola—Jones [6, 7]; it is highly time-consuming and is implemented usually on desktop computers. Other simpler methods, but effective, are based on skin detection followed by image segmentation to find the face region [5]. These methods have the advantage to be less time-consuming and they are suitable to be implemented on mobile devices. The face recognition will be performed scanning picture with given patterns and exploit the symmetry of the face. The used patterns are shown in the Figs. 2 and 3.

The basic idea in Viola Jones algorithm is to find the features associated with each pattern repeated at different scales. All pixels in each region S1, S2, S3, and S4 are summed, and features are calculated as is illustrated in Table 1. Thresholds T1 to T5 will be determined in the training process. The complexity of Viola Jones algorithm is high, despite the using specific techniques to speed up the process (such integral image and Ada-boost algorithm) [7]. Specific assumptions are made

Fig. 3 The symmetry proprieties of face

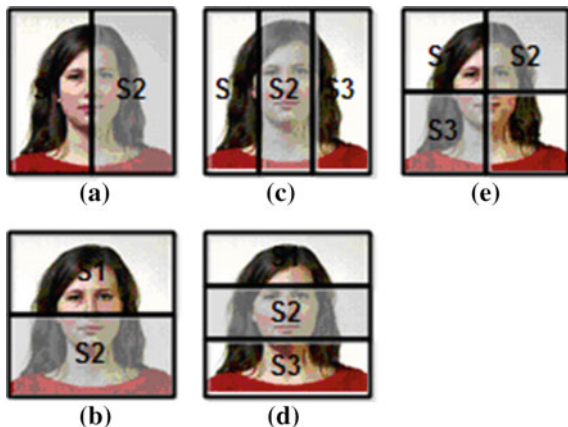


Table 1 The Viola Jones's features and condition

Feature	Condition to be fulfilled
(a) S1-S2	<T1
(b) S1-S2	>T2
(c) S1 + S3-S2	<T3
(d) S1 + S3-S2	>T4
e) S1 + S4-S2-S3	<T5

to make the face detection easier. Images should contain faces with no beard, moustache or glasses, the illumination should be constant and in particular cases the ethnicity may affect the recognition process. The face detection process could be more robust if preliminary processing techniques are involved such as color equalization and edge detection. Color image (usually red-green-blue image or RGB image) is transformed in a binary image and then the lips and eyes are identified by scanning the original image.

The image scanning will estimate the regions where the eyes and lips should be located based on the symmetry properties of human face. Lips are modeled as an ellipse and the eyes are modeled as irregular ellipse [3, 8, and 4] as shown in Fig. 4.

After the region of eyes and lips are determined, the parameters a and b respectively a , b_1 and b_2 are measured using an edge detector, and the training process will run. The training is necessary to determine ellipse parameters in

Fig. 4 The lips and eyes models

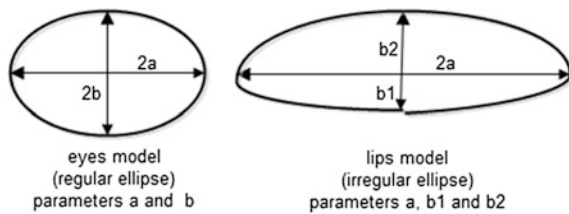


Table 2 Emotion recognition from frontal face image

Detailed operations in emotion recognition from frontal face image
Image preprocessing (color and illumination normalization)
Skin detection
Face segmentation
Eyes and lips segmentation
Estimate parameters of eyes and lips models (a , b) and (a , b_1 , b_2)
Training process (associate the above parameters with a class (emotion))
Classify the emotion using the current face image

particular condition such as neutral, surprise, fear, joy, sadness etc. Based on the training process the emotions are classified. Table 2 shows the detailed operations in emotion recognition from frontal face image.

The image preprocessing involves the normalization in RGB space (R, G, B are the values of the pixel) as:

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \quad (1)$$

The normalized RGB representation ignore ambient and may be invariant to changes of face orientation relatively to the light source.

The skin detection goal is to build a decision rule to distinguish the skin pixel from non-skin pixel. There are many methods to decide if a pixel is skin or non-skin pixel [2]:

Explicitly Rule (based on RGB color space)

Consider that R, G, B , are the values of a pixel it will be classified as skin if the following conditions are fulfilled:

$$\begin{aligned} R > 95 \text{ and } G > 40 \text{ and } B > 20 \\ \max(R, G, B) - \min(R, G, B) > 15 \text{ and} \\ |R - G| > 15 \text{ and } R > G \text{ and } R > B \end{aligned} \quad (2)$$

Nonparametric Skin Distribution Models

These methods estimate skin color distribution without find an explicit model. A number of M lookup tables (LUT) that has N bins each (corresponding to a range of color component) which stores the number of times a particular color occurred in training image is build. The value:

$$P_{skin}(c) = \frac{LUT(c)}{\sum_{k=1}^M \sum_{i=1}^N LUT(k, i)} \quad (3)$$

is the probability that current pixel to be skin pixel. A pixel is classified as skin pixel if $P_{skin}(c) \geq T_{skin}$ where T_{skin} is a given threshold.

Bayes Classifier

The above calculated value is a conditional probability $P(c|skin)$ —probability of observing a color knowing that it represents skin. A probability of observing skin given a particular value of color $P(skin|c)$ could be more suitable for skin detection. This probability is given by Bayes rule:

$$P(skin|c) = \frac{P(c|skin)P(skin)}{P(c|skin)P(skin) + P(c|\sim skin)P(\sim skin)} \quad (4)$$

where $P(c| \sim skin)$ is the probability of observing a color knowing that it is not skin. The probabilities $P(c|skin)$, $P(c| \sim skin)$, $P(skin)$ and $P(\sim skin)$ are computed using the skin and non-skin pixels histograms and the total number of skin and non-skin pixels.

Parametric Skin Distribution Models

The nonparametric models need a large space for storage and the performance depends on the data in the training set. A parametric model is more compact (memory requirements will be lesser) and it has the possibility to generalize the training data. Skin color distribution is modeled by a probability density function (usually Gaussian) as:

$$p(\mathbf{c}|skin) = \frac{1}{2\pi\sigma_s^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{c} - \boldsymbol{\mu}_s)^T \boldsymbol{\sigma}_s^{-1}(\mathbf{c} - \boldsymbol{\mu}_s)\right] \quad (5)$$

where \mathbf{c} is the color vector, $\boldsymbol{\mu}_s$ is mean vector and $\boldsymbol{\sigma}_s$ is covariance matrix. These parameters are estimated using training set: $\boldsymbol{\mu}_s = \frac{1}{n} \sum_{j=1}^n \mathbf{c}_j$ and $\boldsymbol{\sigma}_s = \frac{1}{n-1} \sum_{j=1}^n (\mathbf{c}_j - \boldsymbol{\mu}_s)(\mathbf{c}_j - \boldsymbol{\mu}_s)^T$ where n is the number of color vectors \mathbf{c}_j .

3 Emotion Detection Using Speech Signals

Speech has information on the message and emotions too. To detect emotion one should understand how the speech signal is generated. Voice is produced by varying the vocal tract and the excitation source [9, 10]. During emotion, both components will behave differently comparing with neutral state. There are many features of speech signal that characterize the vocal tract and source excitation. Extracting the most important and robust of these features will lead to emotion recognition. There many kinds of emotions in voice: happiness, surprise, joy, sadness, fear, anger, disgust etc. Systems for emotion detection must capture the voice using a microphone, reduce or remove the background noise. These processing will improve the quality of speech. Then, the speech features are associated with a specific emotion.

The emotion detection based on speech signal analyses features such as:

- pitch
- energy (computing Teager Energy Operator—TEO)
- energy fluctuation
- average level crossing rate (ALCR)
- extrema based signal track length (ESTL)
- linear prediction cepstrum coefficients (LPCC)
- mel frequency cestrum coefficients (MFCC)
- formants
- consonant vowel transition

The pitch in emotional speech is compared with neutral speech and the discriminative power of pitch is measured. Specific emotion, such as disgust, may be detected using semantic and prosodic features. The energy measurement reflects the changes in the airflow structure of speech. The emotion detection uses temporal and spectral properties of speech. These features are due to syllabic, harmonic and formants structure of speech.

Temporal processing such as ALCR and ESTL find the temporal features which differ in amplitude and frequency.

The LPCC, MFCC and formants are related to vocal tract information that is changed with emotion. If the above features will combine, the robustness of the emotion detector will increase.

The following paragraphs describe in more details how these algorithms work [11, 12] and analyze their computational complexity.

The LPCC algorithm estimates the speech parameters by prediction of the current sample as a linear combination of the past samples as shown below:

1. Signal windowing with the Hamming window

$$w(n) = 0.54 - 0.46 \cos\left(2\pi \frac{n}{N}\right), 0 \leq n < N \quad (6)$$

N is the speech window length

2. Pre-filtering with transfer function

$$H_p(z) = 1 - az^{-1} \quad (7)$$

high pass filter with $a = 0.97$

3. Linear prediction: find the linear prediction coefficients (LPC) a_k such that the spectrum of speech signal $S(z)$ to be given by

$$S(z) = E(z) \cdot V(z) \quad (8)$$

where $E(z)$ is the excitation (periodic pulses at pitch period or noise) and

$$V(z) = \frac{G}{1 - \sum_{k=1}^p a_k z^{-k}} \quad (9)$$

with G the gain and p the number of LPC coefficients. Usually, determining of LPC coefficients is done using the Levinson Durbin method, described briefly below:

- (a) Compute the autocorrelation of the speech signal, $s(n)$,

$$R(j) = \sum_{n=0}^{N-1} s(n)s(n+j), j=0 \dots m \quad (10)$$

with m the number of LPC coefficients

(b) Computes

$$A_1 = \begin{bmatrix} 1 \\ a_1 \end{bmatrix} \quad \text{with } a_1 = -\frac{R_1}{R_0} \tag{11}$$

(c) Calculates

$$E_1 = R_0 + R_1 a_1 \tag{12}$$

(d) For $k = 1$ to m

- Computes

$$\lambda = \frac{-\sum_{j=0}^k a_j R_{k+1-j}}{E_k} \tag{13}$$

- Calculates

$$A_{k+1} = \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ a_k \\ 0 \end{bmatrix} + \lambda \begin{bmatrix} 0 \\ a_k \\ \vdots \\ a_2 \\ a_1 \\ 1 \end{bmatrix} \tag{14}$$

- Computes

$$E_{k+1} = (1 - \lambda^2)E_k \tag{15}$$

4. Cepstral analysis: the real cepstrum $s_c(n)$ is the inverse Fourier transform (*IFFT*) of the logarithm of the speech spectrum amplitude:

$$s_c(n) = IFFT [\ln(S(\omega))] \tag{16}$$

The MFCC algorithm determines the Mel coefficients based on audio perception following the steps:

1. Windowing (Hamming window)
2. Pre-emphasis—spectrally flatten the windowed input signal
3. Computes the spectral power of speech window, $|S(\omega)|$ using the fast Fourier transform (FFT)

$$|S(k)| = \sqrt{\text{Re}^2(\text{FFT}[s(n)]) + \text{Im}^2(\text{FFT}(s(n)))} \quad (17)$$

4. Apply Mel filter to the spectral power. The Mel filter comprises a series of overlapped triangular band pass filter banks that maps the powers of the spectrum onto the Mel scale that resembles the way human ear perceives sound. The formula to calculate the Mel filter is:

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)} & f(m-1) \leq k < f(m) \\ \frac{f(m+1)-k}{f(m+1)-f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases} \quad (18)$$

and

$$\sum_{m=0}^{M-1} H_m(k) = 1 \quad (19)$$

M is the number of Mel Filters and $f()$ is the set of $M + 2$ Mel frequencies. The Mel frequencies are calculated using Mel scale

$$\text{Mel}_f = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \text{ and } \text{Mel}_f^{-1} = 700 (10^{f/2595} - 1) \quad (20)$$

as:

$$f(m) = \left(\frac{N}{F_s}\right) \text{Mel}^{-1}\left(\text{Mel}(f_i) + m \frac{\text{Mel}(f_h) - \text{Mel}(f_i)}{M+1}\right) \quad (21)$$

where Mel_f is Mel frequency and f is the frequency, N is the window length, F_s is sampling frequency, f_i and f_h are the lowest and the highest frequency in the Mel filter bank.

The output of the Mel filter bank is:

$$Y(m) = \sum_{k=0}^{N-1} |S(k)|^2 H_m(k), \quad 0 \leq m < M \quad (22)$$

Before next step the logarithmic spectral power is calculated as:

$$Y^*(m) = \log_{10} Y(m), \quad 0 \leq m < M \quad (23)$$

5. Apply inverse discrete cosine transform (IDCT) to Mel banks filter outputs to return in the time domain. A reduction in computational effort is achieved because discrete cosine transform (DCT) accumulates most of the information

contained in the signal to its lower order coefficients. For K coefficients the Mel cepstrum coefficients are:

$$c_m(k) = \sum_{m=0}^{M-1} Y^*(m) \cos(\pi.k(m+1/2)/M) , 0 \leq k < K \tag{24}$$

The first two steps are the same with LPCC algorithm.

4 The Classifier Process

A generic classifier process [13, 14] is illustrated in Fig. 5.

An input \mathbf{x} is analyzed using a set of discrimination functions $g_i: R^n \rightarrow R, i = 1..c$ each giving a score corresponding to a class from 1 to c . The input is then classified in the class with the highest score. Classifier is characterized by a classification error calculated as $\frac{\text{number of misclassification}}{\text{total number of inputs}}$ and classification accuracy given by $1 - \frac{\text{number of misclassification}}{\text{total number of inputs}}$.

The classifier is building in the training phase using as much as possible input combinations to configure the parameters of discrimination functions. The classifier may be designed using an input set \mathbf{S} and tested on the set $\mathbf{R} \subseteq \mathbf{S}$. If \mathbf{R} is close to \mathbf{S} , the classifier's performance will increase. The main alternatives are: re-substitution or R-method ($\mathbf{R}=\mathbf{S}$), hold-out or H method (\mathbf{R} is half of \mathbf{S}), cross-validation or rotation method (divide the \mathbf{S} in K subsets of same size, use $K-1$ subsets to train and the remaining subset to test; repeats this procedure K times and average the K estimates).

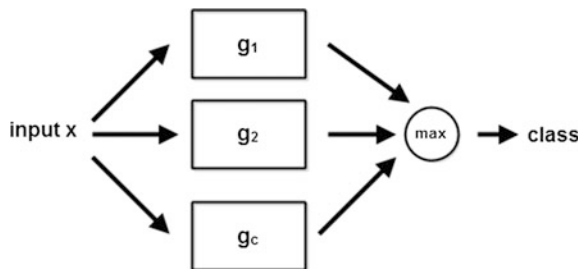
Most known classifier algorithms are summarized below [15, 16 and 17].

The k-Nearest Neighbor Classifier (kNN)

Given a set X and a distance function, find the k closest points in X to the training set. The similarities (inverse of the distances) of the neighbors in the same class are summed. The class with the highest score is assigned to the input set X .

We consider a set of p classes $C = \{c_1, c_2, \dots, c_p\}$ and a training set $X = \{(x_i, y_i) \mid x_i \in {}_1R^m, y_i \in C, i = 1 \dots n\}$, where the point (x_i, y_i) is a

Fig. 5 The classifier process



m-dimensional x_i value classified in class y_i from p classes. For a given input x the distances d_i to each point in the training set are computed as $d_i = \|x_i - x\|$, and the first k distances (sorted in ascending order) are taken. The chosen class for the input x is the class of the majority of the selected k points. The kNN search technique is widely used in various areas such as image processing, multimedia database and classification. In general, nearest neighbor classifiers typically have good predictive accuracy in low dimensions, but might not in high dimensions.

Support Vector Machine (SVM)

The SVM algorithm finds the decision surface (hyperplane) that maximizes the margin between the data points of the two classes.

A set $X = \{(x_i, y_i) \mid x_i \in \mathcal{R}^m, y_i \in \{-1, 1\}, i = 1 \dots n\}$ where the point (x_i, y_i) is a m-dimensional x_i value classified in class y_i . A hyperplane that separating the m-dimensional x_i values into the two classes is given by relation: $\mathbf{w}\mathbf{x} = b$. The optimization problem minimize $\|\mathbf{w}\|$ in the hyperplane $\mathbf{w}\mathbf{x} = b$ with constrain $y_i(\mathbf{w}\mathbf{x}_i - b) \geq 1, i = 1 \dots n$.

The SVM is a binary classifier. For a problem with over two classes, we need to create several SVM classifiers. The number of the classifiers is equal with the number of classes and the algorithm is applied one class versus remained class. Such multiclass SVM algorithm has better performance than a kNN algorithm.

Decision Tree

A decision tree is a procedure for classifying data based on their attributes. Based on the features of the input, a binary tree is build. The input is classified in classes after tree traversal (leafs of the decision tree are the possible classes).

Construction of decision tree no needs any settings, so exploratory knowledge discovery will use such trees. Given a data set \mathbf{D} and an attribute \mathbf{A} having k values, the decision tree is building according to the following steps:

- (a) The data set \mathbf{D} is splitting in k subsets (or class) \mathbf{D}_j
- (b) Calculate the information gain $G(\mathbf{D}, \mathbf{A})$ associated with the attribute \mathbf{A} :

$$G(\mathbf{D}, \mathbf{A}) = E(\mathbf{D}) - \sum_{j=1}^k \frac{\text{card}(\mathbf{D}_j)}{\text{card}(\mathbf{D})} E(\mathbf{D}_j) \quad (25)$$

where $E(\cdot)$ is the entropy calculated as

$$E(x) = \sum_{i=1}^k -p_i \log_2(p_i) \quad (26)$$

The parameter p_i is the probability that x to belong to class k and $\text{card}(y)$ is the number of elements in the set y

- (c) Choose the attribute with the highest gain. This attribute becomes a splitting attribute for the current node in the binary tree
- (d) The step (b) is repeated by choosing the next attribute in ascending order of gains

5 Digital Signal Processors (DSP) Architectures

A digital signal processor must have high-performance, power-efficient architecture that may be used in real time applications with high level of computational complexity. The following features characterize a DSP architecture: fast and flexible arithmetic, large dynamic range for arithmetic computation, the ability of loading two operands in one processor cycle, hardware circular memory addressing and hardware control of iterations. Most emotion detection algorithms (described in the previous sections) need a large dynamic scale for number representation. The processor arithmetic is fixed-point arithmetic or floating-point arithmetic. Floating-point processors are easier to program than fixed-point processors, because it is not needed to adapt the algorithm to conform to a fixed-point representation or emulate the floating-point operations using fixed-point instructions, but they consume more power [18]. Emotion detection algorithms are complex and they should be implemented with no changes, thus flexible computational units are required.

Computational power of DSP processors is determined by the instruction set and the mechanisms which increase the parallelism (hardware loops, computational units that may work simultaneously, DMA—direct memory access channels, large dynamic range). Organization of a microcontroller's memory sub-system has a great impact on its performance. The DSP processors have a modified Harvard architecture with separate memories for the program and data. Memories are large enough to accommodate the data and program and they are accessed through DMA channels. The emotion detection algorithms may need large data memory (especially those algorithms based on the image processing). Memories' capacity is up to thousands of Mbytes. These amounts are enough for instructions codes due to the processor's architecture with a high level of parallelism, a powerful instruction set and a high density code. For data memory, the above amounts are necessary to store image, speech signals and data training set. The parallelism is achieved using a multi-core architecture with at least two processor cores with two or three computational units each that can run in parallel. The specialized blocks, in program sequencer, allow hardware control of circular addressing and iterations, and will increase the level of parallelism. A digital signal processor has specialized arithmetic units (arithmetic and logic unit-ALU, multiply and accumulate unit-MAC and shifter unit-SHIFTER). A rich set of peripherals (such as serial and parallel ports controlled by DMA channels, timers) used to communicate with external devices as digital camera, microphones and various sensors must complete a powerful DSP architecture. These goals can be achieved by a multi-core processor with joint architectures such as DSP (Digital Signal Processing) and ARM (Acorn RISC Machine), that delivers peak performance up to 24 Giga floating-point operations per second.

This section presents two architectures that follow the above requirements: Blackfin dual core processors [19] and super Harvard dual core (SHARC) processors from Analog Devices [20, 21] as representative architectures of fixed-point and floating-point microcontrollers' architectures, respectively.

Blackfin BF561 Microcomputers

The core of Blackfin microcomputer is a 16-bit fixed-point processor based on the unified architecture Micro Signal Architecture (MSA), developed by Analog Devices and Intel to carry out applications with complex computations and power-sensitive. An embedded fast Ethernet controller that offers the ability to connect directly to a network, a high peripheral support and memory management is provided in Blackfin architecture. The clock rate and operating voltages can be switched dynamically for specified tasks via software for reduction power, between 350 MHz at 1.6 V. and 750 MHz at 1.45 V.

The main features of Blackfin core are:

- dual multiply-accumulate (MAC) units
- an orthogonal reduced instruction set computer (RISC) instruction set
- single-instruction multiple data (SIMD) programming capabilities
- multimedia processing capabilities

The Blackfin processor uses a modified Harvard architecture which allows multiple memory accesses per clock cycle, multifunction instructions, and control vector operations. There are several computational units: the multiplier and accumulate MAC unit, the arithmetic unit ALU that supports SIMD operations and has Load/Store architecture, the video ALU that offers parallel computational power for video operations (quad 8-bit additions and subtracting, quad 8-bit average, Subtract-Absolute-Accumulate), the addressing unit with dual-data fetches in a single-instruction. The program sequencer controls program flow (subroutines, jumps, idles, interrupts, exceptions, hardware loops). The Blackfin processors have “pipe line” architecture which allocates best the workload among the processor’s units, which leads in efficient parallel processing. A single hierarchical memory space that is shared between data and instruction memory is used in Blackfin architecture. Figures 6 and 7 illustrate the general architecture of Blackfin processors family and the Blackfin core.

The Multi-core Architecture ARM SHARC

Higher performance may be achieved using a multi-core architecture. These architectures are systems on-chip (SoC) or multiprocessors systems that include three interconnected processors (one general purpose and two DSPs). As an example, the ARM SHARC architecture ADSP-SC58x is shown in Fig. 8.

ADSP-SC58x architecture includes two SHARC + cores and one ARM Cortex-A5 core. SHARC + cores share a crossbar switch with a shared cache memory and interfaces with internal and external memory. The switch provides access to an array of peripheral I/O, including USB. Two high-speed link ports allow multiple systems on-chip or DSP processors to be connected. Communication links include serial ports, analog to digital convertors. The ARM Cortex-A5 will handle secured communication between SHARC + cores. Security cryptographic engines (AES-128 and AES-256 standards) are involved for secure boot and network security support. Operating systems as Linux and Micrium μ C/OS-III are

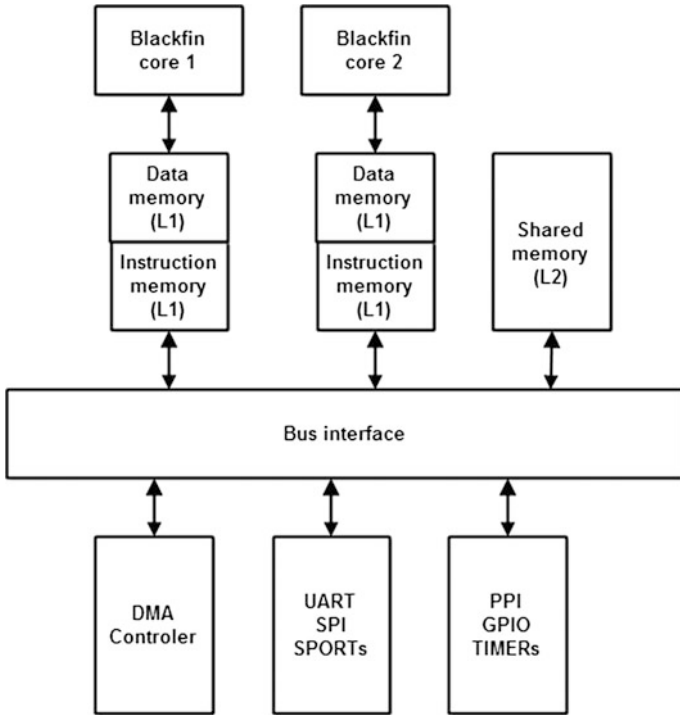


Fig. 6 The Blackfin BF561 general architecture

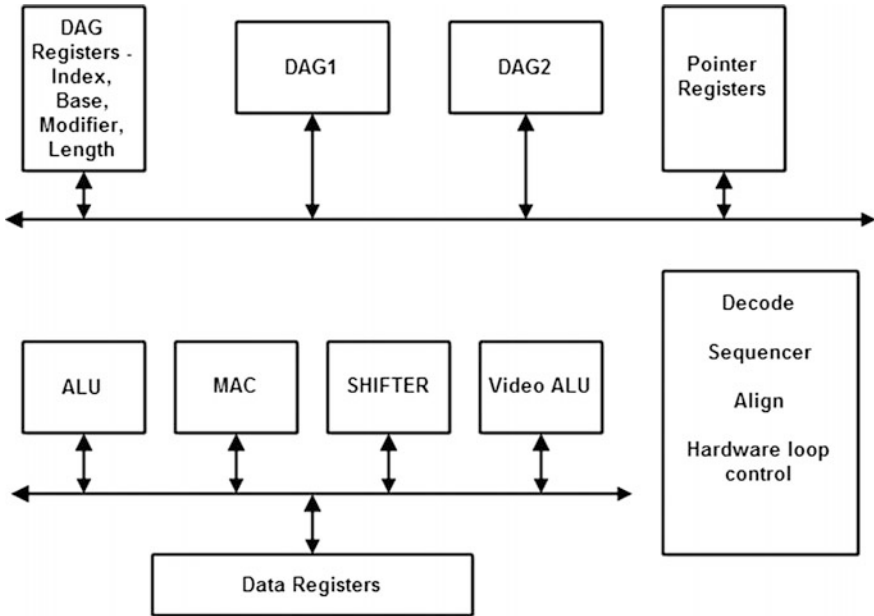


Fig. 7 The Blackfin core

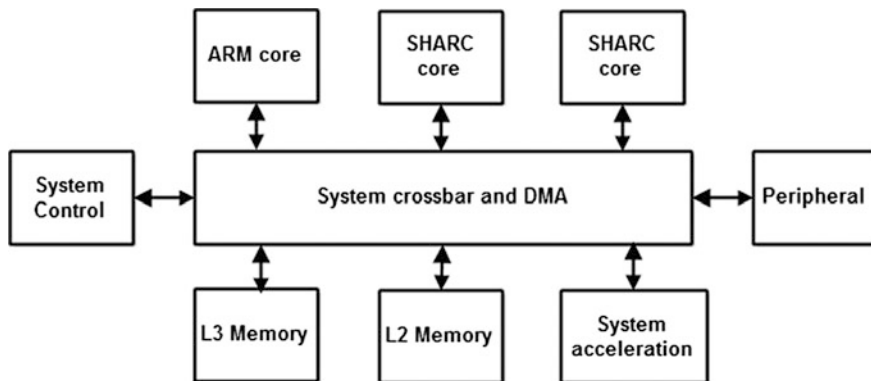


Fig. 8 The multi-core ARM SHARC general architecture (ADSP-SC58x)

available for the ARM Cortex-A5 and/or SHARC + cores. System services and device drivers are provided.

The SHARC processor integrates a SHARC + SIMD core, level one (L1) memory that run at the full processor speed, crossbar, instruction and data cache memory and communication ports. It has a modified Harvard architecture with a hierarchical memory structure. The SHARC + core has two computational processing elements that work as a SIMD engine. Each processing elements contain an arithmetic/logic unit (ALU), multiplier, shifter and register file. In SIMD mode, the processing elements execute the same instruction but they use different data. The architecture is efficient for math-intensive DSP algorithms. The SIMD mode doubles the bandwidth between memory and the processing elements. Two data values are transferred between memory and registers. There is a set of pipelined computational units in each processing element: ALU, multiplier, and shifter. These units support 32-bit single-precision floating-point, 40-bit extended precision floating-point, 64-bit double-precision floating-point and 32-bit fixed-point data formats, and may run in parallel, maximizing computational power. In SIMD mode, multifunction instructions enable the parallel ALU and multiplier operations execution in both processing elements per core, in one processor cycle for fixed-point data and most six processor cycles for floating-point data, depending on data format. The instruction's execution involves an interlocked pipeline and data dependency check. A general purpose data registers exists in each processing element for transferring data between the computational units and allowing unconstrained data flow between computation units and internal memory. Most units have secondary registers that can be used for a fast context switch. As in Blackfin processors, a hardware loop control is provided. The instruction is coded on 48 bits and accommodates various parallel operations (multiply, add, and subtract in both processing elements while branching and fetching up to four 32-bit values from memory). A new instruction set architecture, named Variable Instruction Set Architecture (VISA), drops redundant/unused bits within the 48-bit instruction to

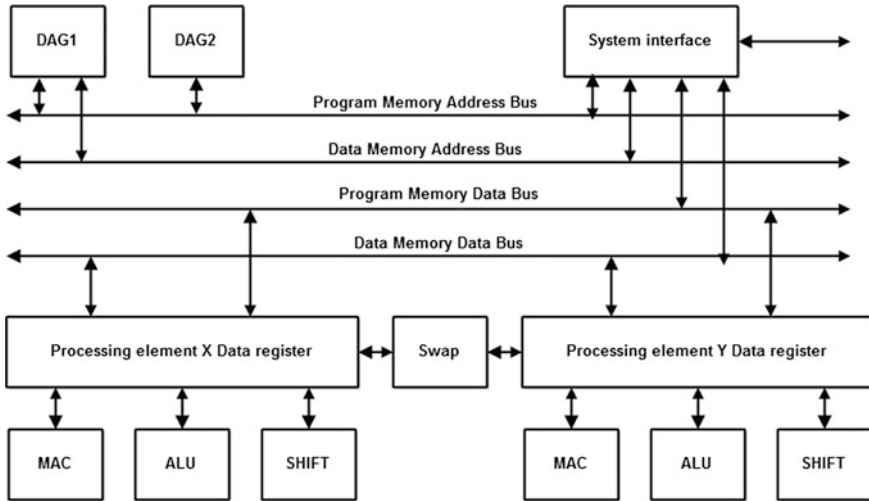


Fig. 9 The SHARC general architecture

create more compact code. With its separate program and data memory buses (Harvard architecture) and on-chip instruction conflict-cache, the processors can simultaneously fetch four operands (two over each data bus) and one instruction (from the conflict-cache bus arbiter), all in a single cycle. The program sequencer supports efficient branching (with low delays) for conditional and unconditional instructions using a hardware branch predictor based on a branch target buffer (BTB). The general architecture of SHARC + core is illustrated in the Fig. 9.

ARM Cortex-A5

The ARM Cortex-A5 is a low-power and high-performance processor based on ARMv7 architecture with full virtual memory capabilities. This processor runs 32-bit ARM instructions, 16-bit and 32-bit Thumb instructions, and 8-bit Java codes. This architecture is illustrated in Fig. 10.

The main functional units in ARM Cortex A5 are:

- floating-point unit (integrated with processor pipe line, optimized for scalar operation and enabled to emulate vector operation)
- media processing engine (with instruction for audio, video, 3D graphics processing)
- memory management unit (separate data and program memories in a Harvard architecture)
- cache controller (improving performance for cache memory on level two in the hierarchical memory sub-system)
- in order pipe line with dynamic branch prediction
- intelligent energy manager

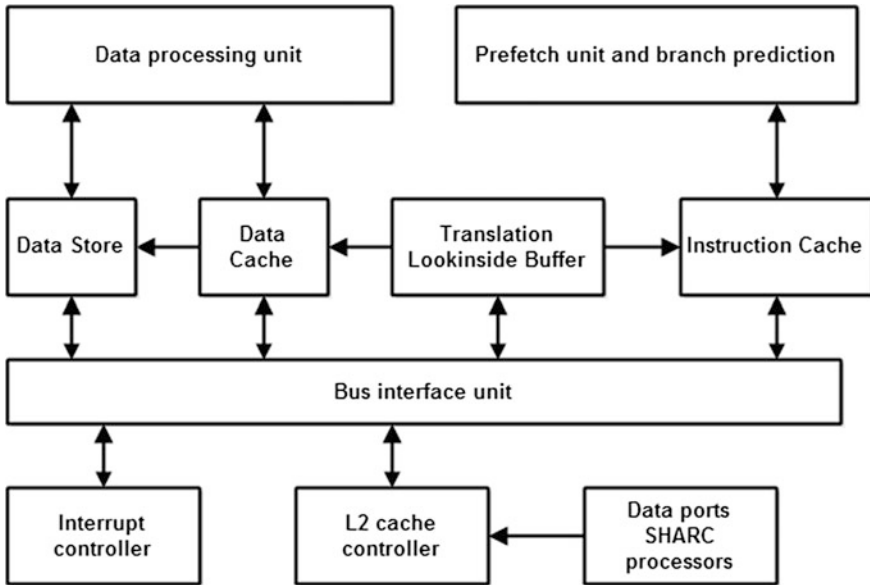


Fig. 10 The core ARM Cortex A5 architecture

Data Processing Unit (DPU) has general purpose registers, status registers and control registers. This unit decodes and executes the instructions. Instructions using data in the registers are fed from the Pre-Fetch Unit (PFU). Load and store instructions that need data to or from the memory are managed by the Data Cache Unit (DCU). The PFU extracts instructions from the instruction cache or from external memory and predicts the branches in the instruction flow, before it transmits the instructions to the DPU for processing. DCU has a L1 data cache controller, a pipe line with load/store architecture, a system coprocessor which performs cache maintenance operations both for data and instruction cache. Bus Interface Unit (BIU) has the external interfaces and provides a single 64-bit port, shared between the instruction side and the data side. The ARM Cortex-A5 media processing engine features are:

- SIMD and scalar single-precision floating-point computation
- Scalar double-precision floating-point computation
- SIMD and scalar half-precision floating-point conversion
- SIMD 8, 16, 32, and 64-bit signed and unsigned integer computation
- 8 or 16-bit polynomial computation for single-bit coefficients
- Structured data load capabilities
- Large, shared register file, addressable as 32-bit, 64-bit and 128-bit registers

The instruction set include: addition and subtraction, multiplication with optional accumulation, maximum or minimum value, inverse square-root,

comprehensive data-structure load (e.g. register-bank-resident table lookup implementation). The floating-point unit is characterized by:

- Single-precision and double-precision floating-point formats
- Conversion between half-precision and single-precision
- Fused Multiply-Accumulate operations
- Normalized and de-normalized data handled in hardware

The memory management unit controls the L1 and L2 memory system and translates virtual addresses to physical addresses and accesses the external memory. The Translate Look-inside Buffer operations are managed by a coprocessor integrated with the core that provides a mechanism for configuring the memory system. An AXI (Advanced eXtensible Interface) provides a high bandwidth for data transfers to L2 caches, on-chip RAM, peripherals, and external memory. It comprises a single AXI port with a 64-bit for instruction and data.

The computational power, memory and peripheral interfaces should be tested to see if they are large enough for implementing complex image and speech processing algorithms such emotion detection.

6 The Experimental Results

This section presents the main results that prove the possibility of implementation the emotion detection algorithms. We testes simplified but effective algorithms using Visual DSP++ IDE, in C language optimized for speed with code profiling enabled. For each algorithm we will measure the execution time for various image sizes and algorithm specific parameters. The microcontroller used was Blackfin BF561 that operates at 750 MHz clock frequency with one core active. Programs performance will be better if both cores in BF561 would be enabled or a multi-core chip is used. Memory amount (between 16 to 128 MB) is large enough to store medium size image and the rich set of input-output peripherals (such as parallel port interface—PPI and universal serial bus—USB) ensures a proper communication between processor and external devices.

The mouth and eyes detect algorithm

The following algorithm was implemented to detect the mouth and eyes (Fig. 11).

The discriminators for mouth and eyes are presented bellow.

Mouth (lips) discrimination

- compute

$$l(r) = -0.776r^2 + 0.560r + 0.2123 \quad (27)$$

```

Read image
For the whole read image do
    Apply skin filter
    Create binary image
    Scan binary image to determine the face region

For face region do
    Apply mouth discriminator
    Determine mouth region
    Measure the mouth irregular ellipse parameters
    Apply eyes histogram
    Determine eyes regions
    Measure the eyes regular ellipse parameters

Apply classifier or update data base in training phase
Determine the type of emotion

```

Fig. 11 Mouth and eyes detect algorithm

- compute

$$f(r) = -0.776r^2 + 0.560r + 0.1766 \quad (28)$$

- the pixel is in mouth region if

$$f(r) \leq g \leq l(r) \text{ and } R \geq 20 \text{ and } G \geq 20 \text{ and } B \geq 20 \quad (29)$$

Eyes discrimination

- convert RGB image in a grayscale image using the formula

$$Y = 0.2989R + 0.587 + 0.114B \quad (30)$$

- apply histogram equalization to grayscale image accordingly with the algorithm bellow:

- (a) compute the probability that a pixel x to have value

$$n \in \{0, 1, \dots, L-1\}: p_n = \frac{\text{number of pixels with value } n}{\text{total number of pixels}} \quad (31)$$

(b) transform the pixels with value k in pixels with value

$$T(k) = \text{round}\left((L - 1) \sum_{n=0}^{L-1} p_n\right) \tag{32}$$

- determine eyes region applying a threshold operation on the histogram equalized image

The results are illustrated in Fig. 12.

Measurements of mouth and eyes parameters are made using the binary images illustrated in Fig. 12. Binary images are scanned horizontally and vertically to find the transitions from black pixels to white pixels and vice versa. Differences between pixel positions corresponding to these two transitions represent the parameters of the ellipses that characterized the mouth and the eyes. Percentage differences between mouth and eyes parameters, determined from neutral image (that is, no emotions) and from image associated to a specific emotion, are calculated. The emotion is determining after measurement of the parameters illustrated in the previous section, as in Table 3.

The “fear”, “happy” and “surprise” are detected easier using the changes direction of b, b_1, b_2 relative to their values in the “neutral” state. For “sad”, “angry” and “dislike” the average differences is computed to discriminate between “sad” and “angry” or “dislike”. The discrimination between “angry” and “dislike” are

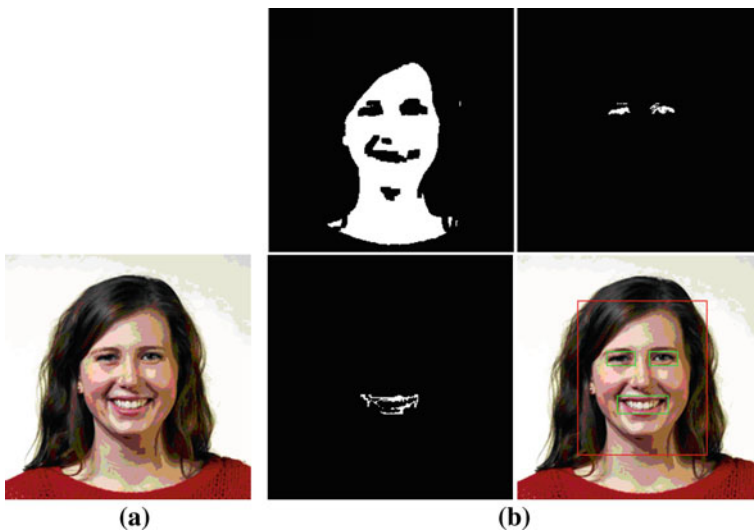


Fig. 12 a Original image b Binary image after skin detection, binary image after eyes detection, binary image after mouth detection and face, eyes and mouth regions determination (from top corner left to right)

Table 3 The parameters variation used in emotion detection

	Increase (relative to neutral)	Unchanged (relative to neutral)	Decrease (relative to neutral)	Average variation of b, b_1, b_2 (relative to neutral)
Fear		b_2	b, b_1	
Happy	b_2		b, b_1	
Sad			b, b_1, b_2	<25 %
Angry			b, b_1, b_2	>25 % and b_1 variation is the greatest
Dislike			b, b_1, b_2	>25 % and b_1 variation is the smallest
Surprise	b_1, b_2		b	

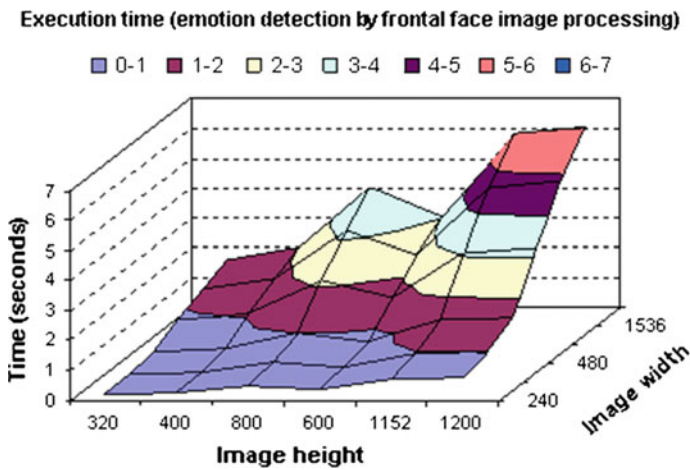


Fig. 13 Emotion detection by face image processing execution time for various image sizes

made using the variation of parameter b_1 . The execution time is illustrated in Fig. 13.

One can see that for medium images sizes the image processing can be done in seconds.

The code uses the rules presented in [18] to optimize for speed: hardware loops, multifunction instructions and video ALU instructions.

We analyzed the MFCC algorithm for emotion detection from speech signals. This algorithm is robust and has better performance than LPCC algorithm [22].

The speech processing takes about 10 μ s (optimized code) for a speech window length of 256 samples, as is shown in Table 4.

The squared error between mean MFCC coefficients for the neutral state and emotion state is illustrated in the Fig. 14.

Table 4 Execution time (emotion detection from speech signals)

Algorithm step	Time (μ s)
Windowing	0.295111
Pre-filtering	0.003222
Spectral power	4.90889
Mel filter	2.018222
IDCT	2.926222
TOTAL	10.15167

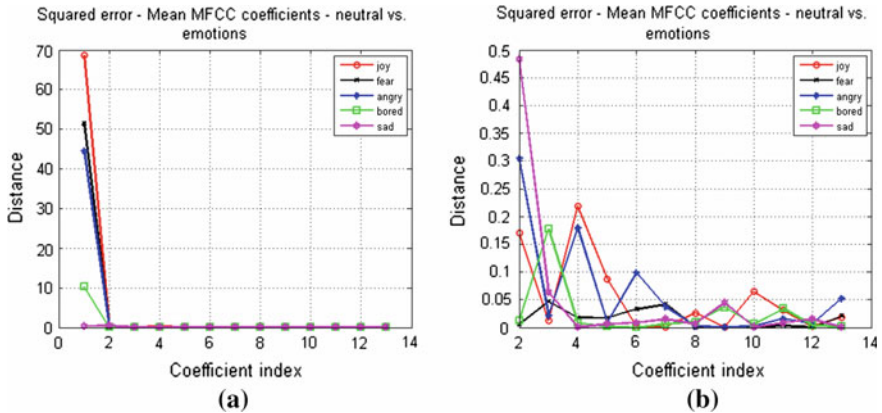


Fig. 14 The mean MFCC coefficients

One can see that the error is greater for the first coefficient (except the “sad” emotion). Discrimination between emotions may use only the first two or three coefficients. Emotions such as “joy”, “fear”, “angry”, and “sad” are discriminated using the first coefficient.

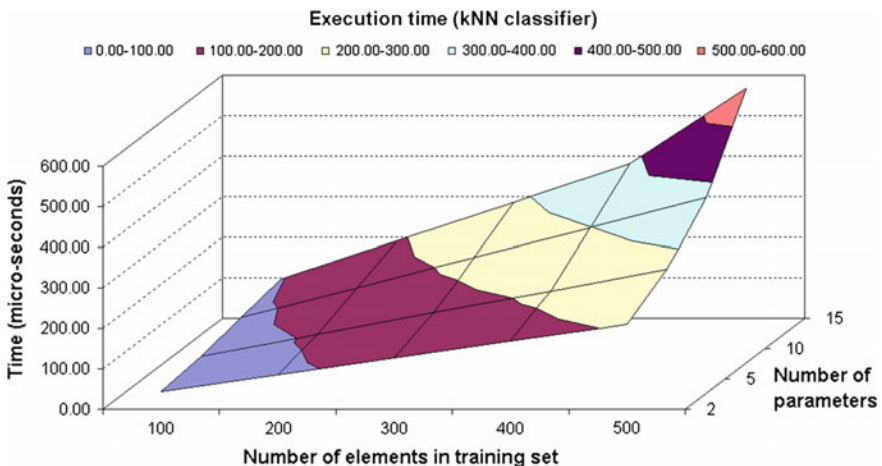


Fig. 15 The kNN classifier’s execution time

The kNN classifier can be implemented in hundreds of microseconds (as it is shown in Fig. 15).

The kNN classifier has been tested for 20 neighbors, 6 classes and up to 15 parameters.

7 Conclusions

This chapter presents an overview of the methods to emotion detection by image and speech processing and it focuses on implementing of such algorithms using digital signal processors. Emotion detection implies frontal face image and/or speech processing and should be implemented in real time. These processing algorithms include face detection, region of interest determining (e.g. mouth, eyes), and spectral analysis (for speech signals). After extracting the relevant parameters, a classifier process is completed to determine the emotion. This process calculates the distances between current parameters, corresponding to the emotion, and the parameters in a training set, corresponding to the possible classes, and determines the most probable class for the current emotion. The main goal of the chapter was to investigate the algorithms for emotion detection and to see if those algorithms may be realized in real time using DSP processors. The results show the possibility to implement most of existing emotion detection algorithms in real time using optimizations methods. For more complex algorithms, the data processing may be performed on a server with more computational power. In this case, the lightweight Internet protocol (LwIP) [18] will transfer the needed data and the results to the mobile device. The chapter shows that this approach is not necessary if analyzed data (that is, the frontal face image size) is not too large (e.g. images with medium size), and the chosen image and speech processing algorithms are not so complex, but still effective. The analyzed processors has enough processing power to implement emotion detection algorithms with reasonable increasing of their basic computational effort for voice and data communication, operating system functions and basic input-output operations. Future work will analyze methods to increase the accuracy of emotion detection from 70–80 % to over 90 % using multi-core DSP processors that can implement more advanced image processing algorithms and classifiers such SVM with operating system support [18, 23] (e.g. Visual DSP Kernel—VDK operating system or Micrium operating system).

The emotion detection algorithms should be integrated in the multimedia services using diverse computing devices interconnected via networking technologies including context-aware media networks [24], therefore new networks architectures should be design for optimal selection of multimedia content delivery [25].

References

1. Klaus R. Scherer, "What are emotions? And how can they be measured?", *Social Science Information & 2005 SAGE Publications* (London, Thousand Oaks, CA and New Delhi), 0539–0184, doi:[10.1177/0539018405058216](https://doi.org/10.1177/0539018405058216) Vol 44(4), pp. 695–729; 058216.
2. Vinay Kumar, Arpit Agarwal, Kanika Mittal, "Tutorial: Introduction to Emotion Recognition for Digital Images", *Technical Report 2011*, inria-00561918.
3. A. Habibizad Navin, Mir Kamal Mirnia, "A New Algorithm to Classify Face Emotions through Eye and Lip Features by Using Particle Swarm Optimization", 2012 4th International Conference on Computer Modeling and Simulation (ICCMS 2012) IPCSIT vol.22 (2012) © (2012) IACSIT Press, Singapore, pp. 268–274.
4. M. Karthigayan, M. Rizon, R. Nagarajan and Sazali Yaacob, "Genetic Algorithm and Neural Network for Face Emotion Recognition", pp. 57–68, book chapter in "Affective Computing", Edited by Jimmy Or, Intech, 2008, ISBN 978-3-902613-23-3.
5. Moon Hwan Kim, Young Hoon Joo, and Jin Bae Park, "Emotion Detection Algorithm Using Frontal Face Image", 2015 International Conference on Control Automation and Systems (ICCAS2005) June 2-5, 2005, Kintex, Gyeong Gi, Korea, pp. 2373–2378.
6. Paul Viola, Michael J. Jones, "Real-Time Face Detection", *International Journal of Computer Vision*, 57(2), 2004, Kluwer Academic Publishers, pp. 137–154.
7. Yi-Qing Wang, "An Analysis of the Viola-Jones Face Detection Algorithm", *Image Processing on Line*, 2014, doi:[10.5201/ipol.2014.104](https://doi.org/10.5201/ipol.2014.104), pp. 129–148.
8. Cheng-Chin Chiang, Wen-Kai Tai, Mau-Tsuen Yang, Yi-Ting Huang, Chi-Jaung Huang, "A novel method for detecting lips, eyes and faces in real time", *Real-Time Imaging*, 2003, pp 277–287.
9. Rahul. B. Lanjewar, D. S. Chaudhari, "Speech Emotion Recognition: A Review", *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* ISSN: 2278-3075, Volume-2, Issue-4, March 2013, pp. 68–71.
10. Dipti D. Joshi, M. B. Zalte, "Speech Emotion Recognition: A Review", *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*, ISSN: 2278-2834, ISBN: 2278-8735. Volume 4, Issue 4 (Jan.–Feb. 2013), pp. 34–37.
11. J. Přibil, A. Přibilová, "An Experiment with Evaluation of Emotional Speech Conversion by Spectrograms", *Measurement Science Review*, Volume 10, No. 3, 2010, pp. 72–77.
12. Taabish Gulzar, Anand Singh, "Comparative Analysis of LPCC, MFCC and BFCC for the Recognition of Hindi Words using Artificial Neural Networks", *International Journal of Computer Applications* (0975–8887) Volume 101– No.12, September 2014, pp. 22–27.
13. Vaishali M. Chavan, V.V. Gohokar, "Speech Emotion Recognition by using SVM-Classifer", *International Journal of Engineering and Advanced Technology (IJEAT)* ISSN: 2249-8958, Volume-1, Issue-5, June 2012, pp. 11–15.
14. Ludmila I. Kuncheva, "Combining Pattern Classifiers: Methods and Algorithms", Ed. John Wiley & Sons, Hoboken, New Jersey, 2004.
15. Xindong Wu et al., "Top 10 algorithms in data mining", *Knowl Inf Syst* (2008) 14:1–37, doi:[10.1007/s10115-007-0114-2](https://doi.org/10.1007/s10115-007-0114-2), Springer-Verlag London Limited 2007.
16. Xuchun Li, Lei Wang, Eric Sung, "AdaBoost with SVM-based component classifiers", *Engineering Applications of Artificial Intelligence*, vol. 21 (2008), pp. 785–795.
17. Niklas Lavesson, "Evaluation and Analysis of Supervised Learning Algorithms and Classifiers", Publisher: Blekinge Institute of Technology, Printed by Kaserstryckeriet, Karlskrona, Sweden 2006, ISBN 91-7295-083-8.
18. Sorin Zoican, "Embedded Systems and Applications in Telecommunications", chapter book in "Embedded Systems and Wireless Technology: Theory and practical applications", edited by: Raúl Aquino Santos, Arthur Edwards Block Science Publishers, 2012.
19. Analog Devices, BF561 - Blackfin Embedded Symmetric Multiprocessor, Data sheet 2009.
20. Analog Devices, ADSP-SC58x/ADSP 2158x - SHARC + Dual Core DSP with ARM Cortex-A5, Data sheet 2015.

21. ARM, Cortex-A5 - Technical Reference Manual, 2010.
22. Siddhant C. Joshi, A.N. Cheeran, "MATLAB Based Feature Extraction Using Mel Frequency Cepstrum Coefficients for Automatic Speech Recognition", International Journal of Science, Engineering and Technology Research (IJSETR), Volume 3, Issue 6, June 2014 1820, ISSN: 2278-7798, pp. 182–1823.
23. Sorin Zoican, "Networking Applications for Embedded Systems", pp. 1–20, chapter book in "Real Time Systems, Architecture, Scheduling, and Application", Ed. INTECH, ISBN 978-953-51-0510-7, edited by: Seyed Morteza Babamir, 2012.
24. Soraya Ait Chellouche, Daniel Negru, Julien Arnaud, Jordi Mongay Batalla, "Context-aware multimedia services provisioning in future Internet using ontology and rules", 2014 International Conference and Workshop on the Network of the Future, NOF 2014, Paris, France, December 3-5, 2014. IEEE 2014, ISBN 978-1-4799-7531-0 pp. 1–5, <http://dblp.dagstuhl.de/db/conf/nof/nof2014.html>.
25. Y. Kryftis, C. X. Mavromoustakis, G. Mastorakis, E. Pallis, J. Mongay Batalla, J. P. C. Rodrigues, C. Dobre, G. Kormentzas, "Resource Usage Prediction Algorithms for Optimal Selection of Multimedia Content Delivery Methods", IEEE International Conference on Communications ICC, London, 8-12 June 2015, doi:10.1109/ICC.2015.7249263, pp. 5903–5906,

Recognizing Driving Behaviour Using Smartphones

Prokopis Vavouranakis, Spyros Panagiotakis, George Mastorakis, Constandinos X. Mavromoustakis and Jordi Mongay Batalla

Abstract In this chapter at first we will present our methodology for recognizing driving patterns using smartphones, and then we will present in detail the android-based application we have developed to this end, which can monitor driving behavior. The latter can be achieved either using data only from the accelerometer sensor or using a sensor fusion method, which combines data from the accelerometer, the gyroscope and the magnetometer. We can recognize events like hard acceleration, safe acceleration, sharp left turn, safe right turn, sharp left lane change, etc. The application to improve the driving behavior of the driver, displays some hint messages to him after each bad-driving event. All the data from the trips (e.g., driving events that take place during a trip), are stored in a database and the driver has the opportunity to review and analyze them whenever he wants. We believe that organizing drivers in some form of a social network and involving them in a game-like procedure for promoting and rewarding the best driver among them, can motivate drivers to more secure driving customs.

Keywords Driving behavior • Smartphone • Accelerometer • Sensor fusion • Android

P. Vavouranakis · S. Panagiotakis (✉) · G. Mastorakis
Department of Informatics Engineering, Technological Educational Institute of Crete,
71004 Heraklion, Crete, Greece
e-mail: spanag@teicrete.gr

P. Vavouranakis
e-mail: akisvavour@gmail.com

G. Mastorakis
e-mail: mastorakis@gmail.com

C.X. Mavromoustakis
Department of Computer Science, University of Nicosia, Nicosia, Cyprus
e-mail: mavromoustakis.c@unic.ac.cy

J.M. Batalla
Warsaw University of Technology, Nowowiejska Str. 15/19, Warsaw, Poland
e-mail: jordim@interfree.it

1 Introduction

Driver behavior monitoring has evolved tremendously in recent years. Monitoring the drivers' behavior, recording their driving events (safe and aggressive) and giving feedback for the recorded events can enhance driver safety. Also the method of monitoring drivers' behavior using the inbuilt sensors of smartphone devices has been evolving as a new trend because of its less cost and considering the fact that many people already own such devices.

In this chapter at first we will review the most popular algorithms for recognizing driving patterns using smartphones, and then we will present in detail the methodology we applied to this end. Finally, we demonstrate the android-based application we have developed accordingly, which can recognize and monitor driving behavior.

1.1 Smartphone Hardware Sensors

Sensors made the smartphones smart. Sensor is a converter that measures a physical quantity and converts it into a signal, which can be read by an observer or by an instrument. There are various types of sensors, which are currently being used in analyzing driver behavior. These sensors are:

- (a) **Accelerometer.** An accelerometer [1] is an electromechanical device that will measure acceleration forces. These forces may be static (z axis), like the constant force of gravity pulling at your feet, or they could be dynamic (x, y axis) caused by moving or vibrating the accelerometer.
- (b) **GPS.** GPS [2] is a satellite based Navigation tracking, often with a map showing where you have been. It gives us the value of longitude and latitude, which determines the point of location on earth.
- (c) **Gyroscope.** Gyroscope [3] detects the current orientation of the device, or changes in the orientation of the device. Orientation can be computed from the angular rate that is detected by the gyroscope. It basically works on the principle of angular momentum and it is expressed in rad/s on 3-axis.
- (d) **Camera.**
- (e) **Microphone.**
- (f) **Magnetometer.** Magnetometers [4] are measurements instruments used for two general purposes-to measure the magnetization of a magnetic material like a Ferro magnet, or to measure the magnetic strength and the direction of the magnetic field at a point in space.

1.2 *Methods for Detecting Driver Behavior*

There are many researchers who have tried to detect driving behavior using the mobile phones' sensors.

Singh et al. [5] developed an android application, which first collects data from accelerometer sensor and GPS sensor. Also collects data from the microphone of the smartphone. Then analyzes the data and detects rash driving events (speed breaker, left turn, right turn, left lane change, right lane change, sudden braking, sudden acceleration). Rash driving events are verified using "Ground Truth". Also the data from the accelerometer are combined with the data from the microphone to detect more rash driving events (lane change is not accompanied with indicator sound) or traffic (slow speed with frequent honking).

Fazeen et al. [6] developed an android application for smartphones for detecting driver's behavior. They have used the accelerometer sensor of smartphones (which are integrated inside the cars) to collect and detect various driver styles or road conditions. They have analyzed the data from the accelerometer sensor (x axis, y axis) to measure the driver's direct control of the vehicle as they steer, accelerate or braking. We can detect the difference between safe and sudden acceleration/declaration because safe acceleration/declaration never reach a g-force of more than $+0.3/-0.3$ g. On the other hand sudden acceleration/declaration reach $+0.5/-0.5$ g. Safe left/right lane change never reaches a g-force of less than -0.1 g/ $+0.1$ g and sudden or unsafe lane change (left/right) reaches a g-force over -0.5 g/ $+0.5$ g. The results of the experiment showed us that the best phone placement location inside the car is the center console of the car. This location gave us the best relative data with the lowest engine feedback and the best results of predicting driving behavior. The disadvantage is that in any vehicle the placement of the smartphone can be anywhere and not only in the center console. So there should be a mechanism for virtually re-orienting the accelerometer. Also it was discovered that the average time to complete a safe lane change was 75 % longer than a sudden lane change.

Chigurupati et al. [7] proposed an android-based application to aware the driver about rash driving events and to improve driver's performance with feedback. The application uses the accelerometer sensor and the GPS sensor for data recording. Also uses the camera of the smartphone for video recording. Then data analyzed to detect rash driving events. The recommended range of accelerating or braking (front/rear direction, x-axis) is -3 g to $+3$ g. The recommended range of turning, swerves or lane change (left/right direction, y-axis) is also -3 g to $+3$ g and that of bumps or road anomalies (up/down direction, z-axis) is -8 to -11 g. So whenever the values of the accelerometer exceed the recommended values it would be consider as a rash driving event. The drawback of this system is that it is not automatic. There must be an administrator to analyze the videos.

Johnson et al. [8] developed an iOS application (they used an iPhone 4), which predicts and characterizes the style of the driver into normal, aggressive or very aggressive. So they built a system, which called MIROD (Mobile sensor platform

for Intelligent Recognition Of Aggressive Driving). This system with the help of the application collects data from accelerometer, GPS, gyroscope, magnetometer and uses the camera for video recording. All data from multiple sensors are fused into a single classifier based on the Dynamic Time Warping (DTW) algorithm. The MIROD system can detect the following events: right or left turns (90°), U-turns (180°), aggressive right or left turns (90°), aggressive U-turns (180°), aggressive acceleration or braking, swerve right or left (aggressive lane change), device removal and excessive speed. The system can detect only aggressive events. Safe changes (for example non aggressive lane change) are not being detected because they do not exert enough force or rotation to the device. If the system detect that a driver's style becomes aggressive provides audible feedback.

Dai et al. [9] have proposed an innovative mobile phone based system, which detects drunk driving patterns. The system was implemented on Android G1 Phone and they used the accelerometer and the orientation sensor. The system can detect (throw windowing and variation thresholding) and alert the drivers about 3 categories of drunk driving behaviors. The first category is related to lane position maintenance problems like weaving, driving, swerving and turning abruptly. The second category is related to speed control problems like suddenly accelerating or decelerating, braking erratically and stopping inappropriately. The last category is related to judgment and vigilance problems like driving with tires on center or lane marker, driving without headlights at night and slow response to traffic signals. In their experiment they have tested the detection performance in abnormal curvilinear movement and problem in maintaining speed. The result of the experiment shows 0 % false negative rate and 0,49 % false positive rate for abnormal curvilinear or lane changing movements. The result also shows 0 % false negative and 2,39 % false positive rate for speed control problems. The drawback of this system is that the set of drunk driving patterns were limited and it was difficult to distinguish them from safe and normal driving patterns.

Eren et al. [10] proposed an approach for estimating driving behavior and detecting if a dangerous driving pattern is safe or risky using smartphone sensors. The application of the smartphone collects data from accelerometer, gyroscope and magnetometer. Analyzing the sensor's data they obtain the speed, the position angle and the deflection from the regular trajectory of the vehicle. After the collection and the preprocessing of the sensor's data via smoothing filter they apply the endpoint detection algorithm. The endpoint detection algorithm helps to estimate the temporal range of the signal (detecting events like maneuvers). If they detect an event they use the Warping algorithm (DTW algorithm) to identify the type of the event overcoming different temporal durations of the same event across different drivers. In the end they apply Bayes Classification to identify if the event was a safe event or a risky event. In the experiment analyzed driving patterns of 15 different drivers. The results of the Bayesian Classification showed that the type of the event and if the driving style was safe or not, has been found correct for the 14 of the 15 drivers.

Chalermpol Saiprasert et al. [11] have proposed a high efficient report system using a mobile smartphone for detection and alert of dangerous over speed driving. The system collects stream of data from the smartphone's GPS sensor and produces

a time series of speed and location profile for a given route. After that a speeding detection algorithm is responsible of detecting whether a vehicle over speeding by finding anomalies in speed profile. If the system detects that a vehicle is speeding can alert in real time the passengers of the vehicle. Also the system has the ability to record the data of the journey of the over speeding vehicles to be used as evidence in case a passenger wants to make a report. Findings of the experiments showed that that the system identified correctly 3 out of 4 cases of anomalies in speed profile. The data from the sensors of the smartphones are the same accurate with the data we watch in the car's speedometer. The sensitivity and the accuracy of the collected data are affected by the reception condition. Also smartphones that are from different manufacturers for the same journey and the same stream of data produce "a fluctuation in instantaneous speed measurements of approximately +4 km/h due to GPS receivers with different capabilities".

Chuang-Wen You et al. [12] developed the first dual camera android application, which uses data from both cameras (front and back) of the smartphone to detect and alert drivers to dangerous driving behaviors and conditions inside and outside of the vehicle. Except the data from the cameras of the smartphone were used data of more smartphone's sensors like GPS, accelerometer and gyroscope. For the detection of dangerous driving behaviors or conditions were used computer vision and machine learning algorithms. The front camera is used to monitor and detect whether the driver of vehicle is tired or distracted. This is conducted using blink detection algorithms, which are detecting micro-sleep, fatigue and drowsiness. The back camera is used to track road conditions, like lane change condition and the distance between cars to determine if driver's car is too close to the car in front. If the system detects any dangerous driving behaviors or conditions it alerts the driver with an audible alert and on the smartphone's screen displays an attention icon. The results of the drowsiness detection experiment showed that the detection accuracy of detecting short blinks is 88,24 %, the detection accuracy of detection long blinks is 85,71 % and the overall accuracy of 87,21 %.

Fadi Aloul et al. [13] developed an innovative automatic notification system that uses an android based application to detect and report car accidents online. The aim of this system is to reduce fatalities by reducing the long response time required to notify emergency responders about traffic accident's data. The application collects data from the accelerometer sensor of the smartphone and then data analyzed using a Dynamic Time Warping (DTW) algorithm. After the data analysis, if there is an accident, the system detects the severity of the accident and the location of the accident using smartphone's GPS sensor. Also the system notifies police, first responders and registered emergency contact (sending an SMS) of the user's personal information (name, blood type, phone numbers of individuals etc.), the location and the severity of the accident. Findings of the experiments, testing the Dynamic time warping (DTW) algorithm showed that DTW algorithm has predict 23 out of 25 non-accident cases and all the 75 accident cases correctly. Also the experiment showed that the overall performance of the DTW algorithm in distinguishing an accident state and a non-accident state has 98 % accuracy.

Nidhi Kalra et al. [14] developed an android-based application, which collects data from accelerometer sensor and then data analyzed to detect patterns of driving events and road conditions. The collected data is raw values of x, y, z axis of accelerometer sensor. After the collection, the raw values have some problems (wide range, data is noisy, etc.) so they need some preprocessing. Then gradient data analyzed to detect driving events or road anomalies. The recommended range of normal braking (in negative direction, y-axis) and sudden braking (in negative direction, y-axis) is -1 g to -3 g and $< -3\text{ g}$ respectively. The recommended range of sudden forward acceleration (in positive direction, y-axis) is $>3\text{ g}$. The recommended range of left turn (in negative direction, x-axis) and right turn (in positive direction, x-axis) is $< -1\text{ g}$ and $>1\text{ g}$ respectively. Also the recommended range of pothole (change in value from positive to negative, z-axis), bump (change in value from positive to negative, z-axis) and rough road (change in value from positive to negative, z-axis) is $\pm 1.5\text{ g}$.

Jin-Hyuk Hong et al. [15] have proposed an in-vehicle sensing platform that uses android smartphone's sensors and 2 other externals sensors, a Bluetooth –based on board diagnostic reader (OBD2) and an internal measurement Unit (IMU) to predict aggressive driving style. The platform with the help of the sensors collects data during driving (speed, acceleration, deceleration) and then apply machine-learning techniques with a number of driving related features for an automated assessment of driving style. The OBD2 reads data from the vehicle like engine RPM, throttle position and vehicle speed. The IMU attached to the upper back of the steering wheel and captures the wheel movement using accelerometer, gyroscope and compass sensors. The findings of the experiment (driving data collection of 22 participants for 3 weeks) showed that the performance of the driver models, which are built using ML techniques in distinguishing an aggressive style and a non-aggressive style, has 90.5 % accuracy for violation-class and 81 % accuracy for questionnaire-class.

Johannes Paefgen et al. [16] have evaluated a mobile application for the assessment of driving behavior based on in-vehicle driving events and gives feedback to drivers. The implemented iOS application of the iPhone collects data from accelerometer, gyroscope and GPS sensors and then data transported to calibration component, determining the 3-dimensional orientation of the device in the vehicle. The calibration functional component contributes to the reliability and the accuracy of measurements. Then data sensors and calibrated parameters transported to trip recording component. During the trip recording sensor data is processed in a data-sampling component to detect critical driving event in real time. Users can access their data via the trip management module. Also they can receive driving feedback via the same module and share their performance to social networks. In their experiment 78 participants drove a vehicle for 45 min while the application was running. For the evaluation of the performance of the application, they installed inside the vehicle a dedicated off the shelf sensing system, which records reference data. Findings of the comparison of critical driving events generated by a

smartphone with reference measurements from a vehicle fixed IMU showed that the measurements from the smartphone tend to overestimate critical driving events. Responsible for that is the deviation from the calibrated initial device pose. The limitation of this work is that only one model was used and threshold values were very low to achieve high-resolution model.

Magana [17] uses the light sensor in the phone to obtain information about the environment in which the car is moving, because the brightness directly affects the visibility of the driver and this influences his anticipation. Another novel method in the work of Magana is the weather information involved in estimating the driving behavior. This information is obtained from the Internet connection of the smartphone.

Araujo et al. [18] present a smartphone application, which uses the information from the embedded sensors and the vehicles state information acquired from the vehicles CAN bus (speed, fuel consumption, GPS, etc.). The gathered data is passed to a fuzzy-based module, which analyzes the data and classifies it and then a suggestion is presented to the driver how to optimize the fuel energy consumption/driving behavior.

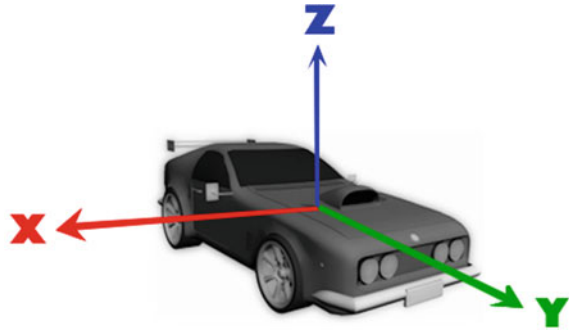
Murphey et al. [19] propose to categorize the different driving styles according to the measure how fast the driver accelerates and decelerates. The developed algorithm extracts jerk features from the current vehicle speed within a short time-window, and classifies the current driving style into three categories: calm, normal and aggressive, by comparing the extracted jerk feature with the statistics of the driver styles on the current roadway.

2 Calibration of the Device

2.1 *Vehicle Coordinate System*

In order to determine driver's behavior, we need to find, how the vehicle is acting. To be able to determine how the vehicle is acting we have to collect data from smartphone's sensors. So the smartphone's collected data will represent the behavior of the vehicle. In different positions and orientations of the device inside the vehicle we get different values. In order to get always a specific range of orientation and acceleration values we have to attach our device in a fixed position (before monitoring and during our trip) and to equate the sensor's axes with the vehicle axes. This equation is done with calibration process, by reorienting smartphone's sensor's axes according to vehicle axes. So vehicle and device will have a common coordinate system, which is the coordinate system of the vehicle. We can see the 3-axes of the vehicle coordinate system in the Fig. 1.

Fig. 1 Axis of vehicle coordinate system



2.2 Calibration

The axes of the vehicle and the axes of the smartphone or tablet are different each other as it could be seen in the figure above (Fig. 1). In order to get accurate (acceleration or orientation) data, in each orientation in which the device is placed inside the car, we have to reorient its axes relatively to the vehicle's axes (Fig. 2).

Calibrating [20] the device relative to the vehicle means that we have to rotate the sensor's axes in order to be the same with the vehicle's axes. So we have to find the rotation angles roll, pitch and yaw. The calibration process (Fig. 3) of the device is carried out in 3 steps and we insure the independence of the sensor's data from the orientation in the car and the device's position.

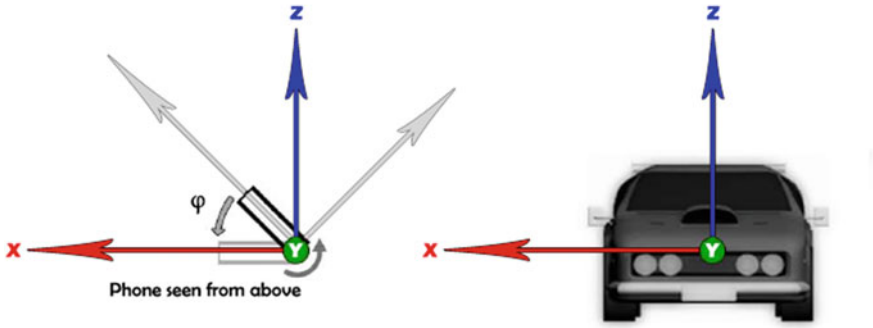
During the first step the vehicle must be motionless. In the first step we have to calculate the roll and pitch rotation angle according to the vehicle's level. The roll and pitch angles are calculated with the use of the atan2 function [21] which measures the angle between two given points. To calculate the rotation angle between two axes we see the accelerometer or sensor fusion data as polar coordinates provided to the atan2 function. Below we can see the equations which we use to calculate the roll and pitch angles.

$$roll = 2 * \arctan \left(\frac{sensor\ Vector[2]}{sensor\ Vector[0] + \sqrt{sensor\ Vector[0]^2 + sensor\ Vector[2]^2}} \right)$$

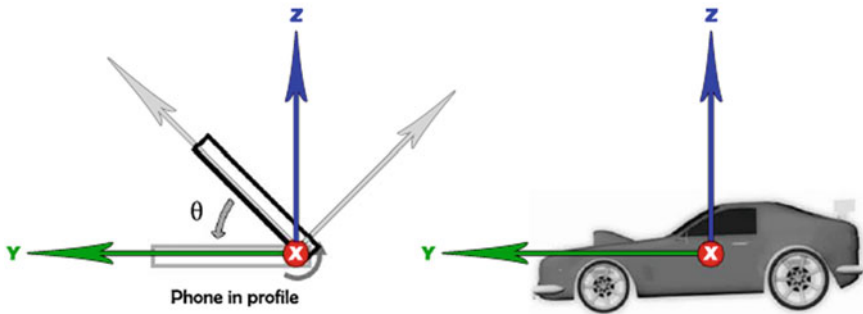
$$pitch = 2 * \arctan \left(\frac{sensor\ Vector[2]}{sensor\ Vector[1] + \sqrt{sensor\ Vector[1]^2 + sensor\ Vector[2]^2}} \right)$$

Then we calculate the XY magnitude offset using the rotated sensor data (according to roll and pitch angles computed before) which is the average magnitude (30 samples) between X and Y axis, in order to check if the vehicle is in motion. If the magnitude value is relatively the same value as the last one, the

Roll angle:



Pitch angle:



Yaw angle:

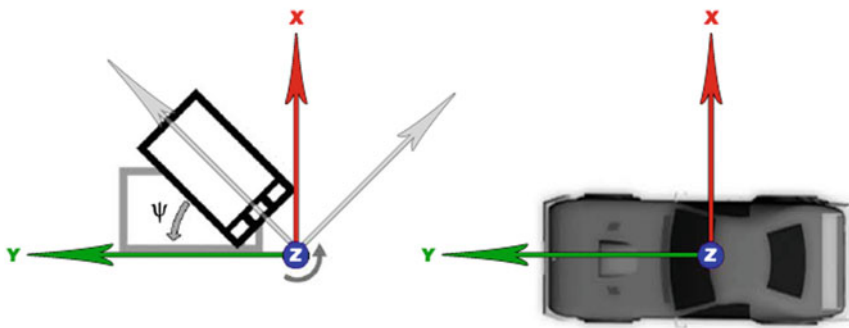


Fig. 2 Illustration of the calibration angles

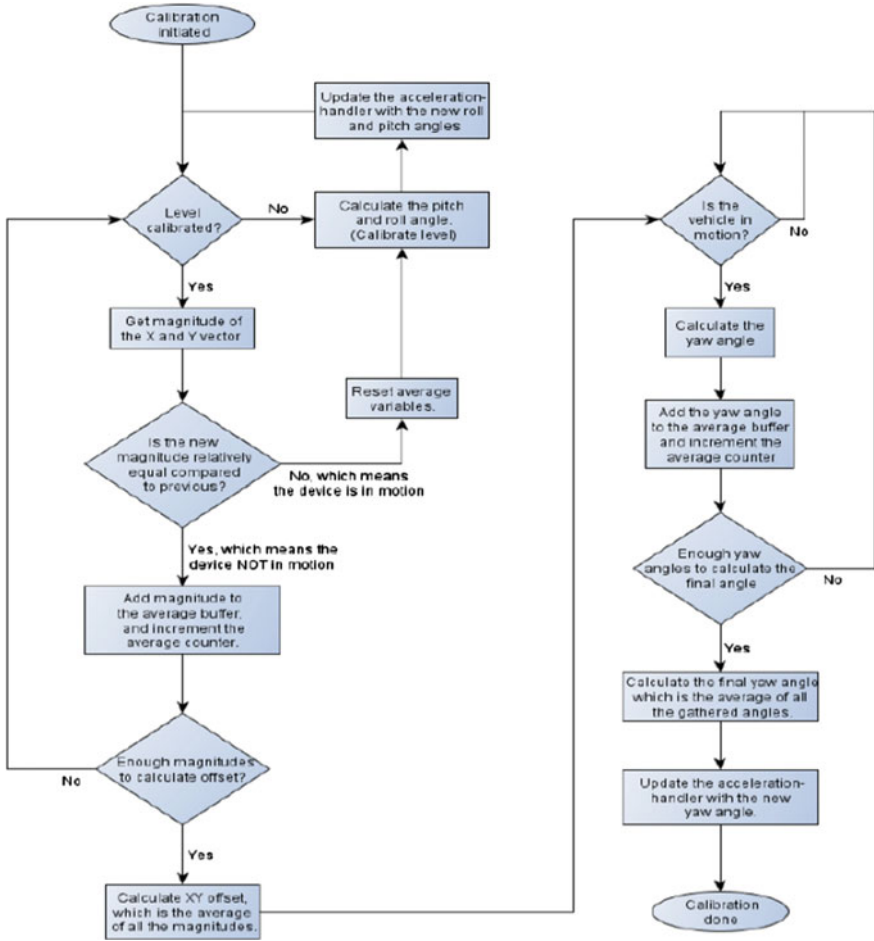


Fig. 3 Flow of the calibration process

vehicle is not in motion and the magnitude is added to a buffer and a counter is incremented.

$$magnitude = \sqrt{rotatedSensor\ Vector[0]^2 + rotatedSensor\ Vector[1]^2}$$

$$buffer = buffer + magnitude$$

$$counter = counter + 1$$

$$xyMagnitudeOffset = \frac{buffer}{counter}$$

If the vehicle is in motion we have to restart the calibration process.

In the second step the driver must start driving forward in order to calculate the yaw rotation angle. The yaw angle is the angle of the driving direction of the vehicle. First we calculate the driving direction magnitude with the help of the following equation:

$$drivingDirectionmagnitude = \sqrt{rotatedSensorVector[0]^2 - rotatedSensorVector[1]^2 - xyMagnitudeOffset}$$

If the driving direction magnitude is greater than a determined threshold, the vehicle is in motion and we calculate some yaw angles using atan2 function like before.

$$yaw = 2 * \arctan\left(\frac{sensorVector[0]}{sensorVector[1] + \sqrt{sensorVector[1]^2 + sensorVector[0]^2}}\right)$$

We have to compute enough angles in order to get the right direction. If we have enough angle values we get the average yaw angle and set it as the yaw angle.

In the last step we have to update the rotation angles in the module of the sensor data in order to get accurate readings. The rotated sensor data around axes are calculated with the help of the following equations:

Rotated sensor data around y-axis:

$$sensorData[0] = sensorData[0] * \cos(roll) + sensorData[2] * \sin(roll)$$

$$sensorData[2] = sensorData[0] * \cos(roll) - sensorData[2] * \sin(roll)$$

Rotated sensor data around x-axis:

$$sensorData[1] = sensorData[1] * \cos(pitch) + sensorData[2] * \sin(pitch)$$

$$sensorData[2] = sensorData[1] * \cos(pitch) - sensorData[2] * \sin(pitch)$$

Rotated sensor data around z-axis:

$$sensorData[0] = sensorData[0] * \cos(yaw) + sensorData[1] * \sin(yaw)$$

$$sensorData[1] = sensorData[0] * \cos(yaw) - sensorData[1] * \sin(yaw)$$

3 Orientation Data via Sensor Fusion Method

The best method to determine the orientation of a device (smartphone or tablet) is sensor fusion [22]. This method combines data from 3 sensors (accelerometer, magnetometer and gyroscope) in order to get the three orientation angles. We could

get these three angles either using data only from accelerometer and magnetometer or using data only from gyroscope. However both ways doesn't work accurately.

Using the first way to get the orientation of the device (determine the direction of magnetic north and south), the accelerometer will provide the gravity vector (we can calculate the pitch and roll angles) and the magnetometer will work as a compass (we can calculate the azimuth angle). Both output data of sensors are inaccurate and noisy, especially the magnetometer's output data (low reaction time).

Using the gyroscope of the device we can get accurate data very fast (with short response time). The output data of the gyroscope provide the angular velocity speed for each axis. By multiplying these angular velocity measurements with the time interval between the last and the current gyroscope output data, we get the orientation angles of the device because we have a rotation increment. The main disadvantage of this way is gyro drift, which is a slow rotation of the calculated orientation. Gyro drift is caused of small errors, in each iteration of multiplying and adds up over time. This leads to a slow rotation of the calculated orientation.

So we are going back to say that the right way to get the orientation of the device is to combine all the above-mentioned sensors. The result of this combination is to avoid both noisy orientation and gyro drift. With this method counterbalance the weakness of the one sensor with the strength of the other. The low noise gyroscope output data is used only for orientation changes in short time intervals. The accelerometer and magnetometer output data is used as support information over long time intervals. With this way we filter out the gyro drift, with the accelerometer/magnetometer data, which do not drift over long time intervals. This process is equivalent to high pass filtering of the gyroscope output data and to low pass filtering of the accelerometer/magnetometer output data. This configuration called complementary filter. In the figure below we can see the overall flow of sensor fusion process [23] (Fig. 4).

All the sensors provide their output data at constant time intervals and these data can be shown as signals in a graph, with the time as the x-axis. The low pass filtering of the accelerometer and magnetometer data was done, by averaging, the orientation angle values over time within a constant time window. Every time a new accelerometer/magnetometer value is available it is weighted with a low pass factor

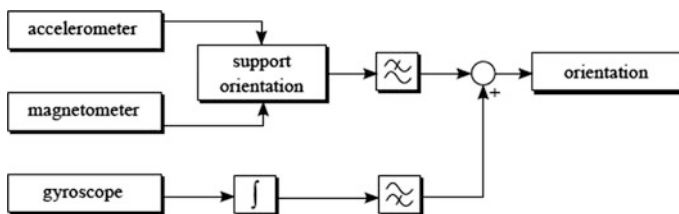


Fig. 4 Flow of the Sensor Fusion with complementary filter

$(1 - a)$ and added to the absolute orientation. The factor (a) is the balance factor and it's equal to 0.98. The equation for the accelerometer/magnetometer orientation is:

$$a = 0.98$$

$$AccMagOrientation = a * AccMagOrientation + (1 - a) * newAccMagOrientation$$

The high pass filtering of the gyroscope data is done by replacing, the high pass component $AccMagOrientation$ with the corresponding gyroscope orientation data. Thus we have the final sensor fusion orientation, which is:

$$SensorFusionOrientation = a * GyroscopeOrientation + (1 - a) * AccMagOrientation$$

In the figure below we can see the intermediate signals in the filtering process, when a device is turned 90° in a direction and after a short time turned back to its initial position. We can see the gyro drift in the gyroscope output data due to the small irregularities in the original angular speed. These deviations add up during the integration and cause an additional slow rotation of the gyroscope-based orientation [24] (Fig. 5).

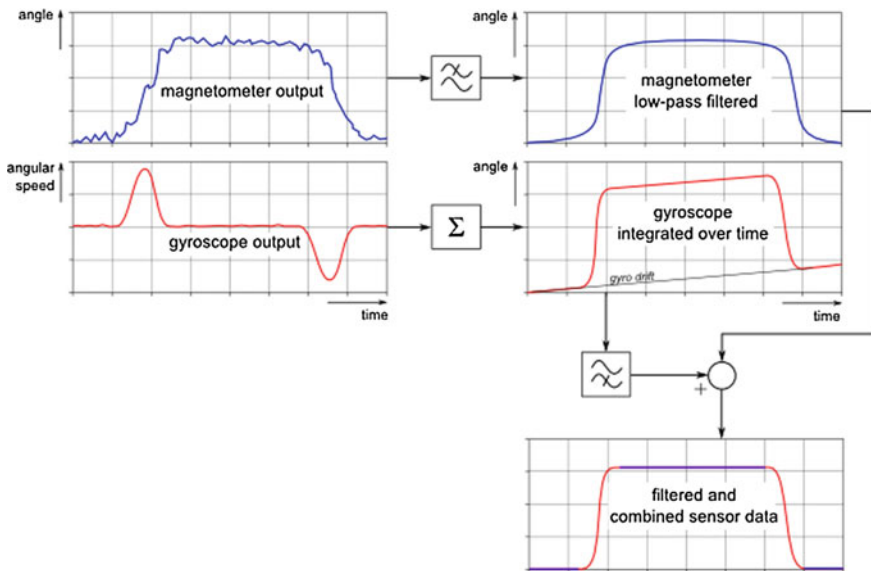


Fig. 5 Intermediate signals in the filtering process when we assuming that the device is turned 90° in one direction and after a short time turned back to its initial position

4 Detection Methods and Testing

For the detection of the driving behavior of the driver, we use 2 different methods. In the first detection method we use the output data of the accelerometer sensor. Collecting and evaluating the values of the accelerometer sensor we can detect how the user drives. In the second method we collect and analyze the orientation values, which we obtain with sensor fusion discussed in the previous chapter.

The required thresholds are acquired by the analysis of data collected during tests and are discussed in Sects. 4.1 and 4.2. The collected (or monitored) values are from the accelerometer (x and y axis) and the orientation sensors (pitch and roll). We collect data using both methods during normal driving events like acceleration, turns and lane changes. Also we collect data during abnormal rash driving events and maneuvers like hard acceleration or deceleration, sharp turns and sharp lane changes.

All data from all sensors (accelerometer, magnetometer, gyroscope) of both methods is run separately through an EMA (exponential Moving Average) algorithm [25]. This algorithm works as a low pass filter and handles the noise from the sensors. We check data (output data of EMA algorithm) of both methods in time window frames of 5 s. In this time window all axis data of the accelerometer and of the sensor fusion are stored and analyzed. For every such 5 s window frame we compute the minimum and the maximum value of all axis data.

The collection of the driving events was conducted in areas where there was not traffic at all. Also for the collection and testing data were used 3 different devices. A Samsung Galaxy tablet (2014 edition), a Samsung Galaxy s3 smartphone (2013 edition) and a LG G3 (2014 edition). The collected data from all devices matched each other. So the sensor's output data more or less are the same for different devices. The device was placed in a fixed position in the central console of the vehicle. After the calibration procedure and while monitoring of the driving behavior is in progress the device should not be moved from its fixed position. If the device is moved while monitoring, the sensor's output data will be wrong and the evaluation of the data will not be accurate.

In the next chapters and for both detection methods, we can see which sensor data we used to detect every driving event. Also we can see graphs for every driving event where the maneuver is clearly distinguished.

4.1 Detection Method Using Accelerometer Data

In this method we have used the three-axis accelerometer sensor to record and analyze various driving events and driver's behavior. We have utilized the x- axis data to detect the left/right direction of the vehicle and therefore driving events like safe or sharp turns and safe or sharp lane changes. For the detection of front/rear direction of vehicle and therefore to measure how the driver accelerates and applies the brakes we have utilized the y-axis data.

After collecting and analyzing our data we determined the threshold values of various events and maneuvers. Acceleration or deceleration of the vehicle is determined by the change in acceleration in y-axis. Safe acceleration is considered when the y-axis value is between 1.3 m/s^2 and 2.5 m/s^2 . When the value is more than 2.5 m/s^2 it is considered as hard (sudden) acceleration. Similar to acceleration, it is considered safe deceleration (normal braking) when we have a value between -1.3 m/s^2 and -2.5 m/s^2 . When the y-axis value is lower than -2.5 m/s^2 it is considered as hard deceleration (sudden braking). In the following table and graphs we can see the thresholds and patterns for the acceleration/deceleration driving events (Figs. 6 and 7, Table 1).

Left or right turn is determined by the change in acceleration in x-axis. As safe left turn is considered when we have x-axis value between -1.8 m/s^2 and -3 m/s^2 . Whenever the accelerometer x-axis value exceeds the upper safe limit (-3 m/s^2), it would be considered as sharp left turn. Similar to left turn, we have considered as safe right turn when the acceleration in x-axis has value from 1.8 m/s^2 to 3 m/s^2 and

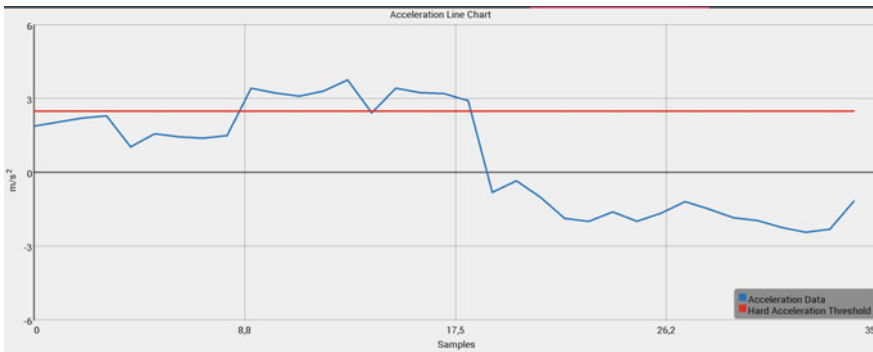


Fig. 6 Safe acceleration pattern and then a hard acceleration pattern

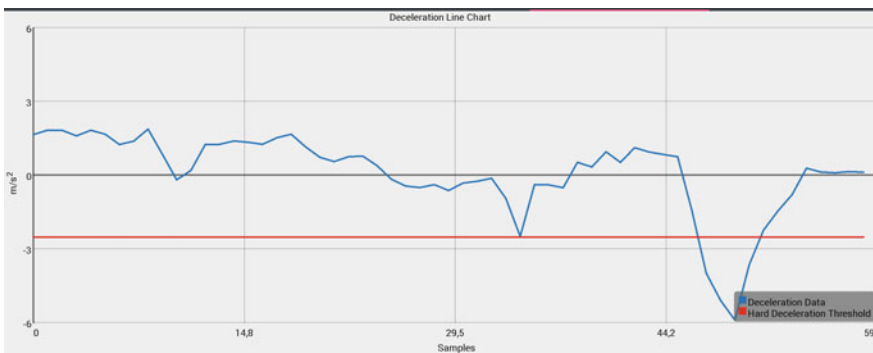


Fig. 7 Safe deceleration pattern and then a hard deceleration pattern

Table 1 Thresholds and data used for the detection of a Safe/Hard—Acceleration/Deceleration using accelerometer’s data

Driving event	Data used for detection	Threshold
Safe acceleration	Y-axis data	1.3 to 2.5 m/s ²
Hard acceleration	Y-axis data	>2.5 m/s ²
Safe deceleration	Y-axis data	-1.3 to -2.5 m/s ²
Hard deceleration	Y-axis data	<-2.5 m/s ²

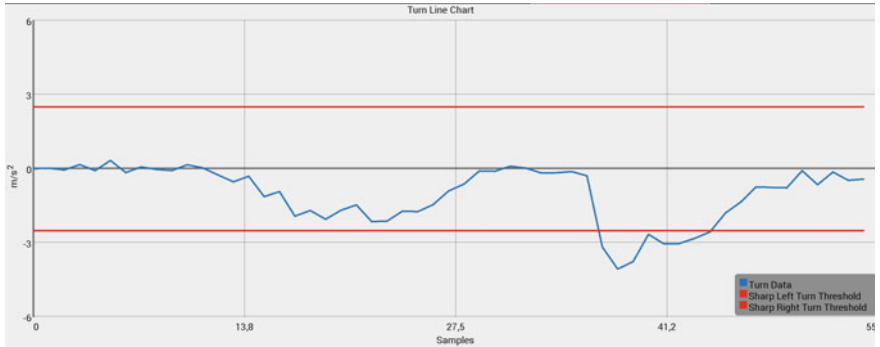


Fig. 8 Safe left turn pattern and then a sharp left turn pattern

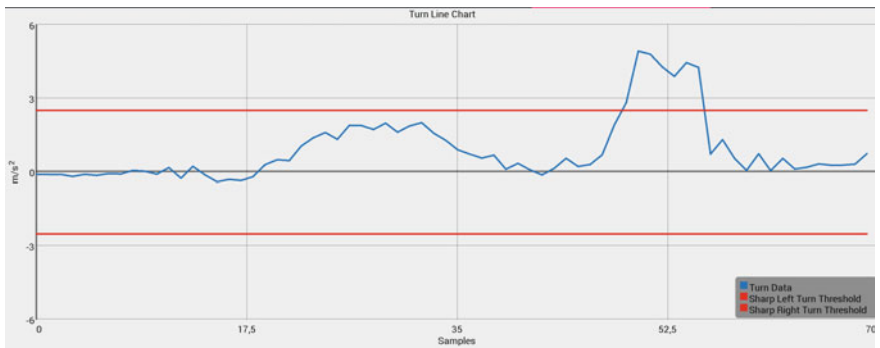


Fig. 9 Safe right turn pattern and then a sharp right turn pattern

as sharp right turn when the value exceeds the upper safe limit (3 m/s²). In the following table and graphs we can see the thresholds and patterns for the safe/sharp left/right turn driving events (Figs. 8 and 9, Table 2).

A lane change also is determined by the change in acceleration in x-axis and by the number and the type of turns within a specific time frame. As safe right lane change is considered when we have a safe right turn and in the next 2 s time window frame we have a safe left turn. As safe left lane change is considered when

Table 2 Thresholds and data used for the detection of a Safe/Sharp—Left/Right Turn using accelerometer’s data

Driving event	Data used for detection	Threshold
Safe left turn	X-axis data	-1.8 to -3.0 m/s ²
Sharp left turn	X-axis data	<-3.0 m/s ²
Safe right turn	X-axis data	1.8 to 3.0 m/s ²
Sharp right turn	X-axis data	>3.0 m/s ²

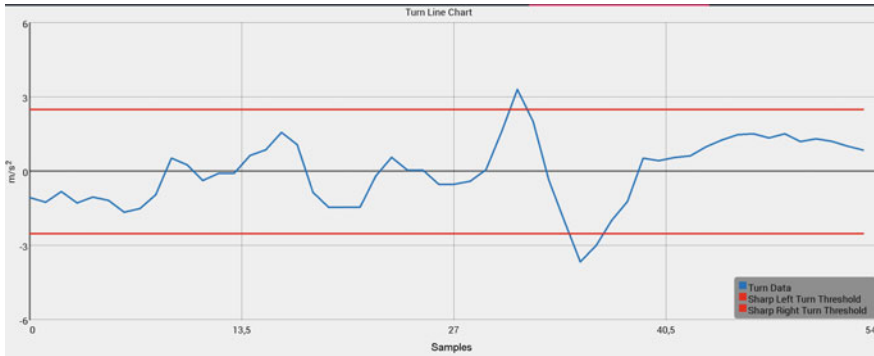


Fig. 10 Safe right lane change pattern and then a sharp right lane change pattern

in a 2 s time window occur two safe turns. The first must be a safe left turn and the second, a safe right. On the other hand, in the case of sharp lane changes we have 2 cases. In the first case one of the turns must be sharp and in the second case both of them must be sharp. For example we have a sharp right lane change when occur 2 turns (the first right and the second left) in a 2 s time window and the first turn is safe and the second sharp or the first turn is sharp and the second is safe. Also as sharp right lane change is considered when both of the turns are sharp, the first one a sharp right turn and second one a sharp left turn.

In the graphs we can see the thresholds and patterns for the safe/sharp lane change driving events (Figs. 10 and 11).

4.2 Detection Method Using Sensor Fusion Orientation Data

In this method we have used the three-axis orientation output data of sensor fusion in order to collect and analyze various driving events. Orientation is defined as a combination of three angular quantities: azimuth, pitch and roll. These three quantities are defined based on 3-axis. The positive X-axis extends out of the right

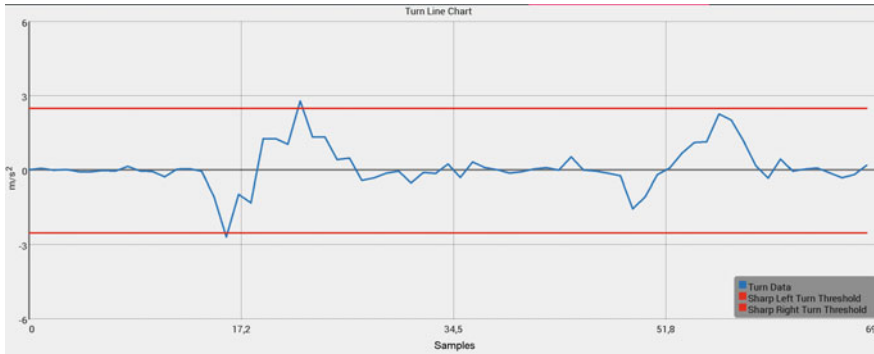


Fig. 11 Safe left lane change pattern and then a sharp left lane change pattern

side of the car, positive Y-axis extends out of the front side of the car and the positive Z-axis extends out of the topside of the car. Azimuth is angle between the positive Y-axis and magnetic north and its range is between 0 and 360°. Positive roll is defined when the car starts by laying flat on the road and the positive Z-axis begins to tilt towards the positive X-axis. Positive pitch is defined when the car starts by laying flat on the road and the positive Z-axis begins to tilt towards the positive Y-axis.

We have utilized roll data to detect the left/right direction of the vehicle and therefore driving events like safe or sharp turns and safe or sharp lane changes. For the detection of front/rear direction of vehicle and therefore to measure how the driver accelerates and applies the brakes we have utilized pitch data.

After collecting and analyzing our data we determined the threshold values of various events and maneuvers. Acceleration or deceleration of the vehicle is determined by the change in orientation in pitch angle. Safe acceleration is considered when the pitch value is between -0.08 rad/s and -0.12 rad/s. When the value is less than -0.12 rad/s it is considered as hard (sudden) acceleration. Similar to acceleration, it is considered as safe deceleration (normal braking) when we have a value between 0.08 rad/s and 0.12 rad/s. When the pitch value is higher than 0.12 rad/s it is considered as hard deceleration (sudden braking). In the following table and graphs we can see the thresholds and patterns for the acceleration/deceleration driving events of sensor fusion method (Figs. 12 and 13, Table 3).

Left or right turn is determined by the change in orientation in roll angle. Safe left turn is considered when the roll value is between 0.10 rad/s and 0.30 rad/s. Whenever the roll value exceeds the upper safe limit (0.30 rad/s), it is considered as a sharp left turn. Similar to left turn, we have considered as safe right turn when the roll angle has value from -0.10 rad/s to -0.30 rad/s and as a sharp right turn when the value exceeds the upper safe limit (-0.30 rad/s). In the following table and graphs we can see the thresholds and patterns for safe/sharp left/right turn driving events of sensor fusion method (Figs. 14 and 15, Table 4).

In case we want to detect safe and sharp lane changes using orientation data, the philosophy and the methodology is the same as in the previous method, detection

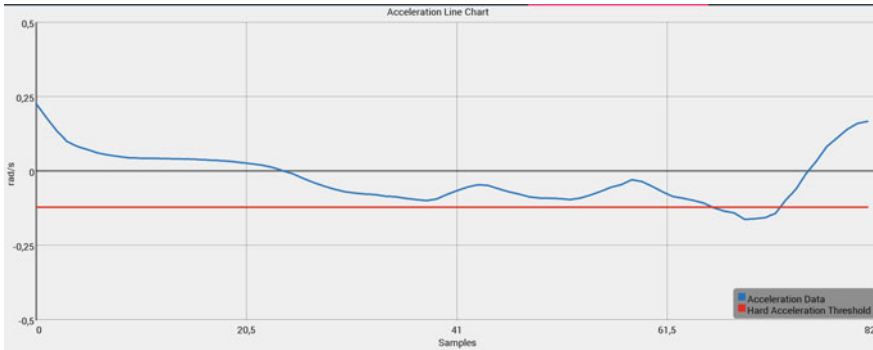


Fig. 12 Two safe acceleration patterns and then a hard acceleration pattern

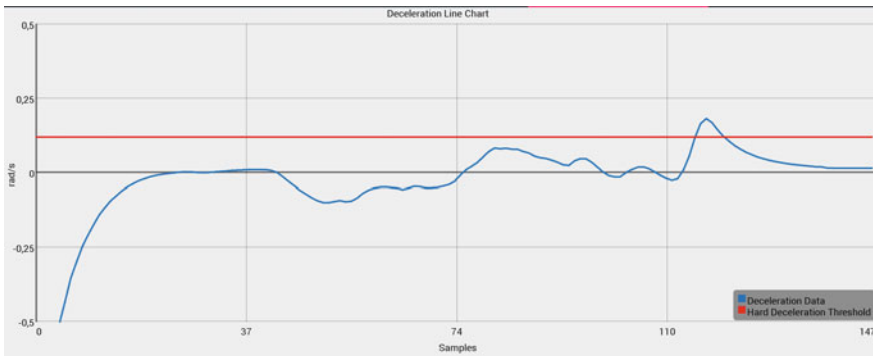


Fig. 13 A safe deceleration pattern and then a hard deceleration pattern

Table 3 Thresholds and data used for the detection of a Safe/Hard–Acceleration/Deceleration using orientation data of Sensor fusion method

Driving event	Data used for detection	Threshold
Safe acceleration	Pitch angle	−0.08 to −0.12 rad/s
Hard acceleration	Pitch angle	<−0.12 rad/s
Safe deceleration	Pitch angle	0.08 to 0.12 rad/s
Hard deceleration	Pitch angle	>0.12 rad/s

using accelerometer data. A lane change also is determined by the number and the type of turns in a specific time. As safe right lane change is considered when we have a safe right turn and in the next 2 s time window frame we have a safe left turn. Similarly, as safe left lane change is considered when in a 2 s time window occur two safe turns. The first must be a safe left turn and the second, a safe right turn. On the other hand, in the case of sharp lane changes we have 2 cases. In the first case one of the turns must be sharp and in the second case both of them must be sharp.

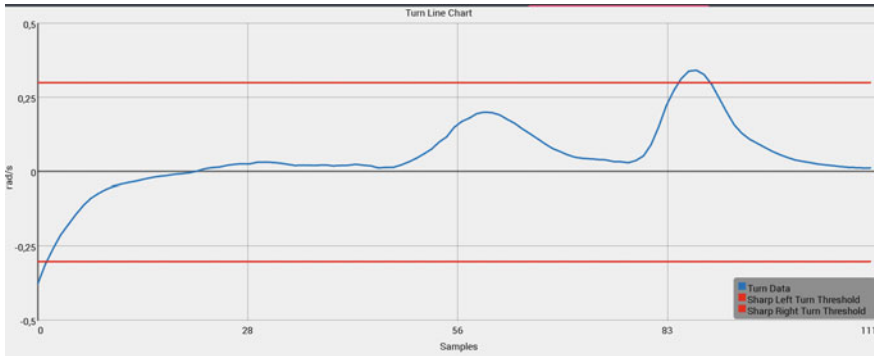


Fig. 14 A Safe Left Turn pattern and then a Sharp Left Turn pattern

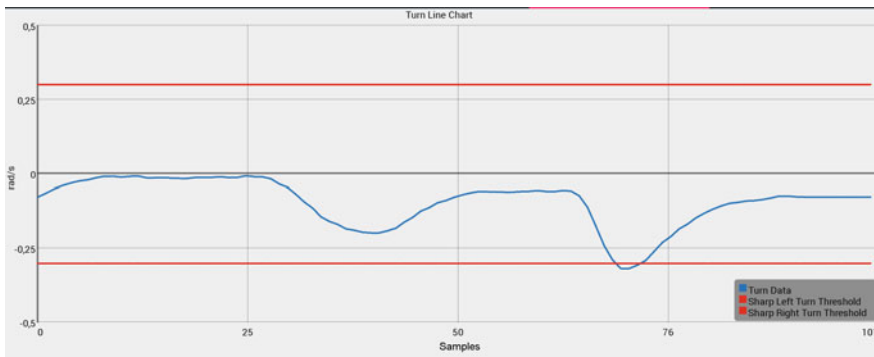


Fig. 15 A Safe right turn pattern and then a sharp right turn pattern

Table 4 Thresholds and data used for the detection of a Safe/Sharp—Left/Right Turn using orientation data of the sensor fusion method

Driving event	Data used for detection	Threshold
Safe left turn	Roll angle	0.10 to 0.30 rad/s
Sharp left turn	Roll angle	>0.30 rad/s
Safe right turn	Roll angle	-0.10 to -0.30 rad/s
Sharp right turn	Roll angle	<-0.30 rad/s

For example we have a sharp right lane change when occur 2 turns (the first right and the second left) in a 2 s time window and the first turn is a safe turn and the second a sharp turn or the first turn is sharp and the second is safe. Also as sharp right lane change is considered when both of the turns are sharp, the first one a sharp right turn and the second one a sharp left turn.

In the graphs we can see the thresholds and patterns for safe/sharp lane change driving events (Figs. 16 and 17).

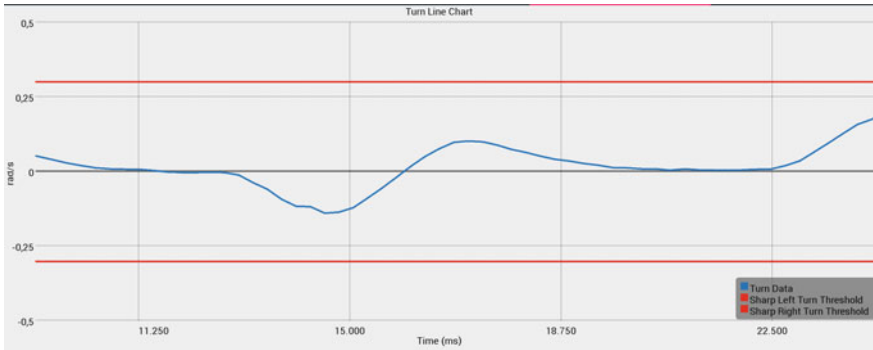


Fig. 16 A safe right lane change pattern

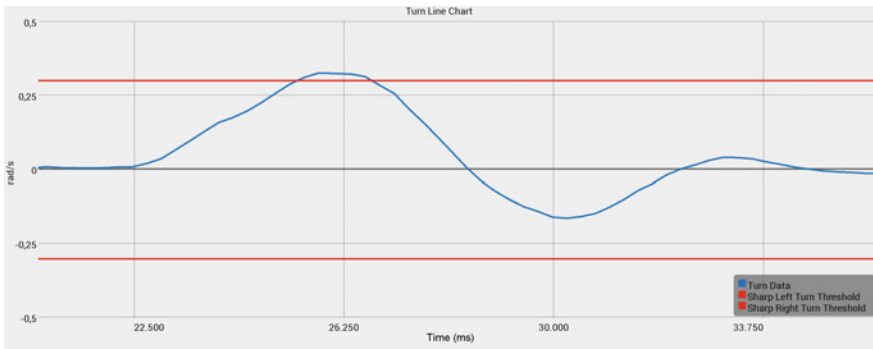


Fig. 17 A sharp left lane change pattern

4.3 Driver Behavior Detection Algorithm

Our algorithm for the detection of driver behavior uses two types of data. The acceleration output data of the accelerometer sensor and the orientation output data of the sensor fusion method. We can detect the behavior of the driver using either acceleration data, either orientation data. The driver can choose which detection method will be used for the detection of his behavior. It can be chosen only one method (type of data). The driver’s behavior detection algorithm doesn’t work, with the use of both data types at the same time.

After we choose detection method, the algorithm analyzes the behavior of the driver based on thresholds of detection data. These thresholds are acquired by testing the data of the detection methods under various driving events and maneuvers. Detection methods and testing is discussed in the Sect. 4. With these thresholds we can distinguish and detect 12 driving events. Six of them are safe driving events and the other six are dangerous driving events. The safe events are: Safe Acceleration, Safe Deceleration, Safe Left Turn, Safe Right Turn, Safe Left

Lane Change and Safe Right Lane Change. The dangerous events are: Hard Acceleration, Hard Deceleration, Sharp Left Turn, Sharp Right Turn, Sharp Left Lane Change and Sharp Right Lane Change.

Our algorithm for the detection of the driver's behavior, works in time window frames of 5 s. In these 5 s all data (3-axis values) of the detection method (accelerometer or sensor fusion data) are stored in a list. In every time window frame, new values are inserted in the list and the same time the oldest values are removed from the same list. Then we check every data (3-axis value) of the list. If the current value, used for the detection of maneuvers is below the threshold of safe driving events, means that none of various maneuvers of which the system can detect is happening. When the current value is bigger than the threshold for safe driving events and below the threshold for dangerous driving events means that, one of the safe-driving events is happening. At last if the monitored current value exceeds the threshold for a dangerous maneuver means that we have a dangerous situation like sharp right turn.

In case of Lane Changes the algorithm checks if in time window frame of 2 s have occurred two turns. One left and one right or the opposite. If one of them turns or both of are sharp turns, which means that the value, used for the turn detection of the car is higher than the threshold for a dangerous turn, a sharp lane changing is happening. On the other hand if both of them are safe turns, a safe lane changing is happening. If the first turn is left turn we have a left lane change and if it is right we have a right lane change.

In Table 5 we can see the summary thresholds of both methods, which are used for the detection of various *safe* and *dangerous* maneuvers.

For the computation of the average speed (avgSpeedKM) of the vehicle at the end of every trip, we have to compute first the total time (totalHours) we were driving and the total distance (distanceKM) we traveled. When we start a new trip

Table 5 Thresholds and data used for the detection of various driving events using accelerometer's data or sensor fusion orientation data

Driving event	Data used (accelerometer)	Threshold	Data used (sensor fusion)	Threshold
Safe acceleration	Y-axis data	1.3 m/s ² to 2.5 m/s ²	Pitch angle	-0.08 to -0.12 rad/s
Safe deceleration	Y-axis data	-1.3 to -2.5 m/s ²	Pitch angle	0.08 to 0.12 rad/s
Safe left turn	X-axis data	-1.8 to -3.0 m/s ²	Roll angle	0.10 to 0.30 rad/s
Safe right turn	X-axis data	1.8 to 3.0 m/s ²	Roll angle	-0.10 to -0.30 rad/s
Hard acceleration	Y-axis data	>2.5 m/s ²	Pitch angle	<-0.12 rad/s
Hard deceleration	Y-axis data	<-2.5 m/s ²	Pitch angle	>0.12 rad/s
Sharp left turn	X-axis data	<-3.0 m/s ²	Roll angle	>0.30 rad/s
Sharp right turn	X-axis data	>3.0 m/s ²	Roll angle	<-0.30 rad/s

the `startTrip()` function is called. During the trip, the total distance travelled is computed inside the `updateLocation()` function. This function is called every time we have a new `Location` from GPS. When the trip is finished the `stopTrip()` function is called and the average speed is computed by dividing the total distance with the total time.

Below we can see the functions, which are called for the computation of the average speed of the vehicle for every trip.

```
updateLocation(Location location)
{
    // if there is a previous location
    if (previousLocation != null) {
        // add to the total distanceTraveled
        distanceTraveled += location.distanceTo(previousLocation);
    } // end if

    previousLocation = location;
}

startTrip()
{
    driving = true;
    startTime = System.currentTimeMillis(); // get current time
    previousLocation = null; // starting a new trip
}

stopTrip()
{
    driving = false; // just stopped tracking locations
    MILLISECONDS_PER_HOUR = 1000 * 60 * 60;

    // compute the total time we were driving
    long milliseconds = System.currentTimeMillis() - startTime;
    double totalHours = milliseconds / MILLISECONDS_PER_HOUR;

    double distanceKM = distanceTraveled / 1000.0;
    double avgSpeedKM = distanceKM / totalHours;
}
```

5 Native Android Application for Driving Behavior Recognition

We developed a native android-based application that can detect and evaluate the driving behavior of the user for all his trips. All trip data that contains statistics, routes and graphs is saved in smartphone's local memory. In the subsections below we can see how the application works, what data are recorded and how it is presented to the user.

When the user is successfully connected, the main menu is presented. In the main menu there are 4 options. The first option is the "New Trip". We choose this option when we want to start a new trip. The second option is the "My trips" and we choose it when we want to review our trips (routes and statistics). In the third option there are our settings and the last option is "Help" where we can find tutorials and information about how we use the application. We can see the main menu in the Fig. 18.

As already motioned above, we choose the "New Trip" option when we want to start a new trip. Before we start driving, the application will tell us to follow some instructions for the calibration of the device. We set the calibration procedure for the device in order to get accurate readings from the sensor's data in any fixed orientation of the device inside the vehicle. In the figure below we can see the instructions for the calibration of the device (Fig. 19).

During the calibration process the device must be attached in a fixed position and the vehicle must be steel. The whole process takes about 5 s to complete. When the signal is given we start driving to our destination. We can see the given signals in the Fig. 20.

In the beginning of the process the signal to keep vehicle steel is displayed and in a few seconds the "Drive vehicle forward" signal is displayed. When we start driving

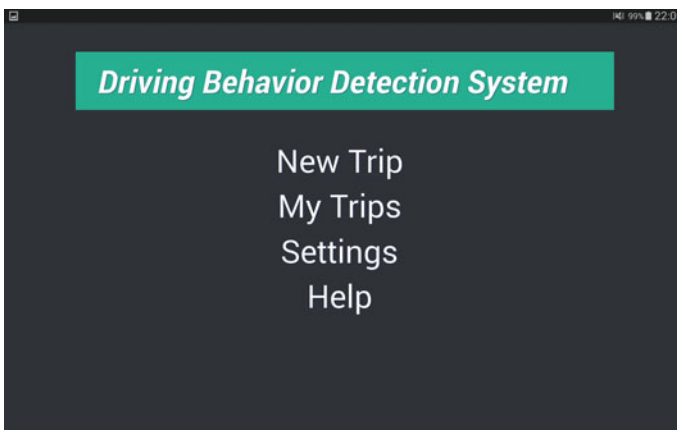


Fig. 18 The main menu of our application

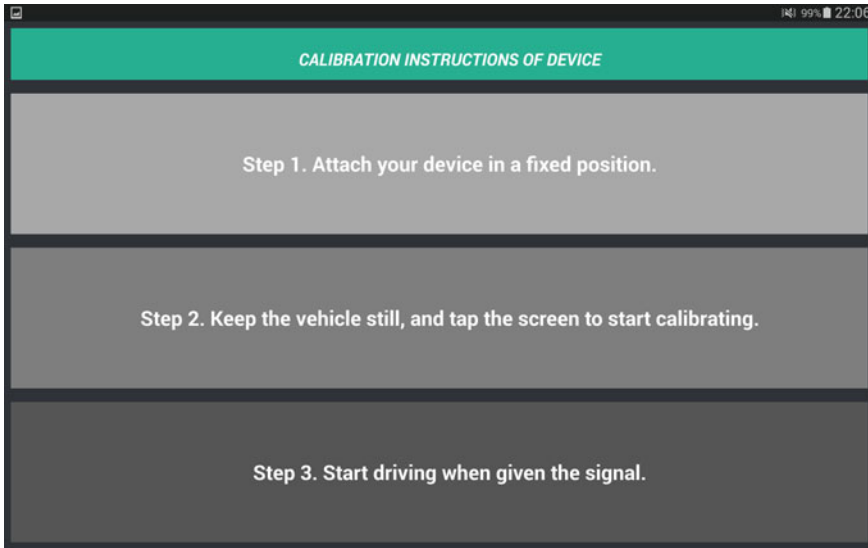


Fig. 19 The calibration instructions screen

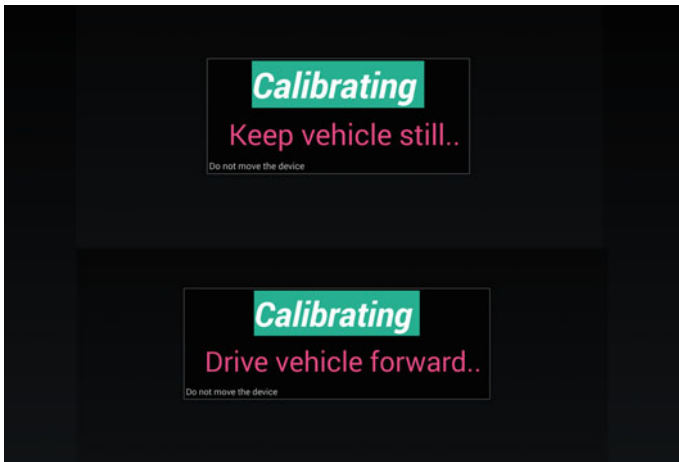


Fig. 20 The given calibration signals as they are presented in our application

forward the calibration is completed and the app starts monitoring our driving behavior until the end of our trip. At this phase the main screen displays the “Monitoring Driving Behavior” message (Fig. 21).

When the system detects a safe driving event (maneuver) the main screen color changes to green and the message also changes to the name of the current safe driving event (Fig. 22).

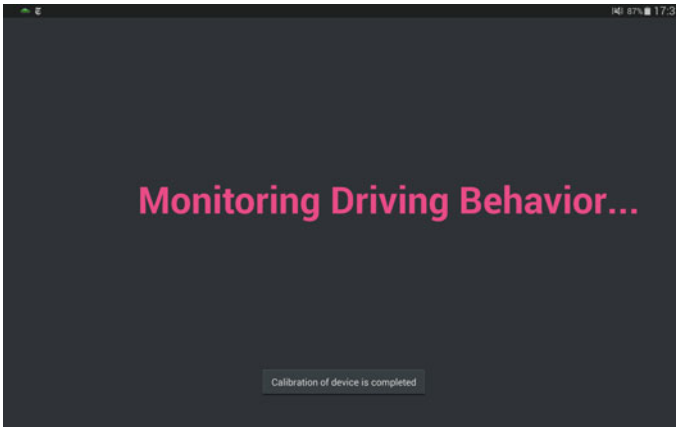


Fig. 21 The basic monitoring option. The system is not detecting any safe or dangerous driving event



Fig. 22 The system detects a safe deceleration

When the system detects a dangerous driving event the color of the main screen changes to yellow and the message changes to “Attention!” followed by the name of the dangerous event. For all the dangerous driving events except the attention message there is also displayed a hint. The hint helps the driver to improve his driving skills. Also, providing constructive feedback to drivers is very important since it helps them to correct bad driving behaviors. The hints that are displayed are: “try to maintain uniform acceleration (or deceleration)”, “try to take left (or right) turn slower” and “try to change the left (or right) lane slower”. The attention message and hint is followed by a notification sound. In the figure below we can see the screen of our device when a sharp left turn is detected (Fig. 23).

By pressing the back button of our device we finish our trip. After that, the system is loading and presenting all information about our trip. All information about trips is presented in 3 different tabs: the info, the map and the graph tabs.

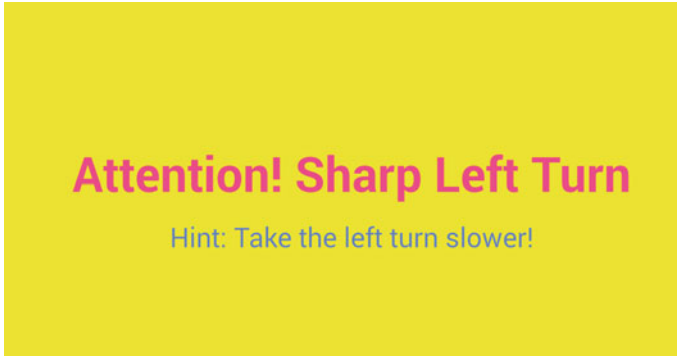


Fig. 23 The system detects a dangerous sharp left turn

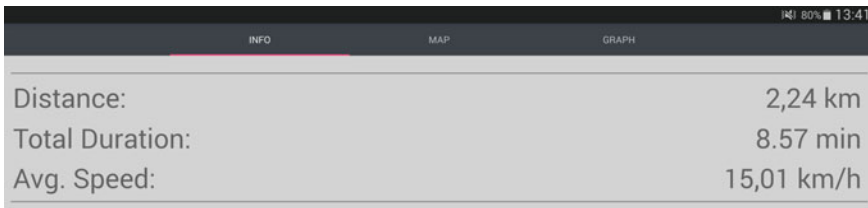


Fig. 24 Info Tab screen of our application

In the info tab (Fig. 24) some general statistics about the trip are presented, like the total distance, the total duration and the average speed.

In the map tab the route of our trip is presented based on Google maps. In the map we can see the starting point of our trip, which is represented by a blue marker and the end point of our trip, which is represented by a light green marker. Also we can see at which point of our route, we committed dangerous driving events. All dangerous driving points are represented with a red marker. When we tap on a dangerous driving point we are presented the name of the event. In Fig. 25 we can see an example of the map tab with the route and the markers as we mentioned before.

The graph tab presents the acceleration line chart, the deceleration line chart and the turn line chart of the trip. We can see one of the 3 charts at the time. If we want to change graph chart we tap on the radio button of the chart we want under the displayed chart.

The x-axis represents the time in milliseconds and the y-axis represents the values of the detection method we have already chosen. These values can be acceleration values (acceleration detection method— m/s^2) or orientation values (sensor fusion method— rad/s).

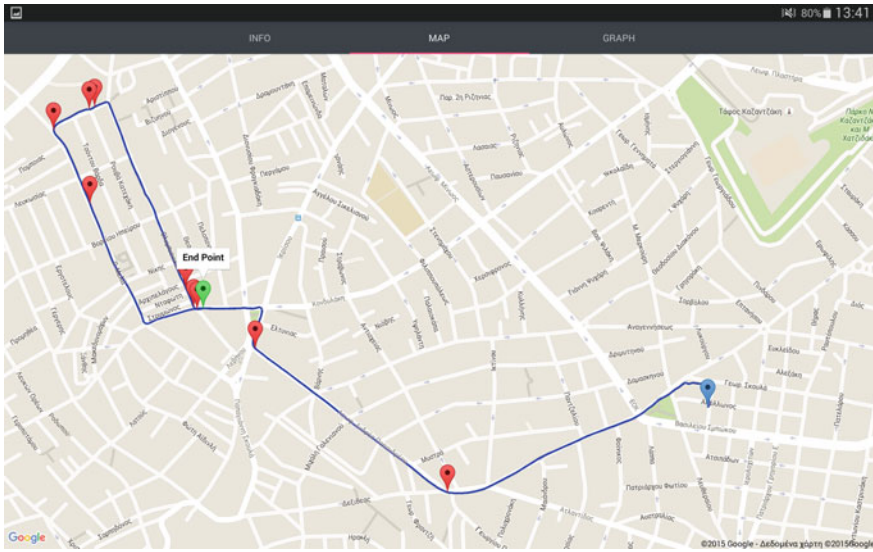


Fig. 25 Map Tab screen of our application

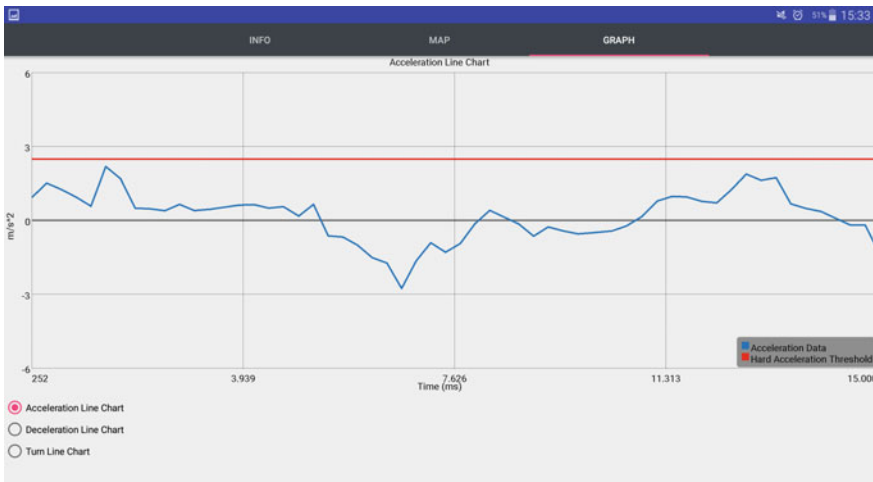


Fig. 26 Acceleration line chart of graph Tab screen of our application

The charts are scrollable in the x-axis, which means that by scrolling horizontally we can see the acceleration or orientation values as a function of time. The displayed x-axis duration is 60 s and as we scroll we can see the next 60 s.

Except the acceleration, deceleration and turn line charts we can see the thresholds lines for each chart. The thresholds line is represented with a red direct

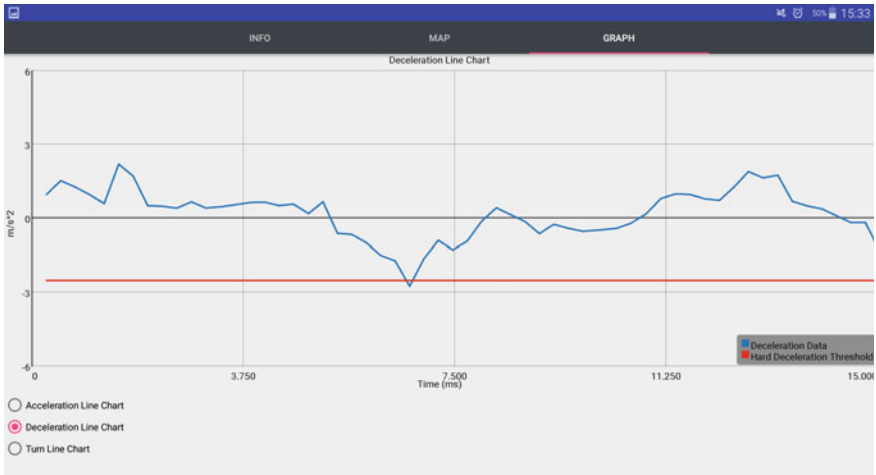


Fig. 27 Deceleration line chart of graph Tab screen of our application

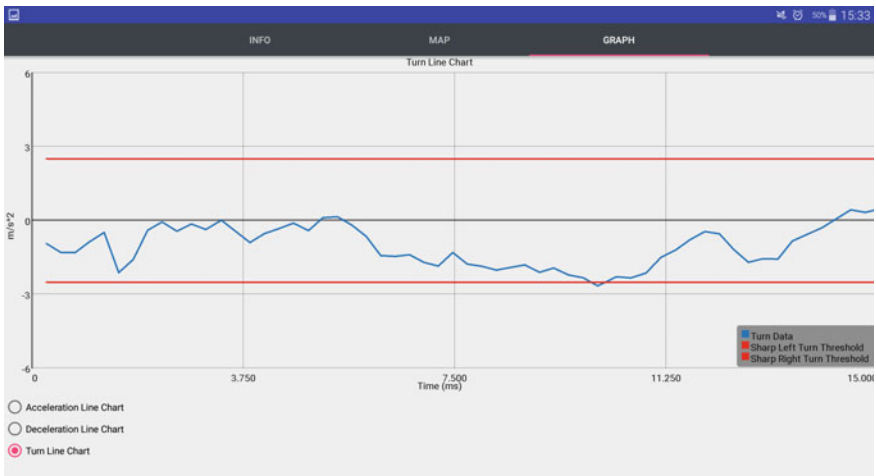


Fig. 28 Turn line chart of graph Tab screen of our application

line with the threshold value for each event. If there is a value that has exceeded the threshold line it means that we have committed a dangerous driving event from the current chart (acceleration, deceleration, turn) at this particular time. In Figs. 26, 27 and 28 we can see an example of these 3 line charts.

6 Conclusions

This work shows that the user's driving behavior could be estimated based on acceleration data of the accelerometer sensor or based on orientation data of sensor fusion method. Sensor fusion method combines data from the accelerometer, geomagnetic field sensor and the gyroscope. The driver can choose the detection method from the settings of the application's main menu. We propose to drivers to use accelerometer data as their detection method. The reason is not that the sensor fusion data is not accurate or reliable but the fact that sensor fusion method is not the best option for the detection of sharp turns or sharp lane changes. When we use only the accelerometer, the algorithm detects sharp turns or sharp lane changes faster than when we use the orientation data of sensor fusion method. On the other hand when we use the sensor fusion as detection method the system detects faster the safe/hard acceleration or deceleration of the vehicle than when we use only the acceleration data.

In our future plans is to extend this work, so the app can be used as a tool offered by Usage-Based Insurance (UBI) providers. In UBI the driver's insurance discount depends on his average rating over his trips. So, better rating means highest discount for his insurance. Hence, the system rewards the user for being a safe driver and also provides him with feedback on his driving habits, making him a better and safer driver. In such a case, using the sensor fusion method or the accelerometer method, it will be useful our system to take into account that sometimes hard decelerations and accelerations or other dangerous driving events are necessary in order to avoid a collision. Hence, the app should be extended to identify patterns in driving habits, so an occasional hard brake, for example, has not a significant impact on the potential rating of the driver.

References

1. Accelerometer, <http://en.wikipedia.org/wiki/Accelerometer>
2. GPS, http://en.wikipedia.org/wiki/GPS_navigation_device
3. Gyroscope, <http://en.wikipedia.org/wiki/Gyroscope>
4. Magnetometer, <http://en.wikipedia.org/wiki/Magnetometer>
5. Singh, P., Juneja, N., Kapoor, S.: Using mobile phone sensors to detect driving behavior. In: Proceedings of the 3rd ACM Symposium on Computing for Development, ACM (2013)
6. Fazeen, M., Gozick, B., Dantu, R., Bhukhiya, M., Gonzalez, M.C.: Safe Driving Using Mobile Phones. In: IEEE Transactions on Intelligent Transportation Systems (2012)
7. Chigurupa, S., Polavarap, S., Kancherla, Y., Nikhath, K.A.: Integrated Computing System for measuring Driver Safety Index. In: International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 2 (2012)
8. Dai, J., Tang, J., Bai, X., Shen, Z., Xuan, D.: Mobile phone based drunk driving detection. In: Proc. 4th Int. Conf. Pervasive Health NO PERMISSIONS, pp. 18 (2010)
9. Johnson, D.A., Trivedi, M.M.: Driving Style Recognition using a smartphone as a sensor platform. In: IEEE 14th International Conference on Intelligent Transportation system, October (2011)

10. H. Eren, S. Makinist, E. Akin, A. Yilmaz: Estimating Driving Behavior by a smartphone. In: Intelligent Vehicles Symposium, Alcalá de Henares, Spain, June (2012)
11. Chalermponl Saiprasent and Wasan Pattara-Atikom: Smartphone Enabled Dangerous Driving Report System. In: 46th Hawaii International Conference on System Sciences (2013)
12. Chuang-Wen You, Martga Montes-de-Oca, Thomas J. Bao, Nicholas D. Lane, Hong Lu, Giuseppe Cardone, Lorenzo Torresani, Andrew T. Campbell: CarSafe: A Driver Safety App that Detects Dangerous Driving Behavior using Dual – Cameras on Smartphones. In: UbiComp12, Pittsburg, USA, September (2012)
13. Fadi Aloul, Imran Zualkernan, Ruba Abu-Salma, Humaid Al-Ali, May Al-Merri: iBump: Smartphone Application to Detect Car Accidents. In: IAICT, Bali 28–30 August 2014
14. Nidhi Kalra, Gunjan Chugh, Divya Bansal: Analyzing Driving and Road Events via Smartphone. In: International Journal Of Computer Applications No. 12, July 2014
15. Jin-Hyuk Hong, Ben Margines, Anind K. Dey: A smartphone-based sensing platform to Model Aggressive Driving Behaviors. In: CHI, Toronto, Canada (2014)
16. Johannes Paefgen, Flavius Kehr, Yudan Zhai, Florian Michahelles: Driving Behavior Analysis with Smartphones: Insights from a controlled Field Study. In: MUM'12, ULM, Germany (2012)
17. V. Corcoba Magana, M. Munoz-Organero. Artemisa: An eco-driving assistant for Android Os. In *IEEE International Conference on Consumer Electronics - Berlin (ICCE-Berlin), 2011*, pages 211–215, 2011.
18. R. Araujo, A. Igreja, R. de Castro, and R.E. Araujo. Driving coach: A smartphone application to evaluate driving efficient patterns. In *2012 IEEE on Intelligent Vehicles Symposium (IV)*, pages 1005–1010, 2012.
19. Y.L. Murphey, R. Milton, and L. Kiliaris. Driver's style classification using jerk analysis. In *IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems, 2009. CIVVS'09*, pages 23–28, 2009.
20. Fr. Hørtvedt, Fr. Kvitvik, and J. A. Myrland. DriSMo - the driving quality application. Bachelor thesis, Gjøvik University College, May 2011.
21. Atan2, <https://en.wikipedia.org/wiki/Atan2>
22. Radoslav Stoichkov, Android Smartphone Application for Driving Style Recognition, Department of Electrical Engineering and Information Technology Institute for Media Technology, July 2013.
23. P. Lawitzki. Application of Dynamic Binaural Signals in Acoustic Games. Master's thesis, Hochschule der Medien Stuttgart, 2012.
24. P. Lawitzki, Android Sensor Fusion Tutorial, <http://plaw.info/2012/03/android-sensor-fusion-tutorial/>
25. ExponentialMovingAverage, https://en.wikipedia.org/wiki/Moving_average

Part IV
New Horizons: Large Scenarios

Cloud Platforms for IoE Healthcare Context Awareness and Knowledge Sharing

Alireza Manashty and Janet Light Thompson

Abstract Due to the growing in elderly population, research in healthcare monitoring and ambient assisted living technology is crucial to provide improved care while at the same time contain healthcare costs. Although the number of healthcare sensors are increasing as part of the Internet of everything growth, there is no robust system so as to act as a bridge between different sensors and systems to facilitate knowledge sharing and empower their detection and prediction capabilities. These systems cannot use the data and knowledge of other similar systems due to the complexity involved in sharing data between them. Storing the information is also a challenge due to a high volume of sensor data generated by each sensor. However, state-of-the-art cloud platforms can provide services to developers to leverage the previously processed similar data and the corresponding detected symptoms. Cloud-based platforms such as HEAL and CoCaMAAL can provide services for input sensors, Internet of Everything devices and processes, and context providers all at the same time. The ultimate goal of these systems is to bridge the gap between symptoms and diagnosis trend data in order to accurately and quickly predict health anomalies.

1 Introduction

Population ageing, the phenomenon by which older people become a proportionally larger share of the total population, is occurring throughout the world. World-wide, the share of older people (aged 60 years or older) increased from 9 % in 1994 to 12 % in 2014, and is expected to reach 21 % by 2050 [1]. Due to technological advancements, older people are also safer and live longer. This ageing population will create many challenges for society and health care systems such as increase in

A. Manashty (✉) · J.L. Thompson
University of New Brunswick, Saint John, Canada
e-mail: a.manashty@unb.ca

J.L. Thompson
e-mail: jlight@unb.ca

diseases, healthcare costs, and shortage of caregivers. Thus, systems and processes are needed that will help managing the health demands of this population. One such solution known as ambient intelligent systems, may provide the answer. Ambient intelligent systems render their service in a sensitive and responsive way and are unobtrusively integrated into our daily environment [2, 3]. Similarly, ambient assisted living (AAL) has become a popular topic of research in recent years. AAL tools such as medication management tools and medication reminders allow the older adults to take control of their health conditions [4, 5]. Usually, an AAL system consists of smart sensors, user apps, actuators, wireless networks, wearable devices, and software services which provides real-time physical and medical information of the patient [6]. However, when a higher level of mined data is required for how it affects the life of the patients, an AAL system is inadequate because it cannot provide the necessary prediction and intelligence for analysis.

Internet of everything (IoE) which consists of not only sensors, but people and processes as well, can create a bigger picture of the daily data that is being recorded by AAL systems. In AAL, most of the data are collected from sensors, video, cameras etc. at the low-level. The result for processing systems is a very diverse collection of different types and formats of data. Processing and aggregation of these data is a major challenge, especially when analyzing in real-time large streams of physiological data, such as electroencephalogram (EEG) and electrocardiogram (ECG). An efficient system depends on improved hardware and software support [7]. Cloud computing and IoE devices are two endpoint technologies that can support the above challenge of remote healthcare and data processing.

IoE can address the problems of interconnectivity between patients, physicians and the ambient devices helping the care-receiver. AAL devices (such as laptops, smartphones, on board computers, medical sensors, medical belts and wristbands, household appliances, intelligent buildings, wireless sensor networks, ambient devices, and RFID tagged objects) are identifiable, readable, recognizable, addressable and even controllable via the IoE [8]. The enormous amount of information produced by them, if processed and aggregated, can help in solving long-term problems and can immediately predict emergencies. Of course there are some challenges when dealing with a large amount of heterogeneous patient data.

Patient's physiological data varies with different activities, age and it varies from one individual to another. In order to process such data and to aggregate it efficiently with other available data sources, a very large memory space and high computing power are required. A comprehensive system requires a complete knowledge repository and must remain context sensitive to satisfy different behavior profiles based on an individual's specialized needs. But performing such a massive task on a centralized model and location is failure prone and slow [9]. Cloud-based and distributed frameworks are more easily scalable and accessible from anywhere especially when combined with IoE devices.

Several systems and middleware are proposed to address AAL data aggregation, processing, detection and even prediction [9–14]. Most of these systems are only tested in limited simulated areas and the data and techniques are not actually used and leveraged by the elderly in the way they require. Different systems have

proposed different architectures of storing, processing, aggregation and decision making. The problem identified in all of the above systems is the absence of a single platform that could act as a middleware for such systems to provide services that all developers and healthcare systems can use to share trends, detection and prediction knowledge among each other.

In this chapter, some of the state-of-the-art approaches to create a framework or platform is discussed. Such a framework can act as a middleware between processed raw data and trends and predicting knowledge. These systems are not only useful for the data provider itself, but also for other systems that might lack the necessary historical knowledge required to successfully detect and predict the unforeseen anomalies.

This chapter continues with providing background knowledge and then discusses challenges in creating such systems. Introducing existing frameworks comes next and then a proposed HEAL model that seeks to act as a bridge between different platforms is provided in detail. This platform provides services not only for sensors and third-parties, but also tools for developers to leverage previously processed similar data and the corresponding detected symptoms. The proposed architecture is based on cloud and provides services for input sensors, IoE devices, processes and people, and context providers. Restful services for developers of other systems are provided as well. A prototype of the model is implemented and tested on a Microsoft Azure cloud platform.

2 Background

Data fusion and integration is the first step towards gaining valuable knowledge from multiple sources of data (i.e., sensors). Multisensor data fusion and data aggregation techniques are used to integrate data from different sensors. Some background on these topics help us understand the nature of problems and possible solutions for leveraging sensor data in IoE context.

2.1 Data Fusion

Data fusion techniques are the methods and algorithms used to aggregate the data from two or more sensors. Also called multisensor and sensor data fusion, there are several levels of techniques when dealing with either low-level or high-level sensor data. *Low-level data fusion* often deals with the raw input of sensors and the techniques used to process and cleanse the imperfect input data. *High-level data fusion* techniques are often needed to retrieve meaningful information from input sensors. Figure 1 shows the basic JDL model for sensor fusion to which addresses the different sensor levels. When dealing with raw sensor data, the process always

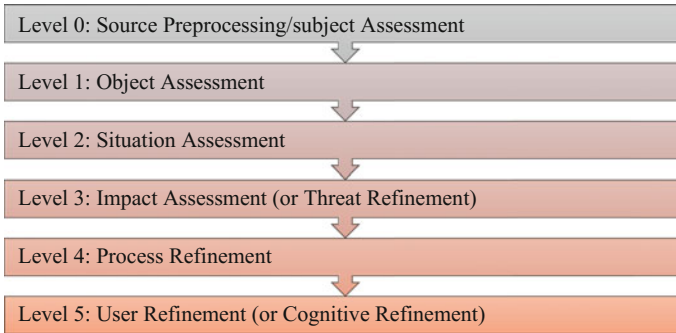


Fig. 1 JDL model levels

starts at level zero. There are many processing steps that should be applied to the raw sensor data at each step.

Depending on the input sensor data quality, sensor fusion algorithms should be able to deal with imperfect, correlated, inconsistent, and/or disparate data [15]. At higher levels of data fusion, when objects and high-level information are acquired, data cleansing algorithms, such as duplicate removal, are widely used. At the highest levels of sensor fusion, events are detected and extracted from the fused sensor data. For example, in a system which detects a heart attack, the input sensor data are binary bits from different wired and wireless devices such as ECG,¹ EEG, oxygen sensor, heart rate monitor, and probably pixels from a 2-D or 3D (Time-of-flight) video camera. At the higher levels, the system is expected to detect anomalies from each device. At the highest levels, events that can only be detected with fusing multiple sensor data are detected and reported as the output of the system.

When dealing with IoE sensors, most of the times multisensor data fusion is required and applied to the input sensors. Then the higher level data fusion is applied to the events reported in the previous steps. Finally, events, usually along with location, define the current context in which a device or person is. Context awareness is the key in autonomous control and AAL.

2.2 *Ambient Assisted Living (AAL)*

AAL technologies provide a complete set of services ranging from input sensors and context awareness to output actuators and third parties; all to support an individual's daily life. AAL systems can specifically assist people who need special monitoring and care, e.g., patients with Alzheimer (See Fig. 2). These systems can

¹Electrocardiography.

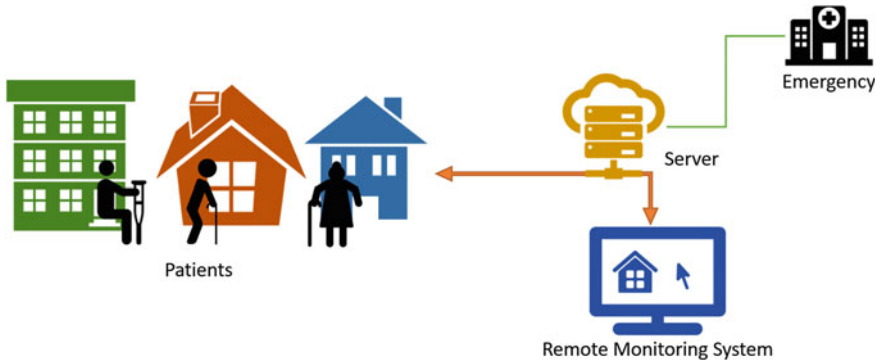


Fig. 2 How remote monitoring systems work in an AAL environment

monitor a patient’s daily activities and report any anomalies to caretakers or in a case of emergency, directly notify the emergency medical services (EMS). Although these systems can be effective in detecting and monitoring, they are usually not intelligent enough to predict events based on historical data. Thus, they can currently be considered as practical solutions for in-home patient monitoring and event detection; but there are still many challenges for event prediction.

3 Challenges

Healthcare monitoring is a major part of IoE, which targets to connect not only physical devices, but people and processes as well [16]. In this chapter, the focus is on outlining the technical challenges and discussing the possible solutions. Privacy in healthcare is also discussed briefly, however, healthcare privacy depends mainly on government legislations and corporate policies and thus requires a separate in-depth review. Therefore, context awareness and knowledge sharing will be discussed here as the main technological challenges towards an interconnected IoE healthcare platform.

3.1 Context Awareness

Intelligent systems capability to aid a person is maximized when they are context aware, i.e., information about the location and surroundings of the person’s is available. Knowing where the person is and what he/she is doing though requires using variety of sensors placed in different locations. In a home environment, for example, whether a person is brushing his teeth, washing his hands or simply looking at the mirror cannot be distinguished by simply using the location of the

person. If not using ID tags (e.g., NFC or RF) for context identification, then complex video processing (e.g., using RGB-D cameras) is required. All these help context aware systems to provide a better living environment by providing intelligent support and monitoring.

Adopting a context aware environment is often challenging for users. Having so many sensors around and especially those always carried by user (e.g., accelerometer sensors for fall detection) are not welcomed by users. Thus, non-invasive approaches are naturally more acceptable to users. Locating a user's location at home using video camera or floor sensors are non-invasive. Whereas carrying a belt or smart phone 24/7 can be quite challenging. These challenges make context aware adaptation much slower than anticipated.

3.2 Knowledge Sharing

Exchanging detection and prediction knowledge between monitoring systems is vital especially in dealing with rare anomaly events. Training data is the key to prediction and detection of events. An unknown event cannot be detected or predicted with a system which has no historical data about the sources or exposure with the event itself. In order for a system to predict an event, it must have prior information about the event.

Often, it is quite unlikely that a new system has information about a rare anomaly for a person, e.g., a heart attack. Nevertheless, this data can be available in another monitoring system capturing the data. Up to this point, we could not find any comprehensive system that can act as a link between two or more real-time health monitoring systems in order to share historical data. This knowledge sharing is valuable, as it can save lives. Especially in the spread of epidemic diseases, if there is no real-time knowledge exchange mechanism for sharing the symptoms of a new type of disease, the number of casualties may increase and disease containment would be slower. Solving this problem requires a new model and computing environment that can always be accessible for other monitoring systems.

Cloud computing platform-as-a-service (PaaS) can be used for solving this problem. Scalability and distributed design for both data sharing and computing can help solve this problem. Plus, most prediction algorithms and techniques are now vastly available in the cloud environment for further integration with other systems; making the cloud ecosystem suitable for this task.

3.3 Real-Time Decision Making

Accurate real-time decision making also requires dedicated computing power and historical knowledge. Wearable devices usually do not possess these, so a central processing system can help with complex prediction and classification



Fig. 3 How predicting future trends and anomalies require train data from past events

computations. In addition to complex processing, an always up-to-date knowledge base may be critical for time-critical situations, e.g., a fast-spreading epidemic disease and multiple data center failures. Thus, a cloud-based data warehouse, real-time data mining and decision making computing power can be critical even for the wearable sensor devices, people and processes in an IoE environment.

To achieve a reliable prediction capability, some previously seen anomalies and events are usually required (Fig. 3). The train data for accurate future prediction may not actually be present in the current system (e.g., a wide spread disease in another country with possible symptoms in a new country). Thus, data exchange and historical data storage and analysis is a necessary part of anomaly detection and prediction and thus real-time decision making.

3.4 Efficient Service Delivery

Most in-home care systems, such as Microsoft Health [17], IBM Watson Healthcare [18] CareLink Advantage [19], only report events and emergencies to specific family members and/or directly to the emergency units. This might result in either missing an emergency situation (due to unavailability of the caretaker) or overcrowding the emergency units with false alarms. Thus, intelligence plays an important role in IoE environments where every sensor, person, and operational process matters. Thus, such systems can make current remote monitoring systems smarter by providing preventative detection and prediction services (Fig. 4).

3.5 Comprehensive Monitoring System

Although many projects and systems have been proposed and implemented in different research centers and industries, most of them only work with specific equipment and in controlled scenarios. Not only do researchers have difficulty

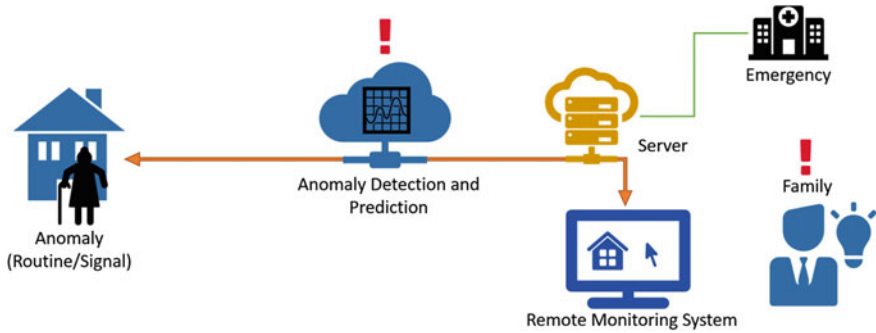


Fig. 4 Predicting an anomaly with the help of an intelligent detection and prediction system

accessing non-sensitive knowledge from such systems, but consumers also suffer from a lack of affordable home-care solutions. If there existed some comprehensive monitoring system standards, a competitive market for wearable devices and monitoring hardware could help lower the prices and increase the shared knowledge. This is the same way technologies like ZigBee could help grow home-monitoring technologies, so a standard comprehensive monitoring platform could also help join homogeneous sensors in a controlled IoE scenario.

4 Existing Frameworks

There are some network-based and cloud-based scenarios for AAL scenarios. OpenAAL [20] and UniversAAL [21], CoCAML [9], cloud prediction platforms and HEAL are some of the frameworks that have been developed recently to address the challenges explained earlier.

4.1 OpenAAL and UniversAAL

OpenAAL and its decedent universAAL have been implemented and tested in some real-world scenarios. OpenALL was a project supported by the European union which became part of universAAL in 2010. UniversAAL was a four-year project supported by the European union which is now continued by ReAAL [22] to implement the project in real environments. The outcome is the universAAL platform currently being piloted in 9 countries with +6000 users [22]. UniversAAL is context-aware, especially on location, and provides a network platform based on OSGIs. Nodes are called AAL Spaces and can communicate with each other (Fig. 5). There is also a Native Android version available for further development. This platform can be considered one of the most significant projects in the AAL

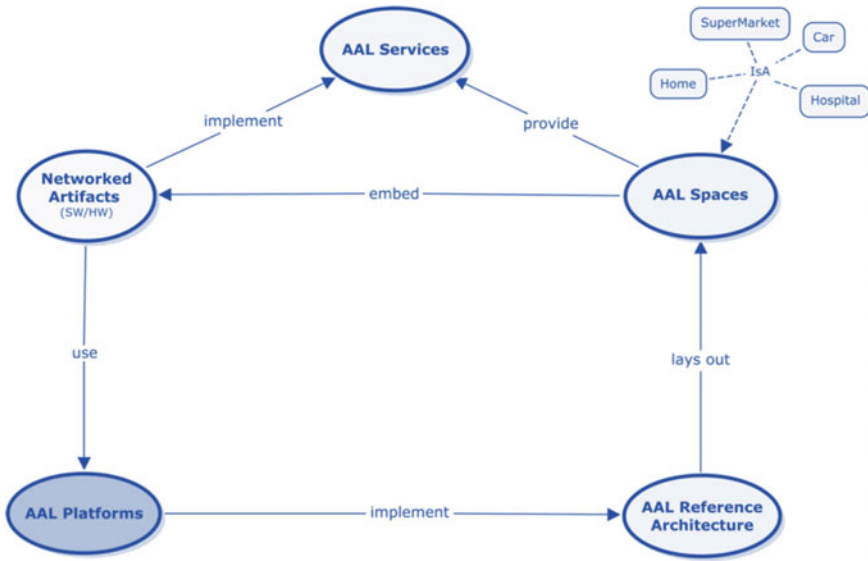


Fig. 5 AAL spaces and AAL platforms interaction

movement, especially in Europe. It can be a very good infrastructure or middleware, yet it is not providing a cloud-based platform for setting up an AAL Spaces and automatically communicate with AAL nodes (although it can be deployed on cloud, there are many possible challenges regarding its setup).

4.2 CoCAML

CoCAML is another cloud-based platform proposed by Forkan et al. The proposed platform is very detailed and its authors considered a variety of services, sensor interactions, and ontology modelings. The platform suggests the concept of *context providers* as high-level data providers. However, only some services deployed have been cloud-based and CoCAML has been only tested with simulation data. Yet, it lacks the notion of *predictors* for prediction and detection of anomalies. Forkan et al. proposed an anomaly prediction schema for AAL later [10], but it lacks generalization required to be used as part of a platform.

4.3 Cloud Prediction Platforms

Because of the rapid advancement of cloud platforms, cloud-service providers are now providing machine learning and prediction PaaS as part of their services.

Microsoft Azure Machine Learning [23] provides a machine learning platform capable of predicting missing information in a context. Most of the machine learning algorithms are implemented and available as drag and drop nodes in its online studio. Azure ML is the newest, amongst others; however, it provides an excellent user interface for the customization of prediction algorithms. It supports R language scripts which are used to manipulate the data and use several already implemented data mining functions. It also supports web services and input and output of each experiment.

Apache MLlib [24] and Google Prediction [25] are also available to provide prediction functionalities on the cloud with implemented libraries and scalable performance. These platforms can be used in conjunction with a health event aggregation platform to provide data mining and prediction anomaly services for an IoE environment. More detailed information on data analysis on the cloud can be found in the book by Talia et al. [26].

Azure IoT

Microsoft is also providing a complete package for IoE, with Azure IoT suite [27]. Combining Microsoft Azure's cloud services with Power BI's reporting and analysis capabilities, Microsoft IoT suite delivers everything from real-time sensor data ingestion and event processing to predicting analysis and online reporting.

Starting with a fleet management demo (Fig. 6), Microsoft shows how the current health status of a truck driver can be seen live in an app. Sensors send the information to the IoT suite; the sensor data goes through different Azure cloud services, including Event Hubs and Stream Analytics. Finally, the required event information reaches Power BI, which enables rich data visualization, especially on

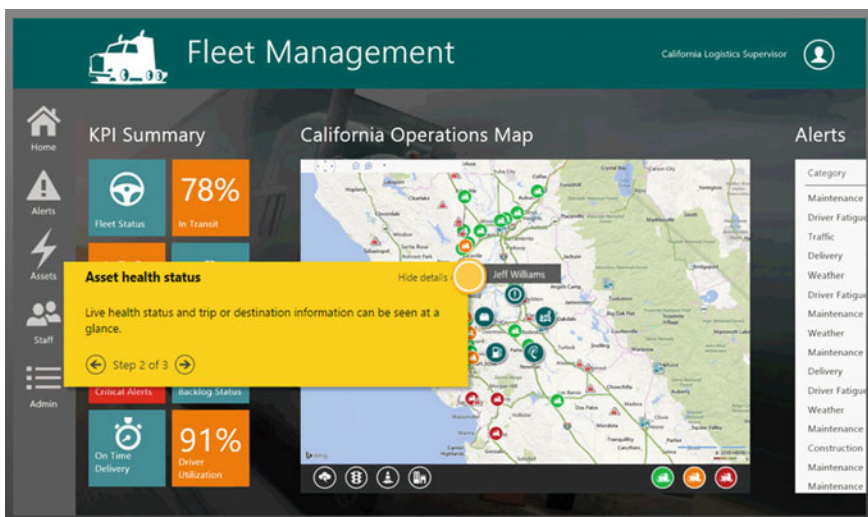


Fig. 6 Fleet management system demo utilizing Microsoft IoT suite

Bing map. This suit and demo can be beneficial in developing scalable cloud-based applications which include A to Z of an IoE monitoring platform.

5 Proposed Framework

To address the challenges and based on the previous frameworks, particularly the CoCAML cloud-based platform, Health Event Aggregation Lab (HEAL) platform is proposed as a cloud-based platform capable of real-time health event aggregation and anomaly prediction (Fig. 7).

5.1 Design

HEAL is proposed to address the challenges mentioned earlier. It is designed as a platform capable of gathering health-related event data, aggregating the related information and detecting or predicting anomalies in near real-time speed. It also helps facilitate the knowledge exchange between different sub-systems.

5.2 System Overview and Methodology

A context-aware knowledge-based framework is proposed here for any event anomaly detection and prediction. This framework makes it possible for third-party systems to provide high-level monitoring data as input and obtain detection and prediction services from the system. Users can work with the system to define their preferences regarding the input and output of the system. The proposed framework consists of the following 3 layers:

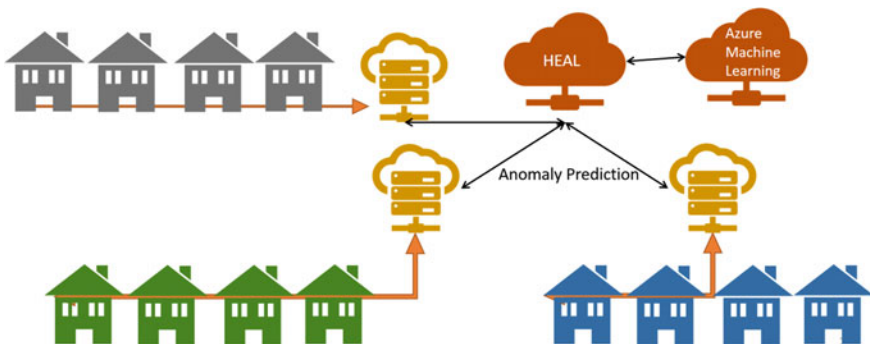


Fig. 7 An overview of HEAL framework

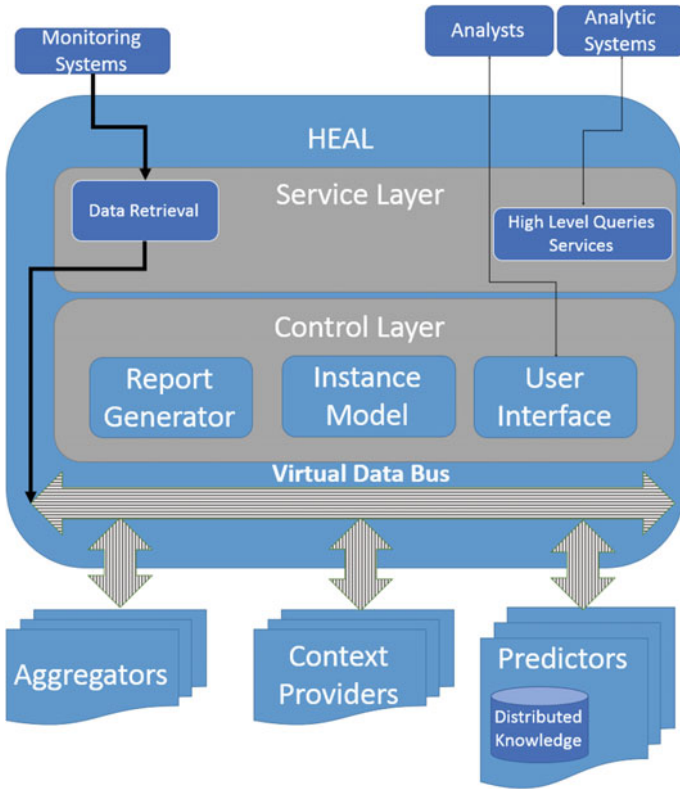


Fig. 8 Proposed cloud-based HEAL platform model

- A. Service Layer: This allows other systems to connect to the framework using REST APIs and SPARQL endpoints, allowing analytical systems and real-time data providers to access the system.
- B. Control Layer: In this layer, the user can control and customize the system and define the interconnection of different inputs and outputs in a model that is created for a user.
- C. Distributed Cloud-Based Data Providers: Based on the data provided in the Control Layer and the data received by monitoring systems, different aggregators can prepare the data for higher level processing by context providers. Based on the input provided by context providers, newly proposed predictor components can retrieve the historical data, save it and provide predictive parameters.

For the proposed distributed model in Fig. 8, two new components of aggregator and predictor are added. Context providers are already defined in CoCaMAAL model [9].

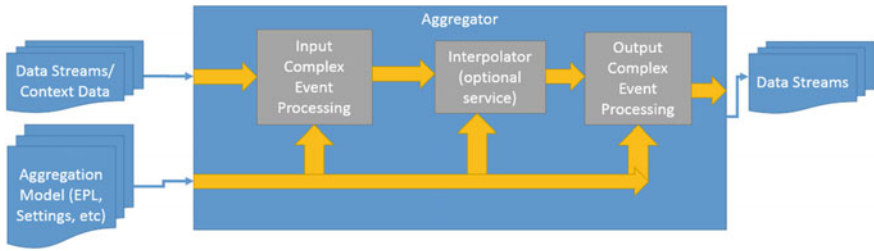


Fig. 9 Proposed aggregator model for HEAL

Aggregators

Aggregators are the bridge between the real-time streams of data from the monitoring systems or high-level streams of data from other parts of the system, including context providers and predictors. As shown in Fig. 9, these components will retrieve the data stream and use event processing language statements provided by the system user, create a different abstraction of the data, make it cleaner, more readable or more prepared for aggregation. In this level, many different formats and data rates are provided. The interpolator component interpolates missing data to increase the data rate so that data stream can easily be aggregated with other data streams. In the final step, the user has another opportunity to define more specific data aggregation statements for the final output of the component.

Predictors

Predictor is another novel component proposed for this model (Fig. 10). In these distributed cloud-based components, data from a specific duration of time or sequence are provided to the predictor as input. The predictor then stores the data in its data warehouse (which is managed by the predictor itself) and then using the prediction engine specified for its purpose will create a prediction model to interpolate or extrapolate the data. The system can then query the predictor to get future

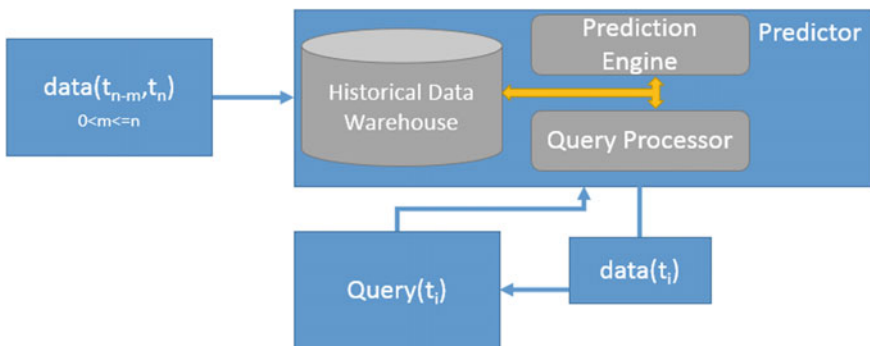


Fig. 10 Proposed predictor model for HEAL platform

data, prediction errors, or possible trends. Having separate distributed predictors help third parties and system analysts share different prediction engines and have specific data warehouse for their data. Some of the powerful current prediction engines are Google Prediction and PredictionIO.

Other Components

- **Monitoring system:** Monitoring system consists of various sensing devices; they collect raw data and send to the upper layer. Sensing devices can be EEG, ECG, electromyography (EMG), accelerometer, fall detector, magnetometer, gyroscope, motion sensor, blood pressure device, blood sugar sensing etc. These sensors together form a Body Sensor Network [28]. Each of these sensing devices works on low power and has capability to transmit the data wirelessly to upper layer in the cloud. Set up of the sensing devices varies individually. Sensors can easily be added or removed from the system without affecting the overall performance.
- **Data Retrieval:** In this layer raw data are directed to the specific aggregator for the event retrieval and more processing.
- **Complex Event Processing:** In this layer of the system, all the incoming real time high-level signals are passed through the high-level complex event processing language such as Esper and NEsper to detect anomalies in the high-level data.
- **Cloud-based historical data ware-house:** All the events, data and the information about the anomalies are saved in the data ware-house for the future purpose. This data necessary for predictors to predict future trends and anomalies and for setting the threshold for the various vital signs for a person.
- **High-level query services:** Access endpoint for the analytic systems with REST and SPARQL endpoints.

5.3 System Implementation and Prototype Development

Most subsystems of the proposed platform are implemented and tested with experimental data. A cloud framework is designed and several applications have been deployed as separate modules on Microsoft Azure. A cloud-based website is also used as the front end to test the platform.

Also, event stream processing using NEsper is used to process real-time signals in predictors. The system itself is tested with a wink detection EEG dataset as a case study for real-time eye wink detection. This test is designed to evaluate the real-time data transfer and processing performance of the system.

To send the recorded EEG data in real-time to the cloud-based server, a Raspberry Pie 2 model B+ is used. Windows 10 IoT core is installed as OS to run the

application on the Raspberry Pie 2, which is a good representative of biometric devices with limited processing power and resources. The system is then tested with 3 running applications on Raspberry Pie 2, sending real-time EEG signals to the Microsoft Azure Event Hub every 100 ms. Event Hub is a real-time event ingestor service that provides event and telemetry ingress to the cloud at massive scale (millions of events per second), with low latency and high reliability [29]. Each event hub partition can handle 1 MB ingress and 2 MB egress per second. Using default 16 partitions, our instance of Event Hub can handle 16,384 messages of size 1 KB per second. The events are then consumed by an instance of Stream Analytics, which is a fully managed, real-time stream computation service hosted in Azure providing highly resilient, low-latency, and scalable complex event processing. It also helps developers to combine streams of data with historic records. Combined with Event Hubs, Stream Analytics is capable of handling high event throughput of up to 1 GB/s [30]. The real-time system test indicated immediate transfer of information from Raspberry Pie to the Stream Analysis. The final analysis results and detected anomalies are then pushed to the web page Javascript client using SignalR instantly (~1 s).

For anomaly detection we have used Tim Van Kastern's public datasets. It has data from three different houses [31] with recorded start time, end time of the activity and type of activity. Sensor Output is binary and represented in a feature space which is used by the model to recognize the activities performed. Using hidden Markov model to predict the anomalies based on Forkan's approach [10], the system could produce similar results in the above real-time context with the data being sent from the Raspberry Pie device.

5.4 *Testing and Evaluation*

The system functionality should also be tested against similar systems. To test the system in terms of accuracy and speed, the system will be deployed to Microsoft Azure Cloud and will be tested as follows.

One of the main challenges to test IoE systems is the test and train data. Stress testing and load testing requires a large amount of data and simultaneous IoT device connecting to the server. The testing problem consists of two main parts: 1. Data 2. Simultaneous connections.

The data required to test the system can come from several places. Ideally, real data should be collected, analyzed and a clean dataset should be created for training and testing purposes. In reality, it is very challenging to record and generate the dataset required for testing. To test the system for accuracy and precision, the data can also either be simulated, or third-party datasets be used.

Testing the system behavior when under heavy load, requires multiple IoT devices to test in practice. To test the system, simultaneous client applications should be created and run on a set of machines to simulate a large number of

sensory devices. When the system is tested successfully, the system will be ready for initial launch.

Finally, the implemented model is to be evaluated as a solution to the challenges mentioned earlier. After deployment to the cloud, the user experience (UX) should be evaluated using surveys sent to both developers and business analysts. This UX should be then compared to similar systems with the same functionality (e.g., universALL and CoCAML).

6 Potential Complications

6.1 Policies, Privacy, and Trust

Government policies are quite strict when dealing with privacy and information exchange. The Personal Information Protection and Electronic Documents Act (PIPEDA) in Canada, for example, creates rules for how private sector organizations may collect, use, and disclose personal information. The law gives individuals the right to access their personal information and governs businesses for sharing information for commercial activities. Although such legislations can protect personal information, they can limit the access to necessary healthcare sensor data that is required to provide further analysis on patients' data.

Even if companies can access personal health information, protecting the information can become another issue. Security measures should be taken to protect the information from external intrusions. Thus, security in every secondary site in which the personal information can be accessed is as important as the primary information site.

Both policies and security measures aim to build trust in users' mind. However, there are still concerns about the application of the personal information. In particular, whether the information retrieved is to be used in favor or against the individual is still a concern. For example, insurance companies are interested in placing premiums based on the current health and possible predictions of user's health status. On the other hand, similar predictions can help patients prevent diseases.

6.2 Security

Securing the data from unauthorized users should be a top priority from the IoT devices to the cloud server and further into the front end. Unauthorized access to personal medical data has severe security consequences, both for the company and the user. Thus, data transmission should be secured and users should be authenticated and authorized.

To ensure data privacy, network messages between IoE services and devices should never travel unencrypted. Depending on the type of the service, specific

message and transport security algorithms are available. Secure socket layer (SSL) can be used to secure most common RestAPI communications via HTTPS. However, the IoE devices require more powerful processors and should be able to update the encryption algorithms as they become obsolete. As this is not possible in most cases on the device side, message security can easily become obsolete due to lack of upgradability in most IoT sensor devices.

Devices and users accessing a centralized IoE server should be authenticated and authorized. Security tokens are widely used to authenticate each request to the server. Bearer tokens enable authentication in each request and expire after a specific time to enable full authentication. After authentication, a role-based authorization enables several levels of access to the system. Authorization in an IoE system enables devices and people to interact with a single system, accessing different layers of secured information.

6.3 Scalability

Regarding scalability, when the need arises for higher processing power, storage or network bandwidth, dedicated servers are not easy to upgrade. Especially for real-time services, it is critical for a system to be able to scale up without interruption. Cloud services are usually capable of scalability. The performance of the system can be increased without the extensive need for planning ahead for data migration and shutting down services during the process. Thus, due to the changing nature of real-time event aggregation, a cloud platform with scalability capabilities is required for IoE and in this case, HEAL.

IoE devices and processes require a 24/7 available backend. One of the main benefits of cloud-based platforms is the already enabled redundancy (also available geo-redundancy) and high reliability. In the case of a primary system failure, the backup system automatically receives and processes the requests. In a large-scale system, this can be critical as even seconds of failure can cost losing millions of messages. Therefore, the importance of reliability and availability of the backend servers should be considered in healthcare IoE applications.

7 Research Trends in IoE Knowledge Sharing Platforms

The platforms discussed in this chapter are the state of the art in IoE cloud computing and have not yet been adopted and used in practice. Testing such platforms in real scenarios require a variety of sensors and processes already in place. Thus, future research that can test different case studies using these platforms can determine their strength and weaknesses. Future models can be then designed to overcome the possible flaws.

Interconnecting different systems of sensors in IoE may infringe some policies or lead to conflict of interest between engaged companies. Research on the effects of these policies on the performance and scalability of IoE cloud platforms can reveal limitations of these systems in practice. Also, suggestions to change policies can facilitate the operation of these systems.

8 Summary

In this chapter, challenges towards designing healthcare knowledge sharing platforms, such as context awareness, knowledge sharing, real-time decision making, efficient service delivery, and the need for a comprehensive monitoring system are discussed. Some of the efforts to address this challenges as a framework are then introduced, such as OpenAAL, universALL, CoCAML, and the state-of-the art cloud prediction platforms. Then to address these challenges, the HEAL framework is proposed which tries to act as a bridge between different monitoring systems. In all these platforms, there are still some possible concerns regarding policies, privacy, security, and scalability which should always be considered in designing and developing these systems. Finally, it is expected that future research trends cover some of the mentioned challenges by developing and testing IoE knowledge exchange frameworks in real-world scenarios.

References

1. UN, *The World Population Situation in 2014*, New York, 2014.
2. M.S. Emile Aarts, Rick Harwig, *invisible Future*, Ambient Intelligence, 2001, pp. 235–240.
3. E. Aarts, *Ambient Intelligence: A Multimedia Perspective*, IEEE Multimedia, 11, 2004, pp. 12–14. doi:[10.1109/MMUL.2004.1261101](https://doi.org/10.1109/MMUL.2004.1261101).
4. I. Qudah, P. Leijdekkers, V. Gay, *Using mobile phones to improve medication compliance and awareness for cardiac patients*, Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments - PETRA'10, 2010, pp. 1. doi:[10.1145/1839294.1839337](https://doi.org/10.1145/1839294.1839337).
5. K. a. Siek, D.U. Khan, S.E. Ross, L.M. Haverhals, J. Meyers, S.R. Cali, *Designing a personal health application for older adults to manage medications: A comprehensive case study*, Journal of Medical Systems, 35, 2011, pp. 1099–1121. doi:[10.1007/s10916-011-9719-9](https://doi.org/10.1007/s10916-011-9719-9).
6. F. Sufi, I. Khalil, Z. Tari, *A cardiod based technique to identify Cardiovascular Diseases using mobile phones and body sensors*, in: 2010 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC'10, 2010: pp. 5500–5503. doi:[10.1109/IEMBS.2010.5626578](https://doi.org/10.1109/IEMBS.2010.5626578).
7. P. Remagnino, G.L. Foresti, *Ambient intelligence: A new multidisciplinary paradigm*, IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans. 35, 2005, pp. 1–6. doi:[10.1109/TSMCA.2004.838456](https://doi.org/10.1109/TSMCA.2004.838456).
8. J. Cubo, A. Nieto, E. Pimentel, *A Cloud-Based Internet of Things Platform for Ambient Assisted Living*, 2014. doi:[10.3390/s140814070](https://doi.org/10.3390/s140814070).

9. A. Forkan, I. Khalil, Z. Tari, CoCaMAAL: A cloud-oriented context-aware middleware in ambient assisted living, *Future Generation Computer Systems*, 35, 2014, pp. 114–127. doi:[10.1016/j.future.2013.07.009](https://doi.org/10.1016/j.future.2013.07.009).
10. A.R.M. Forkan, I. Khalil, Z. Tari, S. Fofou, A. Bouras, A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living, *Pattern Recognition*, 48, 2014, pp. 628–641. doi:[10.1016/j.patcog.2014.07.007](https://doi.org/10.1016/j.patcog.2014.07.007).
11. A. Copetti, J.C.B. Leite, O. Loques, M.F. Neves, A decision-making mechanism for context inference in pervasive healthcare environments, *Decision Support Systems*, 55, 2013, pp. 528–537. doi:[10.1016/j.dss.2012.10.010](https://doi.org/10.1016/j.dss.2012.10.010).
12. WONGPATIKASEREE, High Performance Activity Recognition Framework for Ambient Assisted Living in the Home Network Environment, 2013.
13. Y. Xu, P. Wolf, N. Stojanovic, H.-J. Happel, Semantic-based Complex Event Processing in the AAL Domain Semantic-based Event Processing in AAL, 9th International Semantic Web Conference (ISWC2010), 2010.
14. A. Zafeiropoulos, N. Konstantinou, S. Arkoulis, D.E. Spanos, N. Mitrou, A semantic-based architecture for sensor data fusion, *Proceedings - The 2nd International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBIComm 2008*, 2008, pp. 116–121. doi:[10.1109/UBICOMM.2008.67](https://doi.org/10.1109/UBICOMM.2008.67).
15. B. Khaleghi, A. Khamis, F.O. Karray, S.N. Razavi, Multisensor data fusion: A review of the state-of-the-art, *Information Fusion*, 14, 2013, pp. 28–44. doi:[10.1016/j.inffus.2011.08.001](https://doi.org/10.1016/j.inffus.2011.08.001).
16. D. Evans, *The Internet of Everything - How More Relevant and Valuable Connections Will Change the World*, CISCO Internet Business Solution Group (IBSG), 2012, pp. 1–9.
17. Microsoft, Microsoft Health.
18. IBM Inc., IBM Watson Healthcare. Available at (accessed 04/22/2016): <http://www.ibm.com/smarterplanet/us/en/ibmwatson/health/>.
19. Northern Communications Services, CareLink. Available at (accessed 10/06/2016): <https://carelinkadvantage.ca/>.
20. P. Wolf, A. Schmidt, J.P. Otte, M. Klein, S. Rollwage, B. König-Ries, T. Dettborn, A. Gabdulhakova, openAAL - The Open Source Middleware for Ambient Assisted Living (AAL), AALIANCE Conference, 2010, pp. 1–5. doi:[10.1029/2006GL026143](https://doi.org/10.1029/2006GL026143).
21. S. Hanke, C. Mayer, O. Hoeffberger, H. Boos, R. Wichert, M.-R. Tazari, P. Wolf, F. Furfari, universAAL – An Open and Consolidated AAL Platform, in: R. Wichert, B. Eberhardt (Eds.), *Springer Berlin Heidelberg*, Berlin, Heidelberg, 2011: pp. 127–140. doi:[10.1007/978-3-642-18167-2_10](https://doi.org/10.1007/978-3-642-18167-2_10).
22. M.R. Tazari, ReAAL, 2013. Available at (accessed 04/22/2016): <http://www.cip-reaal.eu/home/>
23. Microsoft Corporation, Microsoft Azure Machine Learning. Available at (accessed 04/22/2016): <https://azure.microsoft.com/en-us/services/machine-learning/>.
24. Apache, Apache Spark MLlib. Available at (accessed 04/22/2016): <http://spark.apache.org/mllib/>.
25. Google Inc., Google Prediction API. Available at (accessed 04/22/2016): <https://cloud.google.com/prediction/>.
26. D. Talia, P. Trunfio, F. Marozzo, *Data Analysis in the Cloud*, Elsevier, 2016. doi:[10.1016/B978-0-12-802881-0.00006-8](https://doi.org/10.1016/B978-0-12-802881-0.00006-8).
27. Microsoft IoT Demo. Available at (accessed 02/09/2016): <http://www.microsoftazureiotsuite.com/demos/remotemonitoring>.
28. G. Fortino, R. Giannantonio, R. Gravina, P. Kuryloski, R. Jafari, Enabling effective programming and flexible management of efficient body sensor network applications, *IEEE Transactions on Human-Machine Systems*, 43, 2013, pp. 115–133. doi:[10.1109/TSMCC.2012.2215852](https://doi.org/10.1109/TSMCC.2012.2215852).

29. Event Hub. <http://azure.microsoft.com/en-us/services/event-hubs/>.
30. Exploring Microservices in Docker and Microsoft Azure. <https://www.microsoftvirtualacademy.com/en-us/training-courses/exploring-microservices-in-docker-and-microsoft-azure-11796>.
31. T. Van Kasteren, G. Englebienne, B. Kröse, Activity Recognition using semi-Markov Models on Real World Smart Home Data Sets, *Journal of Ambient Intelligence and Smart Environment*, 2, 2010.

Survey on Technologies for Enabling Real-Time Communication in the Web of Things

Piotr Krawiec, Maciej Sosnowski, Jordi Mongay Batalla, Constandinos X. Mavromoustakis, George Mastorakis and Evangelos Pallis

Abstract The Web of Things (WoT) can be considered as a step towards the Internet of Everything development. The concept of WoT assumes that objects of the Internet of Things (IoT) seamlessly interact with the Web by re-using web protocols wherever possible. One of the most desirable service in the WoT is real-time communication, due to the event-driven character of many IoT applications. This chapter provides an overview of the protocols which are taken into account in order to ensure real-time interaction of WoT objects. We describe two technologies: WebSocket and WebRTC, which are a part of HTML5 specification and are considered as solutions that bring real-time communication capabilities into the WoT. CoAP, a specialized protocol for use in resource constrained devices, is also presented, as well as two solutions that implement publish/subscribe interaction model. Next, we discuss which protocols can have the greatest impact on the WoT development.

P. Krawiec (✉) · M. Sosnowski · J.M. Batalla
National Institute of Telecommunications, Warsaw, Poland
e-mail: P.Krawiec@itl.waw.pl

M. Sosnowski
e-mail: M.Sosnowski3@itl.waw.pl

P. Krawiec · M. Sosnowski · J.M. Batalla
Warsaw University of Technology, Warsaw, Poland

C.X. Mavromoustakis
University of Nicosia, Nicosia, Cyprus
e-mail: mavromoustakis.c@unic.ac.cy

G. Mastorakis · E. Pallis
Technological Educational Institute of Crete, Crete, Greece
e-mail: gmastorakis@staff.teicrete.gr

1 Introduction

The next, 5th generation of mobile system (5G) is being designed to meet the requirements of serving billions of low-cost wireless nodes, such as sensors, actuators or wearables. Consequently, it is anticipated that 5G will provide the foundation for widespread deployment of the Internet of Things (IoT) [1, 2]. Since mobile networks systematically transform to all-IP architecture, also 5G IoT will be based in general on reusing well-known IP stack wherever possible [3], ensuring unified, homogeneous communication between all entities at the transport layer [4, 5]. The similar trend has emerged also on the application level [6, 7]. Conjunction IoT objects and devices with well-established World Wide Web technologies (HTTP, HTML, XML, JSON etc.) alters IoT into the Web of Things (WoT) [8, 9].

The driving forces that have brought the leading role of the Web in many spheres of human activity (business, social, entertainment etc.) are web services. They represent the building blocks of web-based distributed applications, which have introduced innovations and added values in traditional business sectors as logistic or banking, as well as stimulated the emergence of new gain-revenue “e-*” markets such as e-commerce or e-entertainment. Web services, in accordance with Service Oriented Architecture (SOA), are offered to clients through unified interface, regardless of which technology was used for their implementation and who controls them. In this way, web services provide seamless communication and unified access to different functional entities within and across domains. They are a convenient tool for integration and control of various, distributed systems, which allows implementation of complex scenarios (for example: web application for advanced energy cost optimization which binds building automation system with virtual energy market place and distribution grid control).

Web of Things concept unifies interaction of IoT objects with web applications by reusing standard web protocols. Data from a huge number of different sensors and devices, interconnected with the Internet, will be presented on the Web and accessible for web services. This gives an opportunity to elevate web applications to a completely new level, by integrating the present and future web services with, derived from IoT devices, information about the physical world. In turn, WoT that brings IoT to the web services landscape, can be considered as a significant contributor to the Internet of Everything (IoE), which includes ubiquitous interactions of machines (i.e. IoT objects) and humans (i.e. final consumers of web services).

Many usage scenarios for IoT require that data, which are linked with detected changes in surrounding environment, should be almost instantaneously available for the use by web services. Consequently, there is a need for a real-time communication, at the application level, to ensure that information is disseminated between IoT objects and the Web without undue delay.

In this chapter we give an overview of existing, commonly used technologies for real-time communication in the Web, and analyze their suitability for Web of Things development. Our survey focuses only on solutions that can be applicable

for resource-constrained devices. Furthermore, we discuss the future development trends and forecast for WoT real-time services.

The chapter is structured as follows. In Sect. 2 we briefly introduce a concept and requirements for development of real-time applications in WoT domain. Section 3 presents the main enabling protocols for WoT real-time services. The next Section provides an outlook for future trends and research. Finally, in Sect. 5 we conclude our survey.

2 Real-Time Communication in WoT Domain

World Wide Web has come a long way during the last twenty years. The emergence of Web 2.0, the SOA concept and web services resulted in the Web transformation from a set of simple, static web-pages to the complex ecosystem, with powerful web applications influencing many fields of everyday life. Such transformation caused that original communication model based on transferring batched data from server to a client upon request, has become insufficient. Nowadays, many web applications require real-time communication capabilities, to forward a message to a client as soon as an appropriate data appears on the server side, without waiting for explicit request from the client. For example, web applications for supporting decision making process often rely on accurate information provided by different web components that perform real-time analytics (such components can be fed with data obtained from e.g. social media platforms). To cope with this issue, many solutions have been proposed in recent years to enable real-time communication in the Web, such as Ajax polling and long-polling techniques [10], WebSocket [11] or WebRTC [12], just to mention a few.

It should be noted that considering the Web domain, the term “real-time” does not refer to strict delay constraints, violation of which results in some critical errors that lead to the entire system failure. Such constraints, described as *hard* real-time [13], are characteristic for industrial automation sector where they are usually met by using dedicated infrastructure. In the case of web applications, we have in mind *soft* (a.k.a. *near*) real-time requirements. It means that there is no sharp deadline for data delivery. However, increasing delay in information transfer has a very negative impact on service quality perceived by the user.

Real-time communication becomes even more important when we consider IoT scenarios. IoT concept assumes on-line observation and manipulation of the surrounding environment through different entities (IoT objects) equipped with sensors and actuators. A user of an IoT application expects approximately the same reaction times as he/she experiences in physical world. When the user uses a smartphone to turn on a room light, the bulb should illuminate almost immediately, as it is when he/she uses a physical light switch. Likewise, an output data of sensors should be quickly transferred to IoT applications, because the user will not tolerate stale information about, usually rapidly changing, physical world. In complex scenarios,

which require closely synchronized control over multiple distributed actuators, real-time delivery technologies will be an absolute necessity.

The SOA design principle, the backbone of the current web development, is implemented nowadays in accordance with one of two major service-oriented architectures:

- Web Services that use remote procedure calls (RPC) for client-server interactions. This technology is commonly referred as *WS-** architecture. *WS-** relies on SOAP (Simple Object Access Protocol [14]) messages, which carry XML (eXtensible Markup Language [15]) payload and are exchanged between web components through HTTP-based transport mechanism. The primary idea of *WS-** service is to manipulate web resources using an arbitrary set of operations. *WS-** architecture is considered as a very well suited for enterprise IT systems [16].
- Representation State Transfer (REST) architecture [17]. It assumes that all entities in the Web are abstracted as resources, which are addressable and searchable via Uniform Resource Identifiers (URI). RESTful web services exploit HTTP as an application protocol and manipulate web resources using a uniform set of operations (HTTP's methods: GET, PUT, POST and DELETE). Each end-point is responsible for maintaining its state, what has a positive impact on overall system scalability and robustness.

Although dedicated DPWS (Devices Profile for Web Services) specification [18] was proposed to meet IoT requirements in *WS-** domain, the *WS-** web services seem to be less favored as the way for implementing Web of Things. The main reason is an extensive overhead (in terms of computational and communication resources) linked with XML-based SOAP messages handling [19]. Stateless, lightweight RESTful services are a preferable choice in the context of distributed IoT web applications [20, 21]. No need for maintaining states at server side is crucial if we consider a huge number of IoT objects that can be involved in many IoT scenarios. On the other hand, in REST concept each request carries all the information necessary for a receiver to understand the request and recognize the sender state. Therefore, resource identifiers and state descriptions should be carefully designed, taking into account limitations in processing power and memory capacity of IoT devices.

However, the REST supports point-to-point communication only, which is carried out between client and server. It is definitely not sufficient for IoT purposes. Using IoT web application the user, with a single click, should turn on multiple light sources in a room. On the other hand, a light source should comply with control messages obtained from several entities: users (via user interface), light sensors, presence detectors etc. Therefore, traditional REST architecture considered in the WoT context needs to be enhanced with another models of interactions: one-to-many and many-to-many. Such models can be implemented based on *publish/subscribe* communication pattern. It assumes that interacting entities act as a message publisher (i.e. data sender) or/and a subscriber (i.e. data receiver). The

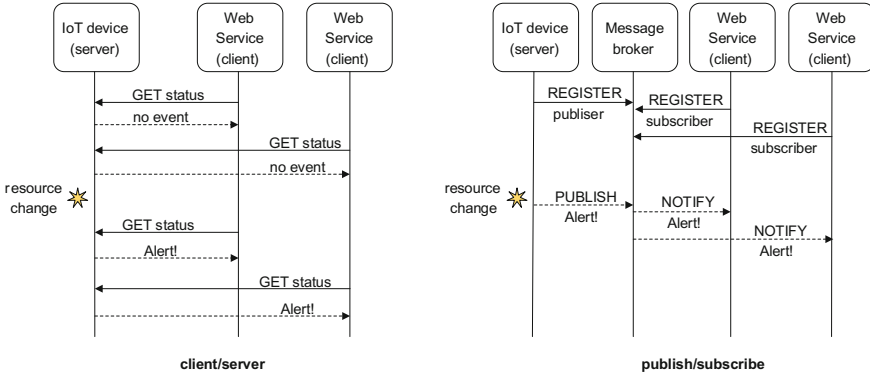


Fig. 1 Client/server (polling) versus publish/subscribe (push) interaction models

publisher sends a message without receiving direct requests from interested entities and even without explicit indication of message receivers. In turn, the subscribers are provided only with messages that they had earlier expressed interest in.

In comparison with the client/server model, publish/subscribe systems ensure high efficiency and scalability since IoT objects can push messages to announce any changes in their states just after detecting them, and there is no need to establish a separate connection with each involved receiver (as it is in the case of traditional polling—see Fig. 1). Publish/subscribe approach can be realized in two ways: (1) as a fully distributed system bases on multicast or peer-to-peer communication, or (2) with the support of an intermediate point, a *message broker*, which implements “store and forward” functionality. The broker is responsible for receiving messages from publishers and distributing them among registered subscribers.

3 Protocols for Real-Time Interactions with WoT Objects

The current Web takes advantage of different approaches to achieve real-time communication. However, some of them are too costly for resource-constrained IoT devices. For example, Ajax long-polling technique requires maintaining connections between IoT objects and remote peers at HTTP level, what is highly resource-consuming.

In this chapter we present an overview of protocols for real-time interactions proposed for WoT domain. The survey covers both the approaches that base on the current web technologies (WebSocket, WebRTC) and are compatible with REST principles that exploit client/server communication (CoAP), as well as the solutions that are in line with publish/subscribe paradigm (MQTT, AMQP).

3.1 WebSocket

WebSocket protocol was specified by IETF [11] in 2011 and afterwards incorporated by World Wide Web Consortium (W3C) into the latest HTML standard denoted as HTML5 [22]. It allows a client to establish a full-duplex connection with a remote web server by using single TCP socket.

The connection establishment is initiated by HTTP handshake procedure, during which the client provides to the server an HTTP Upgrade header to trigger protocol switching. Next, the WebSocket connection is created over already established TCP session that uses standard HTTP or HTTPS port number (80 or 443, respectively). In this way, WebSocket fits well to the environment where end-to-end communication is broken for non-HTTP traffic due to existence of firewalls, NATs or proxies.

Established connection is keeping open as long as needed, allowing asynchronous data exchange between a client and a server (WebSocket standard defines short Ping and Pong messages to implement keep-alive mechanism). Using this connection, an IoT object (a server) can push a message to, for example, a web application (a client) whenever it obtains new data to send. Comparing to previously proposed real-time web techniques, such as Ajax polling and long-polling, WebSocket has much better server performance because maintaining an open low-level (i.e. TCP) connection consumes a small amount of server resources. In contrast, asynchronous communication provided by Ajax technologies requires opening and handling several HTTP connections, which are being closed due to timeouts or when HTTP response was sent (see Fig. 2), what has negative impact on server performance. Moreover, Websocket leads to latency reduction [23], since every re-establishment of HTTP connection, as it occurs in long-polling, requires performing additional TCP three-way handshake. Another advantage of WebSocket

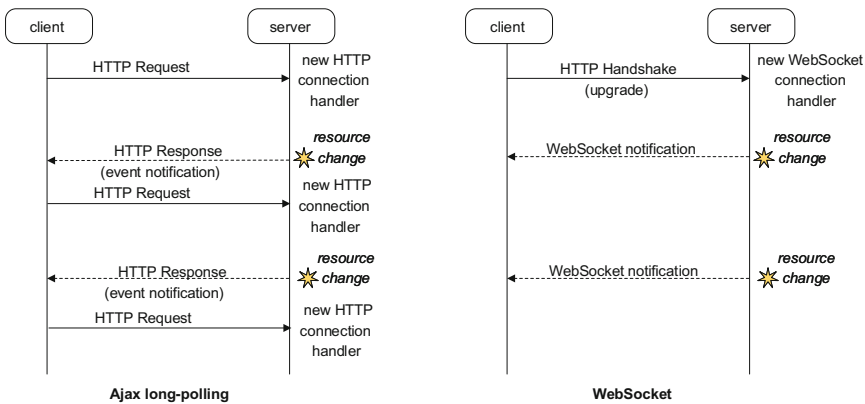


Fig. 2 Sequence diagrams for Ajax long-polling and WebSocket real-time communication technologies

is a very low protocol overhead [18]. Data transferred through WebSocket channel are organized into frames with a header size of 2 bytes only.

Presented features make the WebSocket considered as a good candidate to ensure real-time communication in WoT. In [24] authors propose to use this protocol to connect WSN (Wireless Sensor Network) gateway with sensor cloud service in IoT industrial production system, whereas solution presented in [25] demonstrates using WebSocket to set up communication between smartphone and system controller deployed on Raspberry Pi platform, in home automation scenario.

WebSocket standard specifies two types of application payload: UTF-8 text and binary data. However, during HTTP handshake a client can indicate the use of a specific subprotocol, i.e. an application-level protocol layered over WebSocket connection. One of the registered subprotocol is WAMP (Web Application Messaging Protocol) [26], which offers publish/subscribe communication pattern using URIs as identifiers and JSON for message serialization.

3.2 WebRTC

HTML5 recommendation specifies also WebRTC [12], a technology which provides real-time communication between web clients (in contrast to WebSocket, which operates on client-server connections). The primary goal of WebRTC is to enable inter-personal multimedia services using standard, built-in functionalities of web browsers. In this way, WebRTC introduces a real-time voice and video communication straight into web services, eliminating the need to install dedicated software or plug-ins.

Although WebRTC allows web applications to send and receive streaming data directly to/from remote peers, without relying it through an intermediate server, it still needs out-of-band signaling to initialize peer-to-peer communication. This functionality is typically provided through a dedicated web server (see Fig. 3). WebRTC specification assumes Session Description Protocol (SDP) for connection parameters negotiation. On the other hand, the specification does not define concrete technologies to be used for exchanging SDP messages between peers. Nevertheless, applied signaling protocol should be layered over HTTP (or WebSocket) to easily pass through firewalls and proxies as a standard web traffic. The analogous problem applies to data plane. To cope with this issue, WebRTC exploits RTP/Secure RTP (Real-time Transport Protocol [27]) for media transport and it requires encapsulation of RTP packets into ICE (Interactive Connectivity Establishment) protocol [28], which provides a set of techniques for traversing NATs and firewalls.

Apart from *RTCPeerConnection* API (Application Programming Interface) for creating audio and video connections, WebRTC provides also *RTCDataChannel* API, which enables peer-to-peer data sharing between web clients. The WebRTC data channel is constructed using Stream Control Transmission Protocol (SCTP [29]) as a generic transport service, which is encapsulated in DTLS (Datagram

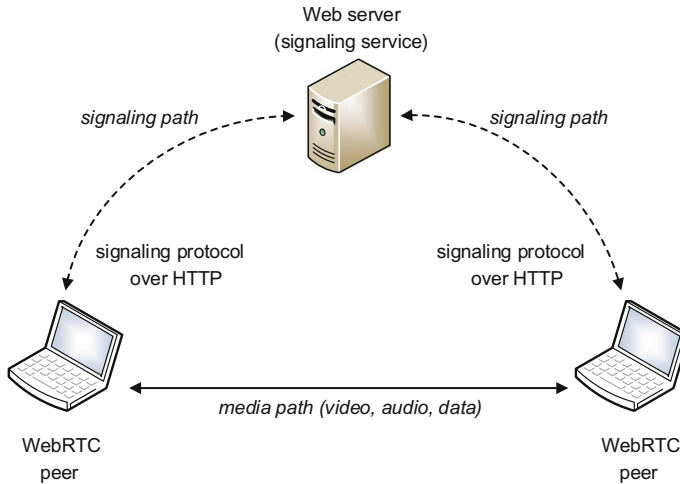


Fig. 3 WebRTC communication model

Transport Layer Security [30]), and next layered over ICE [31]. The data channel is considered as a solution which can merge WebRTC with the IoT, since it will enable to enrich inter-personal real-time communication with information about surrounding environment available at IoT domain. In this way, WebRTC can be thought as a tool for easy integration of voice and video services with IoT objects within one application. It can apply in various IoT scenarios. For example, during WebRTC video teleconsultation a physician can have real-time access to sensor data gathered by patient's personal e-health wearables. There are already available platforms such as open-source Streembit [32] or commercial Skedans [33], which exploit WebRTC to provide real-time collaboration between humans and smart devices (e.g. drones).

3.3 CoAP

Constrained Application Protocol (CoAP) [34] is a solution specifically designed to operate in a resource-constrained IoT environment. It is based on the REST concept and exploits client/server communication model to manipulate the resources using well-known GET, PUT, POST and DELETE requests of the HTTP. Likewise, CoAP response codes are in accordance with the HTTP specification. In this way, CoAP is characterized by easy integration of IoT devices with existing web services.

Furthermore, CoAP is a very lightweight. It is binary protocol and uses fixed header with a length of 4 bytes only, which may be followed by additional options header. Comparison presented in [35] shows that average transaction size in bytes is

almost 10 times smaller in CoAP than in HTTP. Additionally, CoAP operates over UDP to avoid bandwidth and energy-consuming TCP mechanisms, such as three-way handshake and automatic packet retransmissions. Since UDP is a stateless protocol, CoAP introduces two abstract sub-layers. The upper sub-layer is responsible for requests/responses handling, whereas the lower controls data transmission through UDP using four types of messages:

- Confirmable (CON)—the message must be acknowledged;
- Non-confirmable (NON)—acknowledgement of the message is not required;
- Acknowledgement (ACK)—it acknowledges the CON message;
- Reset (RST)—it is used to indicate that received request cannot be properly processed.

CON messages are designed for establishing reliable communication based on retransmissions and timeout mechanisms (see Fig. 4). Real-time data transfer, for which retransmissions are undesired due to excessive delay, can be realized using NON messages. Moreover, NON messages are used for implementing multicast connections [36].

Besides client/server interaction model, CoAP offers also a publish/subscribe functionality [37], which is triggered by setting the *observe* option in the GET request. The server that receives request with this option, registers the sender as the observer of the requested resource. Then, whenever the state of the resource changes, the server sends a notification to all clients from the list of registered observers. To remove a subscription, a client uses a RST message as a response to notification received from a server.

There are many CoAP implementation available nowadays. Some of them are designed for non-constrained devices, such as jCoAP [38] intended for Java-based mobile devices (e.g. smartphones with Android operating system). Other implementations are highly optimized and can be used as a build-in modules in embedded operating systems. The most popular are *Erbium* [39] used in Contiki and *libcoap* [40], which has been ported to TinyOS.

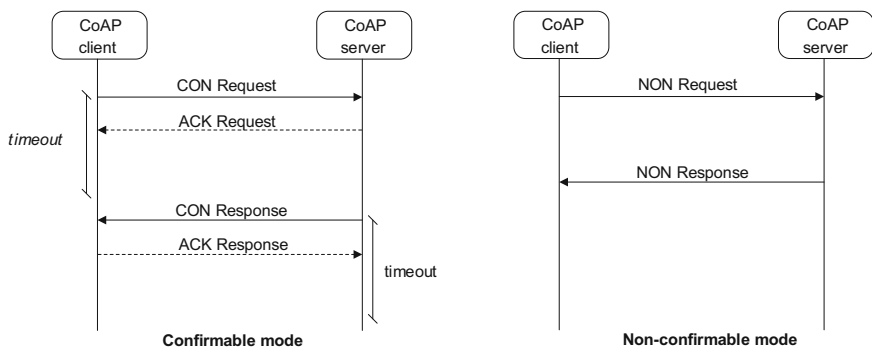


Fig. 4 Message exchange diagram for CoAP confirmable and non-confirmable modes

3.4 MQTT

Message Queue Telemetry Transport (MQTT) [41] is an open, asynchronous publish/subscribe protocol that operates over TCP connections. It is simple and lightweight (2 bytes header), since it was directly designed for resource-constrained devices and low-bandwidth, high-latency or unreliable networks. Hence, it has been proposed for various IoT applications, such as home automation system [42], personal e-health system [43] or to control Smart Lights [44].

The communication between MQTT publisher and subscriber is provided by intermediating broker. Publishers publish messages to specified topics, and a broker distributes them among the registered subscribers of given topic. The topics are composed by slash-separated strings; each string is considered as a level, and subscribers can use wildcards to express interest in a group of topics (+—match all strings on one level; #—match all strings below a level). MQTT broker may hold last published message on each topic (*retained message*) and next send it to a newly connected subscriber, so the subscriber obtains last good value immediately after registration.

The protocol attempts to ensure reliability and assurance in message delivery. It provides three QoS (Quality of Service) levels of guarantees for delivering a message:

- Level 0—a message delivered at most once, no confirmation required;
- Level 1—a message delivered at least once, confirmation required;
- Level 2—a message delivered exactly once, with a handshake procedure.

Higher QoS level needs more communication between a broker and MQTT client so there is a trade-off between QoS and protocol overhead. MQTT defines also the Last Will and Testament (LWT) feature for stressing publisher abnormal disconnection. When a publisher disconnects ungracefully, LWT message (defined by the publisher during registration) is send to its subscribers to inform them about device or network failure. LWT message is ignored when a publisher ends communication with a broker in a proper manner (i.e. by sending DISCONNECT message).

3.5 AMQP

Advanced Message Queuing Protocol (AMQP) [45, 46] is an open, TCP/IP-based publish/subscribe protocol that can be used for brokering of messages between different processes or systems regardless of their internal designs. AMQP broker handles messages by *exchanges* which route the messages to none, one or many queues according to specified rules (called *bindings*) depending on a usage scenario. When publishing a message, publishers may specify various meta-data to be used by an exchange in queue selecting process, but typically a queue is bound to an exchange by usage of a subscription key.

Each AMQP message contains the routing key, which is composed as a chain of strings separated by dots—each string is considered as a level. The subscription keys used by queues are similar to routing keys, but may contain two wildcards, as it is in MQTT. A message goes to a queue when both keys match.

The protocol was designed for business application, where reliability is a must. Therefore, subscribers must acknowledge acceptance of each message what ensures its delivery to an application (TCP protocol guarantees data delivery only at the transport layer). Queues may provide temporary storage when the destination is busy or not connected. Since the message brokering task is divided into exchanges and message queues, it gives more flexibility for developers in designing applications. A message can go through a chain of AMQP brokers and can be processed or modified in many places, as well as be processed in parallel. Moreover, AMQP is a binary protocol so its overhead is quite low, with fixed-size header of 8 bytes. The authors of [47] exploited AMQP in their *Stack4Things* platform designed for managing IoT devices in Smart City scenario.

4 Discussion and Future Trends Analysis

The power of the WoT concept is the seamless integration of IoT objects with the whole web ecosystem. While different vendor-specific solutions cover a large part of the current market (survey presented in [48] shows that almost a half of respondents deployed custom written IoT protocol in their networks), we think that in the near future systems that base on standard web technologies will become predominant. Simplicity is the key to widespread implementation and use of IoT services, while web-based IoT solutions take advantage of the open HTTP protocol and therefore they can be easily integrated with web services without the need for significant investments in a new infrastructure. Furthermore, the possibility of using the existing knowledge, in terms of skilled engineers and deployment best practices, and also well-known tools, could result in another reduction of WoT implementation costs.

In the context of WoT, a real-time communication is of particular importance. Unfortunately, such communication is very difficult to achieve by IoT devices with traditional web mechanisms as HTTP request/response or Ajax polling.

The release of HTML5 specification can cope with this issue, and especially the WebSocket protocol. It enables bidirectional real-time interaction between a client and a server through always-on TCP connection. Compared with traditional HTTP techniques, it decreases the workload of a server and introduces significantly lower communication overhead. Another advantage is a full compatibility with the Web, since it uses HTTP request during initial phase of connection establishment. It also provides built-in security by taking advantage of TLS (Transport Layer Security [49]) cryptographic protocol. WebSocket may become a major candidate in each scenario, where involved IoT devices have enough resources to cope with HTTP/TCP/IP stack implementation (it is worth noting that there has been

developed implementations of a web server, for which ROM and RAM memory requirements do not exceed a dozen kilobytes [50]). What is more, WebSocket does not need full HTTP stack implementation. Only a minimal set of functions is required, with Upgrade option, just to handle the protocol switching procedure.

One more powerful technology introduced by HTML5 is the WebRTC. Although WebRTC and IoT intersection is broadly discussed, and several platforms which merge WebRTC with IoT devices have already been presented, in our opinion it is hard to expect that this protocol will play a significant role in the WoT development. WebRTC requires implementation of full HTTP stack, jointly with additional signaling protocols, thus it imposes too large computational and memory burden to many IoT devices (the authors in [51] indicate that the current WebRTC runtime is quite CPU intensive even in traditional web browsers).

Another key point is that WebRTC has been designed for direct client-to-client communication. In IoT there is no clear separation between a client and a server, since most devices exploit both functionalities (for example, a sensor acts as a client and periodically sends gathered data to a server and, at the same time, it is a server that listens on commands from a control web application). In that case in IoT applications we can dispose of client-to-client restriction and achieve the WebRTC functionality simply by using standard client-server solutions (for example, WebSocket).

Many IoT scenarios are strongly oriented on event-driven flowchart and they need implementation of the publish/subscribe pattern that enables asynchronous communication. This pattern provides a native support for the sporadic connectivity since it allows the IoT device to publish a notification regardless of the connection status of recipients. HTTP with its client-pull communication does not fully match the above requirements, therefore specialized protocols are considered in this area. MQTT arouses considerable interest as an easy and lightweight solution (other considered publish/subscribe protocol, AMQP, has a more reach feature set, but it introduces higher overhead compared to MQTT).

It is also possible to implement publish/subscribe paradigm using WebSocket and its subprotocol WAMP. Such approach significantly simplifies implementation and management of a web application since all modules base on HTML5 framework. WAMP is Web-native, therefore it can interact with web services without any tunneling or bridging. However, due to higher overhead introduced by the WAMP-based approach, in our opinion the MQTT will be the first choice for scenarios that cover devices with more limited resources.

If we consider highly resource-constrained IoT objects, which periodically move into sleep mode for saving energy, the TCP-based solutions as presented above, are inappropriate. Each transition into the sleep mode means termination of TCP connection, and next expensive connection reestablishment. In that case integration with the Web can be performed with the CoAP protocol, which is dedicated for devices with limited computational and communication resources, and exploits connectionless UDP. As presented in [48], CoAP is not widely adopted yet, however in the future it is anticipated as one of the major players in IoT.

Nevertheless, from the Web point of view, CoAP (similar as MQTT) is still a custom protocol, which requires proxies to cooperate with HTTP-based web services, or dedicated servers for hosting CoAP-enabled web applications. However, using proxies entails some drawbacks. Proxy constitutes a single point of failure—when it collapses, all IoT objects from a CoAP domain have broken connectivity with the Web, regardless of which web service (i.e. HTTP server) they use. Required translation between CoAP and HTTP messages also brings in an extra latency, what might be problematic in case of real-time communications. Last but not least, using proxies in IoT environment, which is characterized by high mobility and variability, arises scalability issues. When the number of IoT objects will increase in a given CoAP domain, the objects might have a problem with access to web services due to proxy overload (even though the web services are hosted on various, not overloaded HTTP servers).

Future research directions related with WoT will focus on two main tracks: to determine how the current Web technologies should be adapted to best fit IoT requirements, and to explore new solutions that are suitable for resource-constrained IoT world and still integrate well with the Web [52]. For example, the solution for overcoming the interoperability problem may come out from a new version of HTTP protocol called HTTP/2 [53], which was approved by IETF in 2015. HTTP/2 preserves compatibility with the transaction semantics of its predecessor—it uses the same methods, headers and status codes. Consequently, it is fully compatible with the Web, bringing seamlessly web services to IoT objects without the need of using and maintaining additional devices such as proxies or gateways. Nevertheless, HTTP/2 exploits completely different mechanisms for transfer HTTP messages between endpoints. It provides efficient header compression [54] in order to reduce the protocol overhead. It also introduces a framing layer between HTTP and TCP planes, which is used for multiplexing several HTTP requests into one TCP connection.

Although HTTP/2 was designed taking into account requirements from the current Web only, its properties cause that it can be well tailored to the needs of Internet of Things. Furthermore, to cope with unnecessary delays and overhead introduced by TCP mechanisms, the replacement of TCP by QUIC protocol [55] may be investigated. QUIC is an experimental transport protocol developed by Google. It works over a connectionless UDP protocol, therefore it is characterized by reduced connection set up latency. For this reason, the further research can focus on adaptation of HTTP/2 and QUIC protocols stack to efficiently work with constrained IoT devices, what allows web applications to fully integrate with IoT objects and services offered by them. Moreover, further research activity can be related with other aspects of real-time communication, for example real-time search engines for efficient resource discovery.

5 Summary

Web of Things domain is much more heterogeneous compared to the traditional Web. Currently it is on its early stage of development, and we can distinguish several various solutions which are used, or are anticipated to use in the near future, for the same purposes.

In this chapter we provided a brief overview of the protocols that can be used to ensure (near) real-time communication between WoT entities and analyzed their suitability to meet the WoT real-time requirements. In our opinion, the most promising is WebSocket, which provides full interoperability with existing HTTP infrastructure. One can expect that the embedded WebSocket servers will be common in future IoT resource-constrained devices, causing the users to be able to control and manage the devices simply by using regular web browsers.

Nevertheless, since there is no one protocol that can satisfy all needs, we anticipate that WebSocket-based solutions will be supported by implementation of publish/subscribe interaction model with MQTT protocol, especially when considering devices with small computational and communication capabilities. Likewise, networks which nodes are characterized by very tight resource constraints, can rely on the CoAP protocol to avoid expensive TCP connections. The latter approaches require additional mechanisms (gateways or proxies) to ensure interoperability with HTTP-based web services. For this reason, one of the future research trends will focus on solutions that match the demands of devices with very limited resources, while ensuring seamless integration with the Web.

Acknowledgments This work was undertaken under the Pollux II IDSECOM project supported by the National Research Fund Luxembourg and the National Centre for Research and Development in Poland.

References

1. Chih-Lin I, Mikko A. Uusitalo, and Klaus Moessner, “The 5G Huddle (From the Guest Editors)”, *IEEE Vehicular Technology Magazine*, Volume 10, Issue 1, pp. 28–31, March 2015
2. J. Mongay Batalla, M. Gajewski, W. Latoszek, P. Krawiec, C X. Mavromoustakis, G. Mastorakis, “ID-based service-oriented communications for unified access to IoT”, *Computers and Electrical Engineering Journal*, Vol. 52, Elsevier, pp. 98–113, May 2016
3. M. Gajewski, P. Krawiec, “Identification and Access to Objects and Services in the IoT Environment”, Chapter on C.X. Mavromoustakis et al. (eds.), *Internet of Things (IoT) in 5G Mobile Technologies, Modeling and Optimization in Science and Technologies* vol. 8, Springer International Publishing, Switzerland, 2016
4. J. Mongay Batalla, G. Mastorakis, C. Mavromoustakis and J. Žurek, “On cohabitating networking technologies with common wireless access for Home Automation Systems purposes”. *IEEE Wireless Communications*, October 2016
5. J. Mongay Batalla and P. Krawiec, “Conception of ID layer performance at the network level for Internet of Things”. *Springer Journal Personal and Ubiquitous Computing*, Vol. 18, Issue 2, pp. 465–480, 2014

6. K. Karolewicz, A. Beben, J. Mongay Batalla, G. Mastorakis and C. Mavromoustakis, "On efficient data storage service for IoT". *International Journal of Network Management*, Wiley, May 2016, pp. 1–14, doi:[10.1002/nem.1932](https://doi.org/10.1002/nem.1932)
7. J. Mongay Batalla, K. Sienkiewicz, W. Latoszek, P. Krawiec, C. X. Mavromoustakis i G. Mastorakis, "Validation of virtualization platforms for I-IoT purposes". *The Journal of Supercomputing*, Springer US, 2016, doi:[10.1007/s11227-016-1844-2](https://doi.org/10.1007/s11227-016-1844-2)
8. D. Guinard, V. Trifa and E. Wilde, "A resource oriented architecture for the Web of Things," *Internet of Things (IOT)*, 2010, Tokyo, 2010, pp. 1–8. doi:[10.1109/IOT.2010.5678452](https://doi.org/10.1109/IOT.2010.5678452)
9. D. Guinard, V. Trifa, F. Mattern and E. Wilde, "From the Internet of Things to the Web of Things: Resource-oriented Architecture and Best Practices". Chapter in: *Architecting the In-ternet of Things*, Uckelmann, Dieter, Harrison, Mark, Michahelles, Florian (Eds.), Springer-Verlag Berlin Heidelberg, 2011, pp. 97–129
10. E. Bozdag, A. Mesbah, and A. van Deursen. "A Comparison of Push and Pull Techniques for AJAX". In *Proceedings of the 2007 9th IEEE International Workshop on Web Site Evolution (WSE'07)*. IEEE Computer Society, Washington, DC, USA, 2007, pp. 15–22
11. I. Fette, A. Melnikov, "The WebSocket Protocol", *Internet Engineering Task Force (IETF)*, RFC 6455, December 2011
12. H. Alvestrand, "Overview: Real Time Protocols for Browser-based Applications; draft-ietf-rtcweb-overview-15", *Internet Engineering Task Force (IETF) Internet-Draft*, January, 2016
13. G. C. Buttazzo, "Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications", *Real-Time Systems Series*. Santa Clara, CA: Springer-Verlag TELOS, 2004
14. Nilo Mitra, Yves Lafon (eds.), "SOAP Version 1.2 Part 0: Primer (Second Edition)", *W3C Recommendation, The World Wide Web Consortium*, April 2007. Available online: <https://www.w3.org/TR/soap12-part0/> Accessed 15 Jul 2016
15. Tim Bray et al. (eds.), "Extensible Markup Language (XML) 1.0", *W3C Recommendation, The World Wide Web Consortium*, November 2008. Available online: <https://www.w3.org/TR/xml/> Accessed 15 Jul 2016
16. C. Pautasso and E. Wilde. "Why is the web loosely coupled?: a multi-faceted metric for service design". In *Proc. of the 18th international conference on World Wide Web (WWW'09)*, pages 911–920, Madrid, Spain, April 2009. ACM
17. R. T. Fielding and R. N. Taylor, "Principled design of the modern web architecture," *ACM Transactions on Internet Technology*, vol. 2, May 2002
18. *OASIS Specification: Devices Profile for Web Services (DPWS) Version 1.1*, OASIS, July 2009. Available online: <http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01> Accessed 15 Jul 2016
19. Z. Shelby, "Embedded web services," *IEEE Wireless Communications* 17(6), pp. 52–57, January 2011
20. D. Yazar and A. Dunkels. "Efficient application integration in IP-based sensor networks". In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, page 4348, Berkeley, CA, USA, November 2009
21. D. Guinard, I. Ion, S. Mayer, "In Search of an Internet of Things Service Architecture: REST or WS-*? A Developers' Perspective", Chapter in: *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, A. Puiatti and T. Gu (eds.), Springer Berlin Heidelberg, 2012, pp. 326–337
22. Ian Hickson et al. (eds.), "HTML5. A vocabulary and associated APIs for HTML and XHTML", *W3C Recommendation, The World Wide Web Consortium*, October 2014. Available online: <https://www.w3.org/TR/html5/> Accessed 15 Jul 2016
23. Lubbers, P., and Greco, F. "Html5 web sockets: A quantum leap in scalability for the web". *SOA World Magazine* (2010)
24. Peng Hu, "A System Architecture for Software-Defined Industrial Internet of Things", In *Proc. of the Ubiquitous Wireless Broadband (ICUWB)*, 2015 IEEE International Conference on, IEEE, 2015, pp. 1–5

25. A. Hasibuan, M. Mustadi, I. E. Y. Syamsuddin and I. M. A. Rosidi, "Design and implementation of modular home automation based on wireless network, REST API, and WebSocket," 2015 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Nusa Dua, 2015, pp. 362–367
26. T. Oberstein, A. Goedde, "The Web Application Messaging Protocol; draft-oberstet-hybi-tavendo-wamp-02", Internet Engineering Task Force (IETF) Internet-Draft, October 2015
27. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", Internet Engineering Task Force (IETF), RFC 3550, July 2003
28. J. Rosenberg. "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols". Internet Engineering Task Force (IETF) RFC 5245, April 2010
29. R. Stewart (ed.), "Stream Control Transmission Protocol", Internet Engineering Task Force (IETF), RFC 4960, September 2007
30. E. Rescorla, N. Modadugu, "Datagram Transport Layer Security Version 1.2", Internet Engineering Task Force (IETF), RFC 6347, January 2012
31. H. Alvestrand, "Transports for WebRTC; draft-ietf-rtcweb-transports-06", Internet Engineering Task Force (IETF) Internet-Draft, August, 2014
32. Streembit: Decentralized, peer-to-peer, secure communication system for humans and machines. Project webpage: <http://streembit.github.io/> Accessed 15 Jul 2016
33. Skedans Systems. Webpage: <https://skedans.com> Accessed 15 Jul 2016
34. Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", Internet Engineering Task Force (IETF), RFC 7252, June 2014
35. W. Colitti, K. Steenhaut, and N. De Caro, "Integrating Wireless Sensor Networks with the Web," in *Extending the Internet to Low power and Lossy Networks (IP+SN 2011)*, 2011
36. A. Rahman, E. Dijk (eds.), "Group Communication for the Constrained Application Protocol (CoAP)", Internet Engineering Task Force (IETF), RFC 7390, October 2014
37. Hartke, K., "Observing Resources in the Constrained Application Protocol (CoAP)", Internet Engineering Task Force (IETF), RFC 7641, September 2015
38. B. Konieczek, M. Rethfeldt, F. Golasowski and D. Timmermann, "Real-Time Communication for the Internet of Things Using jCoAP," 2015 IEEE 18th International Symposium on Real-Time Distributed Computing, Auckland, 2015, pp. 134–141
39. M. Kovatsch, S. Duquennoy, and A. Dunkels. "A Low-Power CoAP for Contiki". In *Proc. of IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, Valencia, Spain, October 2011
40. K. Kuladinithi, O. Bergmann, T. Pötsch, M. Becker, C. Görg, "Implementation of CoAP and its Application in Transport Logistics", In *Proc. of the Workshop on Extending the Internet to Low power and Lossy Networks*, Chicago, IL, USA, April 2011
41. OASIS Specification: MQTT Version 3.1.1 Plus Errata 01, OASIS, July 2009. Available online: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html> Accessed 15 Jul 2016
42. Y. Upadhyay, A. Borole and D. Dileepan, "MQTT based secured home automation system," 2016 Symposium on Colossal Data Analysis and Networking (CDAN), Indore, Madhya Pradesh, India, 2016, pp. 1–4
43. Y. F. Gomes, D. F. S. Santos, H. O. Almeida and A. Perkusich, "Integrating MQTT and ISO/IEEE 11073 for health information sharing in the Internet of Things," 2015 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, 2015, pp. 200–201
44. N. K. Walia, P. Kalra and D. Mehrotra, "An IOT by information retrieval approach: Smart lights controlled using WiFi," 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), Noida, 2016, pp. 708–712
45. S. Vinoski, "Advanced Message Queuing Protocol," in *IEEE Internet Computing*, vol. 10, no. 6, pp. 87–89, Nov.-Dec. 2006. doi:10.1109/MIC.2006.116

46. OASIS Specification: Advanced Message Queuing Protocol AMQP Version 1.0, OASIS, 2012. Available online: <http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-overview-v1.0-os.html> Accessed 15 Jul 2016
47. G. Merlino, D. Bruneo, S. Distefano, F. Longo and A. Puliafito, “Stack4Things: Integrating IoT with OpenStack in a Smart City context,” Smart Computing Workshops (SMARTCOMP Workshops), 2014 International Conference on, Hong Kong, 2014, pp. 21–28
48. M. Thoma, T. Braun, C. Magerkurth and A. F. Antonescu, “Managing things and services with semantics: A survey,” 2014 IEEE Network Operations and Management Symposium (NOMS), Krakow, 2014, pp. 1–5
49. E. Rescorla “HTTP Over TLS” Internet Engineering Task Force (IETF), RFC 2818, May 2000
50. Deze Zeng, Song Guo, and Zixue Cheng, “The Web of Things: A Survey”, Journal of Communications, vol. 6, no. 6, September 2011
51. T. Sandholm, B. Magnusson and B. A. Johnsson, “An On-Demand WebRTC and IoT Device Tunneling Service for Hospitals,” Future Internet of Things and Cloud (FiCloud), 2014 International Conference on, Barcelona, 2014, pp. 53–60
52. J. Heuer, J. Hund and O. Pfaff, “Toward the Web of Things: Applying Web Technologies to the Physical World,” in Computer, vol. 48, no. 5, pp. 34–42, May 2015
53. M. Belshe, R. Peon, M. Thomson, (ed.) “Hypertext Transfer Protocol Version 2 (HTTP/2)” Internet Engineering Task Force (IETF), RFC 7540, May 2015
54. Peon, R. and H. Ruellan, “HPACK: Header Compression for HTTP/2”, Internet Engineering Task Force (IETF), RFC 7541, May 2015
55. J. Iyengar, I. Swett, “QUIC: A UDP-Based Secure and Reliable Transport for HTTP/2; draft-tsvwg-quic-protocol-00”, Internet Engineering Task Force (IETF) Internet-Draft, June, 2015

Crowd-Driven IoT/IoE Ecosystems: A Multidimensional Approach

**Xenia Ziouvelou, Panagiotis Alexandrou,
Constantinos Marios Angelopoulos, Orestis Evangelatos,
Joao Fernandes, Nikos Loumis, Frank McGroarty,
Sotiris Nikolettseas, Aleksandra Rankov, Theofanis Raptis,
Anna Ståhlbröst and Sebastien Ziegler**

Abstract During the past few years an astonishing paradigm shift has occurred towards a new participatory value creation model driven by users. Open collaborative innovation practices have emerged in which an increasing number of users mutually collaborate by openly communicating their ideas, sharing best practices, and creating new knowledge across sectors. These online, distributed, crowd-driven networks take advantage of underlying network effects in order to harness the collective power and intelligence of the Crowd. Such novel paradigms fuel an

X. Ziouvelou (✉) · F. McGroarty
University of Southampton, Southampton, UK
e-mail: p.ziouvelou@soton.ac.uk

F. McGroarty
e-mail: f.j.mcgroarty@soton.ac.uk

P. Alexandrou · S. Nikolettseas · T. Raptis
Computer Technology Institute & Press Diophantus, Patras, Greece
e-mail: aleksandro@ceid.upatras.gr

S. Nikolettseas
e-mail: nikole@cti.gr

T. Raptis
e-mail: traptis@ceid.upatras.gr

C.M. Angelopoulos · O. Evangelatos
Université de Genève, Geneva, Switzerland
e-mail: Marios.Angelopoulos@unige.ch

O. Evangelatos
e-mail: orestis.evangelatos@unige.ch

J. Fernandes
Alexandra Instituttet A/S, Aarhus, Denmark
e-mail: joao.fernandes@alexandra.dk

N. Loumis
University of Surrey, Guildford, UK
e-mail: nikolaos.loumis@surrey.ac.uk

increasing interest in mobile crowdsensing (MCS) methods in the context of IoT/IoE, which leverage the power and the wisdom of the crowd to observe, measure, and make sense of particular phenomena by exploiting user-owned mobile and wearable devices. However, when one examines the design and development of such ecosystems, realises that there is a gap in existing research. While emphasis has been placed upon the technical aspects, the success of such ecosystems is dependent on a number of diverse criteria. This chapter aims to fill this gap by providing a framework, which adopts a holistic approach based on multiple perspectives (namely technical, business, and people perspectives) and facilitates the design and development of crowd-driven ecosystems. This model is examined in the context of a hybrid crowd-driven IoT/IoE ecosystem, IoT Lab, in order to exemplify how these perspectives can be used to promote an ecosystem's success and detail the challenges faced. This analysis is extended through the introduction of the “*Crowd-driven Ecosystem Index (CEI)*”, which measures the coverage intensity of each of the key ecosystem parameters, denoting this way the propensity of success of a crowd-driven network.

1 Introduction

We are witnessing an ever-increasing interest in actively engaging with the crowd for particular purposes that range from simple ratings to task-oriented activities, and from problem solving to innovation creation across different industrial sectors. A paradigm shift has occurred towards a new participatory value creation model driven by users. As such, open collaborative value creation ecosystems have emerged providing an environment that nurtures this change in the role of the users from passive to active co-creators. These online, distributed crowd-driven networks leverage the network effects so as to harness the collective power and intelligence. Within these ecosystems users actively collaborate by openly communicating their ideas and data, sharing best practices and creating new knowledge that augments our innovation potential across various sectors.

As a result, recently there has been an increasing interest in studying the integration and the corresponding potential of this emerging participatory model in the

A. Rankov
DunavNET, Novi Sad, Serbia
e-mail: aleksandra.rankov@dunavnet.eu

A. Ståhlbröst
Luleå University of Technology, Luleå, Sweden
e-mail: anna.stahlbrost@ltu.se

S. Ziegler
Mandat International, Geneva, Switzerland
e-mail: sziegler@mandint.org

context of the Internet of Things (IoT)/Internet of Everything (IoE)¹ paradigm. This interest has also been fuelled by the high acceptance rates of truly portable hand-held and wearable smart devices that enable the creation of crowd-driven networks. These networks seek to exploit the embedded sensing capabilities and the intrinsic mobile nature of such devices. Consequently, the IoT/IoE environment has introduced new user-centric sensing paradigms, like mobile crowd sensing (MCS) [3, 4], that go beyond traditional sensing techniques (e.g., sensor networks, etc.) by leveraging both the power and the wisdom of the crowd. These paradigms enable the IoT/IoE environment to sense, observe, measure, and make sense of real-world conditions (e.g., environmental, etc.) and activities (e.g., personal activities and interactions, etc.) by engaging user-owned mobile and wearable devices.

However, despite the promising participatory value creation paradigm and the numerous advantages that it provides, crowd-driven IoT/IoE ecosystems are still in their initial stages and face many challenges as their success relies on several crucial elements. Due to their nature, these ecosystems necessitate that multi-disciplinary perspectives are addressed and combined during their conception and development processes. The fact that these networks are driven both by the research community and several organisations, makes it even more challenging to integrate and drive an optimal solution. As an example, motivators and support for incentives must be investigated in different perspectives, namely business, technology and end-user (people) perspectives, in order to understand for each one of them what are the corresponding needs, barriers, constraints, etc. and therefore conceptualise a model that addresses them in a combined way. Such a model will facilitate towards this end both the design and the development of successful crowd-driven ecosystems.

To the best of our knowledge, existing research efforts place emphasis only upon the technical aspects of the design and development of crowd-driven ecosystems. On the contrary, little attention is put on how this process should be designed and undertaken accounting for non-technical ecosystem elements as well. This chapter identifies and addresses this gap by providing a framework, which adopts a holistic approach based on multiple perspectives (technical, business and people perspectives) and facilitates the design and development of crowd-driven ecosystems.

This chapter is organised as follows: Sect. 2 presents the evolution of ecosystems placing emphasis on the emerging crowd-driven ecosystems and presents a taxonomy of the key aspects of each type of ecosystem. Section 3 presents a multi-dimensional approach for the design and development of crowd-driven ecosystems. While, Sect. 4 examines the proposed model in the context of a hybrid crowd-driven IoT ecosystem, namely IoT Lab, and introduces the “*Crowd-driven*

¹Due to the non-existence of a standard IoT definition [1] one can identify a variety of IoT definitions in the existing literature. The spectrum includes both narrow definitions, which perceive IoT only as an interconnection of “things” and broad ones that view the IoT idea as implying concepts that relate to the interconnection of people, processes and data in addition to “things” [2]. As such, in the course of this chapter we perceive IoT and IoE as acronyms for the same conceptual paradigm.

Ecosystem Index (CEI)”, which measures the coverage intensity of each of the key ecosystem parameters, denoting this way the propensity of success of a crowd-driven network. The chapter concludes in Sect. 5.

2 The Rise of Crowd-Driven Ecosystems

The notion of ecosystems is directly linked to the natural world. Coined in 1935 by the British botanist Arthur Tansley [5], the notion of ecosystems was introduced in order to denote a community of living organisms interacting with each other and their environment as a system. Such *biological ecosystems* were considered to be evolving systems, that are “dynamic, constantly remaking themselves, reacting to natural disturbances and to the competition among and between species” [6], p. 11. As an analogue of such biological ecosystems, the concept of *industrial ecosystems* was presented a few years later [7] denoting ecosystems where all material is recycled infinitely and efficiently by changing the habits of manufacturers and consumers maintaining this way our standard of living without causing environmental devastation [7], p. 145. In the business context it was Moore [8]² who made the parallel and proposed that a company can be viewed “not as a member of a single industry but as part of a business ecosystem that crosses a variety of industries” (p. 76). This transition from standalone companies to integrated corporate systems and eventually crowd-driven ecosystems is powered by technology, user participation and the move towards open innovation [10]. Emerging crowd-driven ecosystems leverage the network effects and harness the collective intelligence of a large number of contributors. Four distinct ecosystem types are distinguished in this chapter: knowledge ecosystems, business ecosystems, innovation ecosystems and crowd-driven ecosystems. In the sections that follow, we provide a short overview of these ecosystem concepts and provide a taxonomy (Table 1).

Existing business literature has long recognised the advantages of geographically clustered organisational entities that benefit from their co-location and the dynamic knowledge interactions that occur between them [11]. Such *knowledge ecosystems* play a central role in increasing knowledge creation and the speed of innovation diffusion [12] through evolutionary networks of collaboration [13]. In the online environment, such knowledge interactions can be identified within open source communities where knowledge creation and co-creation among community members that exhibit virtual proximity/co-location [14] is evident. Although research in knowledge ecosystems has implicitly assumed that such knowledge ecosystems evolve into business ecosystems, existing studies in the area indicate that there is a

²According to Moore, a business ecosystem is “an economic community supported by a foundation of interacting organizations and individuals—the organisms of the business world” [9]. As he suggests it is “conscious choice” that differentiates between ecological and social systems [9], p. 18.

Table 1 Taxonomy of the different types of ecosystems

	Knowledge ecosystem	Business ecosystem	Innovation ecosystem	Crowd-driven ecosystem
Function	New knowledge creation	Customer value (knowledge commercialisation)	Innovation creation/co-innovation	Crowd-driven shared value creation/co-creation
Connectivity	Decentralised and distributed	Geographically clustered or Global and distributed	Geographically clustered or Global and distributed	Global and distributed
Mode	Physical or online	Physical or online	Physical or online	Online
Relationships	Synergistic and co-operative	Competitive and collaborative (“co-opetition”)	Co-operative, collaborative	Co-operative and collaborative (mass collaboration)
Openness	High degree of openness or closed	Various degrees of openness	High degree of openness or closed	High degree of openness
Structure	Dynamic inter-organisational, inter-personal	Dynamic or static, and inter-organisational	Dynamic inter-organisational and inter-personal	Dynamic
Key actor	University, research organisation/institute	Large company	Large company or community	NGO/Non-profit initiative or community or company

(Own elaboration extending [15, 17])

disconnection between the development of each type of ecosystem as they have different value creation processes [15]. **Business ecosystems** are seen as economic communities of interacting “organisms of the business world” [9], p. 9 with many horizontal relations with a “coopetition” structure (both collaborative and competitive relationships) [8] aiming to jointly deliver a product or service to customers [15]. According to Moore, such ecosystems are a composition of customers, lead producers, competitors, and other stakeholders,³ while “the keystone species”, are leadership companies with a strong influence [9], p. 25. Business ecosystems focus on the commercialisation of knowledge and aim to deliver value to the end users as an interrelated system of interdependent companies rather than as individual companies ([15, 17]). These nested business networks act as a source of competitive advantage for individual business entities, and depending on the ecosystems’ degree of productivity, robustness and ability to create opportunities for new firms they can succeed [18]. The online environment facilitates the creation, co-evolution and expansion of such ecosystems across diverse business sectors.

Innovation ecosystems, on the other hand, can be either physical or online/virtual networks that focus on fostering creativity, as well as, triggering, developing and diffusing innovation and enabling technological development among diverse entities in an open or closed context. They are based on successful examples of agglomeration whether in geographic, economic, industrial or entrepreneurial terms [19] and unlike business ecosystems, they lack the customer (demand) side [20]. Innovation ecosystems have emerged as a multilevel, multimodal, multinodal, and multiagent system of systems [21] where innovation, co-creation, and co-innovation occur in order to generate shared value [22]. Living labs are seen as open innovation ecosystems centered on systematic user co-creation practices, which integrate research, and innovation practices in real life communities and settings.

The emergence of **crowd-driven ecosystems**, has been powered by technology, open innovation, and participatory value creation processes driven by users. These virtual distributed ecosystems have created a global meta-environment for facilitating a change in the role of the users from passive to active creators, co-creators, collaborators and co-innovators. Crowd-driven ecosystems leverage the distributed network effects and harness the collective power and intelligence of the user community that massively collaborates [23] creating in such a way shared value [24]. These open, collaborative user-driven, value creation ecosystems enable individuals to collaborate by openly communicating their ideas, sharing data, best practices and creating new knowledge that enhances the innovation potential of our society. In particular, they explore the direct and indirect interactions with the user community through crowdsourcing, crowdsensing and crowdfunding processes harnessing this way the collective crowd capital.

This ability to exploit the capacity of the crowd has been fueled by the Internet of Things (IoT)/Internet of Everything (IoE), introducing new user-centric

³See [16] for an overview of business ecosystems and literature in relation to peripheral and non-peripheral actors in the business ecosystem definition.

paradigms, such as mobile crowd sensing (MCS) [3, 4]. MCS goes beyond traditional sensing techniques (e.g., sensor networks, etc.) leveraging both the power and the wisdom of the crowd in order to sense, observe, measure and make sense of real-world conditions (e.g., environmental, etc.) and activities (e.g., personal activities and interactions, etc.) using user-owned mobile and wearable devices. **Crowd-driven IoT/IoE ecosystems** can exist in many forms, concentrating on specific crowd-driven functions (crowdsourcing, crowdsensing, crowdfunding, etc.) or, increasingly, they can be hybrid; not tied to a specific mode. In the former case, we can identify mobile crowdsourcing ecosystems such as OpenStreetMap (crowdsourced map of the world), Waze (crowd-driven traffic navigation) as well as, web-based ones such as Amazon Mechanical Turk. Similarly, in the context of crowdsensing ecosystems we can identify networks such as PhoneLab (open access smartphone testbed), Ushahidi (user geo-location data) and APISENSE (crowd-sensing for experimental datasets) among others that monitor from city noise [25], to climate [26] and emergencies [27].

One can identify only a few crowd-driven ecosystems that integrate both crowdsourcing and crowdsensing perspectives. Examples of such distributed hybrid crowd-driven IoT/IoE ecosystems are *mCrowd* (crowdsourcing and participatory sensing) and *EpiCollect* (crowdsourcing and crowdsensing for survey purposes), which cover crowdsourcing and crowdsensing elements just partially. Additionally, they do not facilitate the integration of existing physical IoT testbeds and existing FIRE testbeds with any crowd-driven resources such as smartphones. To our knowledge, the only crowd-driven IoT ecosystem that integrates both crowdsourcing and crowdsensing (opportunistic and participatory sensing) elements, while it assimilates smartphones with existing testbeds, is *IoT Lab*.

However, when one examines the design and development of such crowd-driven ecosystems, would find a gap in existing research. To date the emphasis is placed only on technical perspectives related to crowd-driven IoT/IoE ecosystems. While the existing literature gives a lot of emphasis upon the technical aspects for the development of such networks and provides useful insights into what needs to be addressed; there is relatively little direction on how this process should be designed and undertaken, accounting for non-technical ecosystem elements. Hence, there is a need for a unified framework that embraces a holistic approach, to address different parameters that are of critical importance for the design and development of such crowd-driven ecosystems is required.

3 Crowd-Driven Ecosystems: A Multidimensional Approach

Our examination of the various crowd-driven ecosystems enables us to identify key perspectives that describe them and facilitate a holistic analysis as well as a set of key thematic areas that detail further such ecosystems (Fig. 1). The first is the people-centric perspective that encompasses the crowd views and needs for the

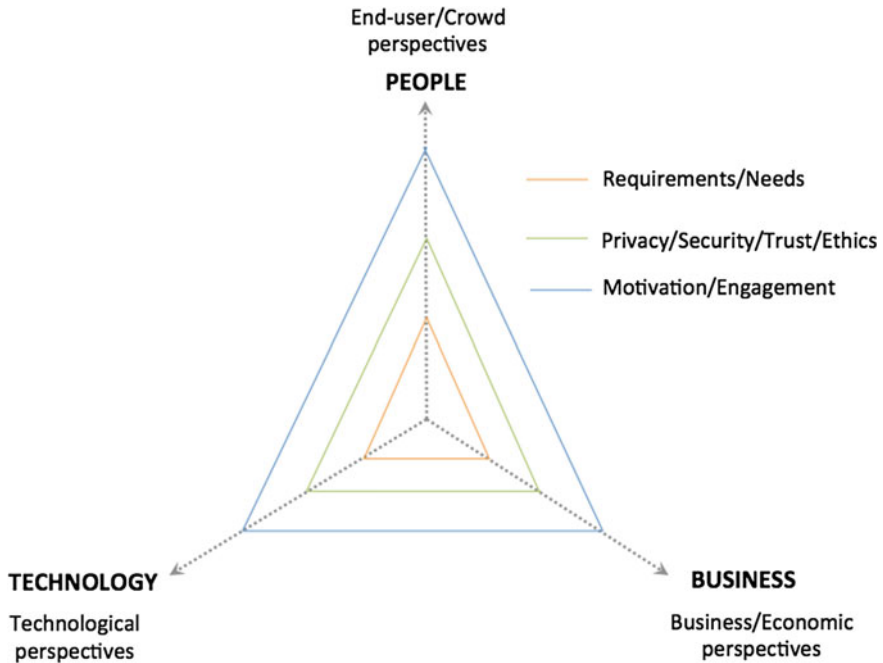


Fig. 1 A multi-dimensional model for crowd-driven IoT/IoE ecosystems

creation and co-creation of value within a given network. These needs are centered on both direct (e.g., privacy, security, trust, etc.) and indirect requirements (e.g., motivation, engagement, ethics, etc.). Secondly, the business-centric perspective whose focus is the generation of economic and business value of a given crowd-driven ecosystems; factors that affect the network sustainability. Lastly, the technology-centric perspective focuses on the technological aspects (e.g., ecosystem architecture and components, technological resources, etc.) relevant to the design and development of a crowd-driven network. The above aspects connect the people-and business-centric perspectives while integrating emerging technological advancements.

In order to further analyse these perspectives we identified some key horizontal thematic areas which describe each of these perspectives and involve: (a) *requirements and needs*: entails the definition of the needs and conditions to be fulfilled as well as the assessment of the relative importance or “value” of these specifications; (b) *privacy, security, trust and ethics*: involves the identification of each of these elements in the context of a given crowd-driven ecosystem and exploration of the inter-relationship between them, aiming to align them so as to ensure the success of the ecosystem; and (c) *motivation and engagement*: consists of overcoming the obstacles of adoption by achieving and sustaining high levels of user participation and active involvement throughout the life-cycle of the ecosystem. The sections that

Table 2 Parameters to be considered for the design and development of crowd-driven IoT/IoE ecosystems

Key thematic areas		Perspectives	
Requirements/Needs	Technical perspective	Business perspective	People perspective
<ul style="list-style-type: none"> ● Privacy/Security/Trust/Ethics 	<ul style="list-style-type: none"> ● Components of the system/types ● Interactions between the components/networking ● Types of Resources and their representation ● Services provided/supported ● Technologies invoked/used ● Scalability/Resilience/Modularity ● Organisation/platform – future extensions ● External factors/threats to the system/access control ● Interaction with end-users ● Heterogeneity of data collected from the crowd ● QoS (Quality of Service) 	<ul style="list-style-type: none"> ● Ecosystem structure and governance ● Ecosystem business model (value creation and value capture) ● Value creation: <ul style="list-style-type: none"> – Identify a unique value proposition (unique product/service/offering that provides value for each stakeholder of the ecosystem) – Value co-creation – Value creation for the partner/collaborator network ● Value capture ● Sustaining the crowd-driven networked ecosystem: Identify sustainability model (economic, community and innovation perspectives of sustainability) & co-evolution model 	<ul style="list-style-type: none"> ● Stimulate users' needs of the system, and their needs in the system ● Engaging and interesting cases ● Easy to use ● Intuitive and encouraging design ● Measurement of the crowd activity/quality of the crowd involvement/Crowd preferences ● Users in control ● Status of ongoing activities ● User achievements visibility
	<ul style="list-style-type: none"> ● Privacy/security/integrity/trust ● Privacy by design approach ● Personal data protection norms ● Identity management ● Information Security ● Data/information integrity ● Perceived trustworthiness of the system ● Clear description of the required data to the user (provided by either user or device) 	<ul style="list-style-type: none"> ● Ensure security and privacy of the system and convey to crowd and ecosystem stakeholders ● Clarify data ownership and privacy ● Facilitate the creation of secure and trusted relationships on top of the technical secure/privacy-aware/trusted infrastructure ● Ensure ethical ecosystem practices ● Internal business reputation mechanisms 	<ul style="list-style-type: none"> ● Clear description of the required data (provided by either user or device) needed ● Secure, trusted and easy to use system ● Personal integrity intact throughout the use ● Design for the right to be forgotten ● Crowdsensing:

(continued)

Table 2 (continued)

Key thematic areas		Perspectives	
	Technical perspective	Business perspective	People perspective
	<ul style="list-style-type: none"> ● Relation between the trustworthiness and privacy concerns ● Transparency in information security important for trustworthiness of the system and participation rate ● DB protection from external attacks ● Reputation 		<ul style="list-style-type: none"> ● Opportunistic sensing (coming from people) and information security <ul style="list-style-type: none"> – Data from sensors is involuntary – No control over data collection (what, when, where) – Raises serious privacy concerns – Design for the right to be forgotten ● Crowdsourcing <ul style="list-style-type: none"> ● Participatory crowdsourcing <ul style="list-style-type: none"> – Voluntary data gathering method – individual chooses what she/he wants to report to the system – Minimal privacy concerns but no control after reporting – Important information security, integrity, availability to correct stakeholders

(continued)

Table 2 (continued)

Key thematic areas	Perspectives		
	Technical perspective	Business perspective	People perspective
Motivation/Engagement	<ul style="list-style-type: none"> ● End-user involvement from conception to product ● Importance of co-design and co-creation for defining the technical product ● Understanding the crowd needs ● Consideration of requirements from multi-stakeholder perspective ● Paying/incentives ● Incentive mechanisms 	<ul style="list-style-type: none"> ● Motivate and engage with the crowd and all different ecosystem stakeholders: incentive mechanisms design ● Address ecosystem evolution parameters and crowd motivation and engagement 	<ul style="list-style-type: none"> ● Understanding what motivates the crowd and their participation ● Fun ● Fame ● Fortune ● Fulfilment

follow describe in detail the different perspectives. An overview of the key factors for each perspective is presented in Table 2 so as to facilitate the design and development of crowd-driven IoT and IoE ecosystems.

(a) *People's Perspective*

Viewing the crowd-driven eco-system from a people's perspective set emphasis on understanding what triggers people to start using the system and what keeps them continue participating in it. At this point, it is important to understand the *needs and requirements* of the end-users, in order to ensure that they get a value from using the system. We must underline the fact that users' needs should be satisfied on two different levels focusing both on the needs of the system and needs in the system. More specifically, a crowd driven eco-system should offer a high level of attractiveness, usability, and foster engagement in the crowd. Usually, in crowd-driven eco-systems, the crowd is consisted by end-users who participate on their spare-time without any previous training, or by being forced or paid to use the system. Consequently, a crowd-driven platform must be very easy to use with a low entrance barrier that creates a clear value for the end-users.

To ensure that end-users wants to participate in, and contribute to, crowd-driven ecosystem it is also important that they can feel safe and secure and that their privacy maintains intact privacy even if they share personal data. This means that end-users privacy must be protected on two different levels, one being the data protection and how the personal data (e.g. personal profile) is protected, managed, used and stored in the back-end systems and the other level being the user interface of the intermediary platform where privacy is related to what is shown about the crowd participants in a public sphere in the platform, i.e. user profiles, engagement in tasks, data they have shared, i.e. the system must be transparent and open. Today data is increasingly viewed as the "holy grail" to understanding end-users and to foster innovation. In many crowd-driven eco-systems end-users share some of their data consciously and openly, such as personal profiles and e.g. photos, but end-users can also share data that is not as obvious and aware to them such through their smart phones sensors, or other wearables, i.e. participatory sensing. Hence, these types of systems should follow privacy-by-design principles that make the users safe and not having to consider all possible threats that they might face from using the system. Hence, individuals should have the power to determine when, how and to what extent information about them is communicated to others [28, 29].

While ensuring that end-user privacy is protected by the crowd-driven ecosystem, equally important is to understand what *motivate* the crowd to achieve the best outcome of crowdsourcing [30] and crowdsensing. In previous research on crowdsourcing and motivation (e.g., [31–33]), factors such as enjoyment, career concerns, satisfying intellectual interest, increase of status, supporting the community, feeling affiliated and create social contacts have been identified. Research has led to the conclusion that crowds are motivated differently depending on the type of crowdsourcing initiative they are engaged in [34]. For instance, in collaborative crowdsourcing such as Zooniverse and OpenIdeo, contributing to a larger

cause is what mainly motivates the crowd. While in compensation focused crowdsourcing, such as Amazon Mechanical Turk and iStockPhoto, the crowd is mainly motivated by the possibility to earn money and in competition focused crowdsourcing, such as InnoCentive and NineSigma, the main motivator is the challenge and to win a prize. Overall, motives such as enjoyment, having fun and the ability to kill time with meaningful activities, stretches along all different types of crowds. Hence, motivations for crowds can be summarised into fun, fame, fortune and fulfilment.

(b) *Technical Perspective*

From a technical perspective, crowd-driven ecosystems require a focus on the technological enablers needed for leveraging crowdsourced infrastructure as experimental resources and potentially federating them with other IoT experimenting facilities. With respect to “*requirements and user utility*”, a virtualisation roadmap of crowdsourced resources is needed. In order for the devised solutions to be replicable, the roadmap should employ standardised federation and virtualisation architectures such as the Slice Based Federation Architecture (SFA); the de facto standard used nowadays for experimental resources virtualisation and federation. Up until now, SFA has been used for federating resources of computer networks. Therefore, in order to also effectively address emerging paradigms, such as Mobile Crowdsensing Systems, SFA needs to be significantly extended appropriately so as to also include crowdsourced and other IoT resources. One potential extension of the representation and abstraction mechanisms could take place via the IPSO Application Framework that defines the communication interfaces of constrained embedded devices and smart objects that would enable crowdsourced resources to be represented as regular IoT resources. A second necessary extension is the design and implementation of such mechanisms that support the opportunistic integration of crowdsourced resources (e.g. smartphones, tablets, etc.) with the corresponding Mobile Crowdsensing functionalities. Such an extension would empower experimenters to opportunistically augment the capabilities of experimental facilities and establish two-way interaction schemes with the end-users (e.g. push notifications and reception of sensory readings and user feedback). Crowd-driven ecosystems, due to the strong personal nature of corresponding devices (e.g. smartphones) raise significant issues that are not present in regular IoT resources. In order to properly address this sensitive nature of personal devices, several privacy, anonymity and security mechanisms need to be integrated, in order to offer an increased level of trustworthiness towards the end-users providing access to their personal devices. In this respect, the integration of crowdsourced resources that can be provided by the general public raises important challenges with respect to “*Privacy, security, trust and ethics*”. In order to efficiently and effectively address such issues technical solutions guaranteeing privacy and anonymity for the contributors need to be considered as well as multi-dimensional authentication and authorisation methods that enable the collected data to be treated in a secure way; for instance, to guarantee security when transmitting, storing and accessing data.

With respect to “*motivation and engagement*”, technical perspectives may focus both on the users of a platform and to the crowd contributing to a crowdsourced facility. The users of crowd-driven platforms in most cases consist of researchers who seek to exploit the diverse and numerous resources available. Therefore, such a platform needs to take under consideration technical solutions that guarantee scalability, resilience, and efficiency. For instance, in order to provide an efficient way of representing all the available resources, while being able to store vast volumes of data, hybrid database schemes shall be considered, as they combine both relational and non-relational databases. The contributors of the platform—which is the general public—can be provided with a crowdsourcing tool, e.g. a smartphone application, with an appealing and modern “look and feel” that will be non-intrusive to the smartphone user.

(c) *Business Perspective*

The business perspective places emphasis upon the economic activity related to the crowd-driven ecosystem. As such, it examines the value creation and capture, processes within similar ecosystems, as well as, the creators and co-creators of the aforementioned value. This perspective is centered on three generic thematic areas, presented above, as it can be seen in Table 2. Concerning “*requirements and needs*”, the main attention of the business perspective is paid to the ecosystem structure and governance and its business model of the key value creation and value capture elements. When it comes to the ecosystem value creation, it is critical to identify the unique value proposition of the ecosystem, in order to provide value and increase the utility of all the different ecosystem entities, as well as, to explore the value co-creation processes and its co-creators. Another essential parameter of the crowd-driven ecosystem relates to its sustainability. As such, the identification of the appropriate sustainability model is critical so as to analyse different sustainability variables such as economic, community and innovation perspectives, in addition to the evolution of the crowd-driven network itself and how this impacts its sustainability.

“*Privacy, security, trust and ethics*”, constitutes a central thematic area for the business perspective as it significantly impacts the ecosystem adoption. As such, it focuses upon initially ensuring the security and privacy of the crowd-driven ecosystem and subsequently conveying this assurance to the crowd and all ecosystem stakeholders via appropriate set of policies and rules of conduct, while also defining data ownership principles within the ecosystem. These elements will set the basis for the creation of secure and trusted relationships as an additional layer above a technically secure (system security and data security within the ecosystem), privacy-aware and trusted infrastructure. In addition, the business perspective also examines the design of intra-ecosystem reputation mechanism(s), creating this way reputation capital for the ecosystem users and its activities, which will enhance further the creation of trusted environment. Given the role of ethics in the formation of trust, the adoption of ethical practices (code of ethics) within a

crowd-driven ecosystem is critical for reinforcing its trustworthiness and conveying the sense of security that users need.

Finally, the main emphasis of the business perspective with respect to the “*motivation and engagement*” is primarily to motivate and engage the crowd and all different ecosystem stakeholders via the design of an incentive mechanism that will account for intrinsic and extrinsic motives. These incentives will influence the behaviour of individuals and organisational users within the ecosystem. As such offering the right incentives to each of the different actors will ensure the active engagement and motivation of users. A key variable for the success of this process is the acknowledgement of the ecosystem evolution and the user evolution within it. Thus, designing an incentive model that accounts for the dynamic evolution of the ecosystem as well as its users is critical in order not only to foster but also to maintain the crowd motivation and engagement throughout the ecosystem lifecycle. Additionally, depending on the type and focus of the crowd-driven ecosystem, one should also account for user interactions with each other within the system. Therefore, in order to facilitate user interaction and co-creation within such ecosystems, different design logic should be applied. This will enable and encourage users to network and collaboratively contribute in the innovation development process of the given ecosystem.

4 IoT Lab: An Innovative Crowd-Driven IoT/IoE Ecosystem

IoT Lab [35] project mainly focuses on the area of Internet of Things and crowd sourcing. It is developing a research platform that combines Internet of Things (IoT) testbeds together with crowd sourcing and crowd-sensing capabilities. IoT Lab aims to enhance the existing static IoT testbed infrastructures by utilising ad hoc crowd devices (smartphones, mobile/portable devices) creating a distributed crowdsensing IoT infrastructure as well as a crowdsourcing infrastructure leveraging the collective intelligence of the crowd. By doing so IoT Lab creates an open and collaborative ecosystem for crowd-driven research and experimentation, that enables a wide range of multidisciplinary experiments. As such, it enables researchers to exploit the potential of crowdsourcing and Internet of Things testbeds for multidisciplinary research with more end-user interactions.

On one side, IoT Lab approach puts the end-users at the centre of the research and innovation process. The crowd consists the core of the research cycle with an active role in the research from its inception to the results’ evaluation (crowd-driven research process) as shown in Fig. 2. It enables a better alignment of the research with the society, end-users needs, and requirements. On the other side, IoT Lab aims at enhancing existing IoT testbeds, by integrating them into a Testbed-as-a-Service (TBaaS) and by extending the platform adding crowd sourcing and crowd-sensing capabilities. To achieve such aims, the IoT Lab focuses its

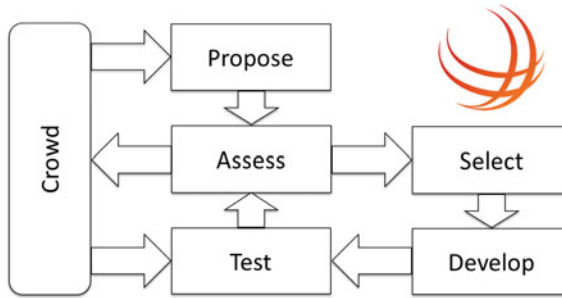


Fig. 2 IoT Lab Crowd-driven Research Process

research and development on the following areas: (a) Crowdsourcing and crowd-sensing mechanisms and tools; (b) Integration of heterogeneous testbeds; (c) Virtualisation of testbed components and integration into a Testbed as a Service; (d) Testing and validating the platform with multidisciplinary experiments; (e) Research end-user and societal value creation through crowdsourcing, and (f) “Crowd-driven research”.

IoT Lab follows a multidisciplinary approach and addresses important issues such as, privacy, and personal data protection through ‘Privacy by Design’ approach and built-in anonymity. The IoT Lab tools include the Testbed-as-a-Service (TBaaS) that enable, the researchers to create and manage their experiments and interact with all the available testbed and mobile resources, through a web interface. Regarding the interaction with the participants, an IoT Lab application has been developed, currently targeting the Android platform. Through this application, mobile resources from users provide both sensing information (crowdsensing), as well as, knowledge from the crowd (crowdsensing) by participating in different ongoing researches. These interactions include actions such as answering a questionnaire, annotating data, taking a picture, and sharing sensor data. The overall view of the IoT Lab as a Service is depicted in Fig. 3.

In order to analyse further the IoT Lab ecosystem, we adopt the multi-dimensional approach presented in Sect. 3. The sections that follow present an analysis of the key drivers and major challenges in its design and development of this crowd-driven IoT ecosystem.

4.1 Technical Perspective of the IoT Lab

The IoT Lab platform aims to reach beyond the notion of a static federation of IoT testbeds. More specifically, since the primal phase of its design, the role of the end-user, as well as the corresponding dynamics inferred were put in the forefront. The way the end-user would interact with the facility, the particular type of interactions supported, and the degrees of freedom the end-user would be provided by



Fig. 3 IoT Lab as a Service

the platform were carefully and thoroughly addressed. This direct interaction between the platform and its end-users posed important issues with respect to privacy, trust, and protection of the user-generated data. Another important aspect has been the design of mechanisms that successfully engage end-users into using and contributing to the platform. These aspects were considered and addressed on top of the various facility federation difficulties that have been addressed (although in a different context) by other federated facilities as well.

4.1.1 Key Technical Drivers for the IoT Lab Platform

One of the key drivers for IoT Lab has been the modern technology landscape as it has been defined after the introduction of truly portable and highly personal devices such as smartphones and smart wearables. These modern smart devices come with significant communication capabilities by supporting several wireless radio interfaces. Furthermore, the constantly evolving micro electromechanical systems (MEMS) technologies have not only changed the dimension and precision of sensors, but also their integration potential, allowing them to be installed in common mobile devices (like smartphones) and enhancing the user experience. Modern mobile phones ship with vivid screens, staggering photographic sensors, integrated GPS receivers, and a plethora of embedded context aware sensors. The increasing adoption rates of modern mobile devices by the public in conjunction with the development of smart services and the continuous evolution of advanced network technologies have paved the way for a new paradigm; namely the Mobile Crowdsensing Systems (MCS) that seek to exploit the embedded sensory capabilities of such devices carried by people in their everyday life.

In the MCS paradigm smart devices, such as smartphones, are not only regarded as data collection points but also as a direct means of interaction with their owners. In this context, the intrinsic integration of smartphone devices to the IoT Lab platform was regarded as the way to go in order to enable the facility to establish a two-way interaction scheme with its end-users. On one hand, it would enable the facility to opportunistically augment its sensing infrastructure via crowdsourcing. On the other hand, it would provide a direct and innovative way of interaction with its end-users, enabling them to provide input on their personal preferences and perception.

Apart from this novel perspective on the technical aspects of the IoT vision, the IoT Lab platform is also driven by the rapid expansion of the IoT domain that is foreseen for the coming years. Various business analysis estimate that the Internet of Things will be the largest device market in the world by 2019 and it will be more than double the size of the smartphone, PC, tablet and the wearable market combined [36]. However, currently 99.4% of physical objects that may one day become part of the Internet of Everything are still unconnected. Cisco estimates that there were about 200 million things connected to the Internet in the year 2000. Currently this number has been increased to approximately 10 billion [37]. IoT Lab seeks to leverage this infrastructure by integrating the corresponding networking technologies (LTE, Bluetooth, NFC, etc.) into a service platform that will be open and easily accessible to people and scientists.

4.1.2 Key Technical Challenges

Federating several existing experimenting facilities, each one with its own application focus and design choices under a common federated platform is already a difficult task that poses some key challenges. Although several similar efforts (i.e. federating already existing facilities) have already taken place successfully in the past, the vision of the IoT Lab to incorporate crowdsourced resources into an IoT meta-testbed demonstrates unique characteristics and, therefore, challenges. Fig. 4, the final federation architecture is depicted; in the following, we discuss the main issues that had to be addressed with a particular emphasis on privacy and trust for the crowd.

IoT Testbed Federation and Crowdsourced Resources

The main attribute of the IoT Lab architecture is modularity. Each individual experimenting facility that is about to be federated is treated as a standalone module whose details are obfuscated, via a virtualisation mechanism. In particular, each individual facility has been designed to address a different IoT domain. For example, the testbed of the University of Surrey was developed with a focus on algorithmic design and experimentation, while the testbeds of University of Geneva and the Computer Technology Institute are focusing on end-user applications for smart buildings. That said it is evident that these differences lead to significant diversions in terms of design choices, technologies used, and services provided. The virtualisation layer creates a diverse set of facilities capable of synergising with

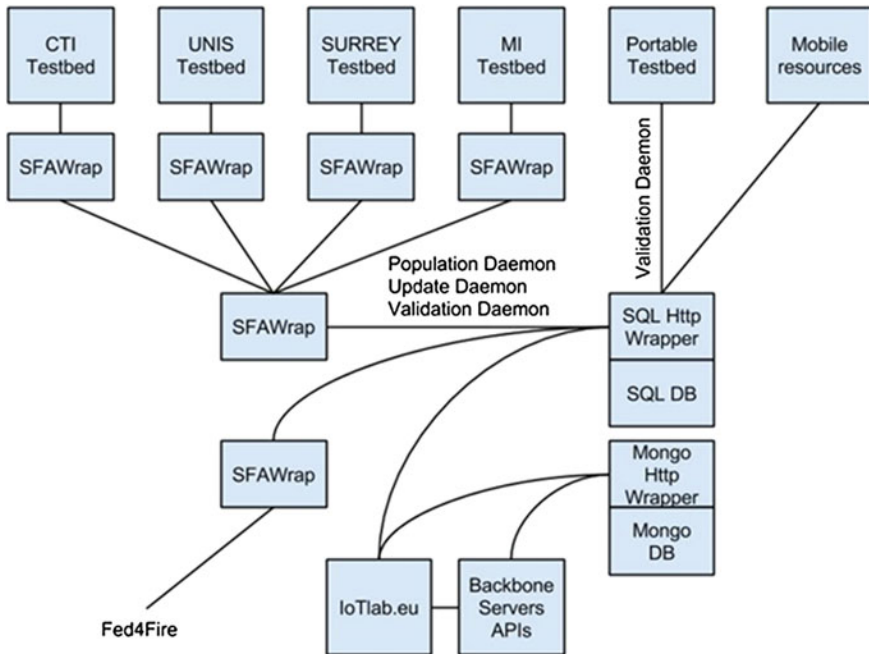


Fig. 4 The IoT Lab modular architecture

each other towards a unified platform. This is achieved via a commonly understood set of RESTful APIs used by the various modules, in order to communicate with each other. These APIs offer the services and the resources of each facility as services that can be consumed along with the necessary information needed in order to place these services in a common semantic context. For instance, each facility not only provides access to its resources (e.g. wireless sensor nodes and their measurements), but also provides information on the protocols via which this access can take place (e.g. HTTP or CoAP), the type of sensors (temperature, human presence, etc.), the units the measurements are taken (e.g. Celsius degrees, binary, etc.) and other. This information follows a predefined structure, namely the RSpec format (RSpec format, GENI), which also defines the database scheme of the Resource Repository of the platform.

The same virtualisation layer is also followed by crowdsourced resources (i.e. smartphones and wearables) provided by the end-users of the facility and the general public. The challenge here has been to address the ephemeral nature of the availability of such resources, their highly volatile numbers and their highly personal nature. These aspects have been taken under consideration when extending the RSpec format, which has initially been initially designed to support regular communication networks, and later, IoT resources. The particular characteristics of crowdsourced resources also posed significant challenges with respect to storing

and managing the collected data. Addressing these challenges led to the use of a hybrid database scheme, which combines both relational and non-relational databases. In particular, a relational database was used in order to serve as a Resource Repository; that is maintaining all necessary meta-information of the available resources such as location, type, owner, availability, participation in experiments, etc. A non-relational database was used in order to store and manage the actual collected data; given the big volume of collected data, this provided for better post-experiment data processing and more efficient data management. The information stored in the two databases is correlated, by having the two databases sharing some common keys, thus providing coherence to the entire set of stored information.

Privacy/Security/Trust/Ethics

In our design and implementation, we obtain, store, and use specific data that comes from users' smartphones. For this reason, we ensure a full compliance with general personal data and privacy protection rules. In this view, any unnecessary collection of personal data should be avoided and it is ensured that any collected data is to be handled in full conformity with the applicable good practices. Our research is interested in moving further by developing, a "privacy by design" architecture. In order to achieve this, we identified certain technical measures that maximise the privacy and anonymity of the participants as well as the protection of their data, and implemented them in a holistic platform.

To achieve privacy for end-users we issue credentials to eligible participants so that they can authenticate themselves, revealing only the information they want to reveal or, simply, prove only their eligibility to provide sensor data and nothing beyond this. Our design grants the ability of sending only the fixed user information, including device identification, as well as, a general-purpose data container to receive the gathered sensor data. To prevent remote dumping of personal data, access is available only to the smartphone's resources and data for which the user has given explicit approval. The platform enables the users to fully control and change their access preference parameters at any time, including the types of data that they are willing to share. All the participants have full control on their personal data, with the rights to access, modify, delete or hide it. The platform is designed in a way that provides collaborative crowd monitoring and control of ethical and personal data protection issues. Additionally, a flag mechanism is implemented in the smartphone application that enables users to characterise, according to their opinion, an experiment as "*aiming at an unethical objective*", in case they believe the experiment violates the adopted personal data protection rules.

Since, by design, the targeted use cases do not require users to give identifying information (e.g. they can register to the platform using only a pseudonym) anonymity is preserved. However, it is necessary to anonymise the users' devices since information sent by a mobile device may lead to user identification, in some cases. Two basic assumptions are made: (i) no SIM identifying information is sent over, and (ii) we do not consider techniques that masquerade a user's IP address when interacting with the platform. The former is a use case assumption since it is not required by mobile devices using SIM cards to transmit SIM related

information. The latter assumption is made because IP masquerading and IP anti-spoofing techniques are beyond the scope of our work.

With regard to data protection, we have taken steps to provide an *Authentication and Authorisation system*, as well as, ensuring data transmission and storage security. Persons in charge of user's data processing will be granted access electronically with special authorisation credentials and by giving them specific processing rights. These credentials will consist of an identity number and a password. Where authorisation profiles with different scopes have been defined for the persons in charge of data processing, an appropriate authorisation system will be used for access control. Authorisation profiles for each group of data processors with the same access rights will be defined and configured prior to the beginning of processing, in a way that allows access only to the data that are necessary for the processing. In order to ensure secure data transmission from users' mobile devices, we consider the deployment of TCP/IP security protocols for all data transmissions and connections. TLS/SSL protocol appears to be a good candidate to base our security solutions for all connections: Web applications connections (applications for experimenters and users to have access in their data), Android device connections (between the users' Android devices and the platform) and Testbed connections (for the interconnection between the testbeds). Furthermore, data storage is protected against any illegitimate access by external parties. We address this issue at several layers. The first layer enforces database access control, through a username/password based authentication mechanism, as well as, through imposing discrete roles for different user classes. At a second level, we implement a mechanism to disassociate data from their originators, i.e. the participating users. Finally, a data encryption layer is implemented. More specifically, we store users' personal and sensor data in separate files, with different access requirements and restrictions for each separate file. By doing so, there is no way that the sensor data (e.g. GPS position) and the socioeconomic profile (e.g. education level) of a user can be linked to the particular user through an identifying token (e.g. email), leading to lift the anonymity. In order to achieve this dissociation, the sensitive and anonymised data is stored in separate tables in a MySQL database. For additional security measurements, all database tables that store sensitive data are encrypted. Moreover, each time a new user is registered, a process with administrator privileges will decrypt the table, insert the user information and then encrypt the table again.

Motivation/Engagement

As previously mentioned, end-user motivation and engagement are crucial to crowdsourcing ecosystems. Different types of incentives are commonly used in these ecosystems in order to maximise the impact and use of the platforms. For the IoT Lab case, a combination of intrinsic and extrinsic types of incentives has been implemented. A Hybrid incentives model with gamification is followed, that allows researchers to allocate budgets for their researches. As part of the research description, the researcher is able to specify for each type of action the counterpart to be attributed to the participant, for performing such action, in the form of points. The **Incentives and Reputation framework** component of the platform, as the name indicates, is responsible for handling all the incentives-related functionalities,

that include, triggering the attribution of points when the participant fulfils a specific action. At the time a research finishes, it offers the choice to the participant to either exchange the collected points by vouchers, or to donate them to a specific charity. Those charities are selected from a list of registered and validated institutions in the platform. Another important functionality offered is the collection of badges by every user. These are acquired and collected by the users when fulfilling a specific task or achievement. The badges can be seen as an intrinsic form of incentives, but also serve as an important reputation mechanism that allows better classification and filtering of users.

4.2 *Business Perspective of the IoT Lab*

IoT Lab is an innovative crowd-driven IoT/IoE ecosystem that utilises the emerging participatory value creation model that is driven by users. It is an open, collaborative, user-driven, value-creating IoT ecosystem that explores the potential of crowdsensing (opportunistic and participatory sensing) and crowdsourcing to extend the IoT testbed infrastructure. In particular, IoT Lab aims to enhance the existing static IoT testbed infrastructures by utilising ad hoc crowd devices (smartphones, mobile/portable devices) creating a distributed crowdsensing IoT infrastructure as and crowdsourcing infrastructure leveraging the collective power and intelligence of the crowd. By doing so, IoT Lab creates an open, collaborative IoT ecosystem that assimilates smartphones with existing testbeds for crowd-driven research and experimentation that enables a wide range of multidisciplinary experiments.

The business perspective plays a key role in the design and development of the IoT Lab ecosystem as it facilitates the creation of a successful and sustainable hybrid crowd-driven network. In particular, it focuses upon the three key thematic areas presented above so as to ensure the creation of shared value with the given ecosystem. The sections that follow present the key drivers and challenges of the business perspective.

4.2.1 **Key Business Drivers for the IoT Lab Platform**

One of the key drivers for IoT Lab has been the *emerging market* that it addresses. In particular, IoT Lab explores the potential of a multidisciplinary, crowd-driven experimentation that amalgamates traditional IoT testbed infrastructures with ad hoc crowd devices, leveraging this way both the collective power and the collective intelligence (crowd capital) of the crowd. As such, understanding the demand side and identifying the market needs has been a critical aspect. The need for “crowd-sourcing driven research” has been addressed by integrating crowd capital across various experimentation phases (Fig. 5), such as: experiment conceptualisation, execution, analysis and commercialisation (optional phase).

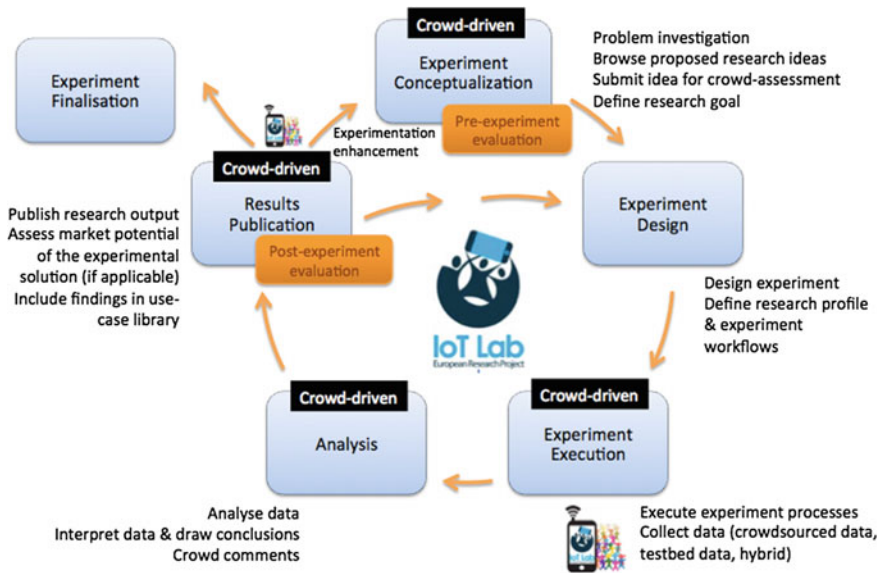


Fig. 5 IoT Lab experimentation life-cycle model

Another key driver has been the change from the users’ passive role, to the active co-creators and collaborators that in conjunction with the increasing interest and ability, due to the rise of smart devices and network technologies, in actively engaging with the crowd for particular purposes (i.e., ratings, task-oriented activities, problem solving, etc.). As such, the adoption of a *user-centric participatory model* has been instigated in numerous industrial sectors. This has been fueled by the Internet of Things (IoT), the emerging technological advancements and the smart personal devices (wearables, smartwatches, smartphones, etc.) that provide the ability to harnessing both the power, and the intelligence of the crowd that can now sense, observe, measure and make sense of real-world conditions. These driving forces have been critical for IoT Lab as it focuses upon the development of a crowd-driven IoT ecosystem that encompasses all these business-level innovations in a single infrastructure providing this way a greater scope for multi-dimensional experimentation.

4.2.2 Key Business Challenges

The aforementioned drivers and IoT Lab’s innovative approach provide one side of the business aspects related to this crowd-driven ecosystem. However, in order to allow a clear understanding of the implementation of the multi-dimensional model for such networks, it is vital to identify the challenges and complexities that impact their design and development.

Requirements/Needs

One of the challenges in the context of IoT Lab was the identification of market data. Lack of data, due to the emerging nature of the market that it addresses, not only hinders the ability to generate insight, but also creates a barrier to the identification of realistic market estimation. As a result of this difficulty, additional qualitative methods were utilised in order to understand the demand side and the needs and requirements of the primary users.

In addition, the creation and co-creation of value hinders a number of challenges. These are both related to the effective design as well as the management of this co-creation process [38]. Users should be provided with the necessary tools that will ensure both their connectivity as well as their ability to interact and co-create value at different levels. In the context of IoT Lab the different phases and types of crowd involvement have been defined (Fig. 3). This facilitated the design of the co-creation process and the introduction of the necessary tools (i.e., voting, assessment mechanisms, etc.) that will act as the co-creation infrastructure within this ecosystem. This infrastructure also serves as the basis for crowd interaction in the context of IoT Lab and therefore, it affects the quality of the co-creation experience between users and the ecosystem [39]. As such, by placing emphasis upon this infrastructure and its ability to create a variety of experiences for the IoT Lab users we invest in the unique value for each individual in the ecosystem, which will impact value co-creation within this ecosystem [39]. Nurturing and managing this value creation and co-creation process constitutes another key challenge. Following the DART model of value co-creation [39] we addressed this challenge in the context of the IoT Lab ecosystem. Despite the fact that this framework considers the “consumer-company interactions” we analysed its building blocks: dialogue, access, risk assessment, and transparency, in an open community environment governed by a non-profit association—the IoT Lab Association.

Emerging crowd-driven IoT/IoE ecosystems demand that we revisit not only how we create/co-create value, but also how we capture value. This constitutes another challenge for an open ecosystem such as IoT Lab. This is due to the fact that portions of value are captured by different entities in addition to the ecosystem itself. As such the identification of the appropriate business model for this crowd-driven ecosystem has been critical for its success. The identification of a business model that will account both for the value creation and the value capture elements [40] within the IoT Lab community has been critical. This is also related to the sustainability of the ecosystem, as the envisioned business model acts as the basis for the design of a sustainability model that will examine economic, community, and innovation views aiming to nurture an evolving ecosystem.

Privacy/Security/Trust/Ethics

Due to the fact that the IoT Lab ecosystem is based on crowdsourcing and crowdsensing principles, the acquisition, storage, and usage of crowd-driven data is central to its activities. As such in order to ensure compliance with personal data and privacy protection rules; a “privacy by design” approach has been followed. However, although ensuring the security and privacy within IoT Lab has been a critical aspect of its design and development process, conveying this assurance to

the crowd and the community of users creates a challenge. Reassuring users and all IoT Lab stakeholders about their data privacy and security is an aspect that significantly impacts the adoption of the ecosystem. Consequently, the establishment of the IoT Lab Association as well as introduction of policies that define among others data ownership principles within the ecosystem and rules of conduct facilitate this process. These parameters, pave the path for the creation of secure and trusted relationships and co-creation environment, as they act as an additional layer above a technically secure, privacy-aware, and trusted infrastructure.

Another business level challenge relates to the establishment of intra-ecosystem security and trustworthiness. That is security and trustworthiness at a micro-level (actor-level) that relate to the different stakeholders within the IoT Lab ecosystem. This challenge can be addressed with the design of an intra-ecosystem reputation mechanism that will increase the trustworthiness (reputation capital) of the individual actors and their activities, within the ecosystem (i.e., initiate a new crowd-driven research, etc.), which will enhance further the creation of trusted environment. Finally, transparency in the interactions between the IoT Lab Association (governance body) and the broader ecosystem community is critical for the formation of trust. In addition, the adoption of ethical practices (i.e., code of ethics) within the IoT Lab ecosystem will alleviate trust concerns of the user community and will convey a sense of security that users need.

Motivation-Engagement

Finally, another major challenge in the context of IoT Lab has been the design of an incentive mechanism that will motivate and create an active and engaged community of users. Offering the right incentives, for each actor while acknowledging the different phases of the IoT Lab ecosystem evolution, as well as, the evolution of the user himself within the system, has been a challenge. This implies the need for an incentive model that accounts for the dynamic evolution of the IoT Lab ecosystem and its users, in order to foster and maintain the crowd motivation and engagement throughout the ecosystem lifecycle. As such, updating the incentive model for the different phases of the IoT Lab is critical, as this will align with its actual community needs. Given the co-creation element of the IoT Lab special emphasis has been placed not only upon the design logic but also upon the incentives that will trigger users to network and collaboratively contribute in the crowd-driven research process.

4.3 People/End-User Perspective of the IoT Lab

In relation to IoT Lab, there are at least two main end-user groups that need to be taken into consideration in the process of developing the system. These are the crowd (i.e. the contributors) and the experimenter (i.e. the requesters). By means of IoT Lab, end-users (i.e. the crowd) can contribute to an experiment either initiated by the crowd or the experimenter community. Hence, IoT Lab must handle two different end-users' perspectives while ensuring a high level of utility, usability, and

usefulness of the platform. In this context, it is important to balance between the different drivers for each user groups while creating value for the whole eco-system. Hence, a symbiotic relationship between the three aspects of the crowd, the experimenters (requesters) and the IoT Lab platform becomes an important goal to reach. These different end-user groups are also motivated differently and thus, different incentive models focusing on creating value through the use of the IoT Lab platform should be applied.

For the end-user group of researchers, there are additional requirements that come into focus. The end-user groups' system requirements are highly connected to ensure that the collected data is reliable, trustworthy, and of high-standard quality. Moreover, the aforementioned requirements ensure that end-user groups have control over the data collection process, that the selection of respondents is easy and reliable, and that the contributors' privacy is ensured. This group of end-users is mainly motivated to use this type of system since the latter provides access to real world data that would be difficult to collect otherwise. It also enables researchers to combine new types of data and contexts, such as, levels of happiness related to a specific location. Furthermore, it enables users with limited, or lack of, technical background to perform sensor-based. Consequently, new data collection methods and research questions derive by the use this type of technology—questions that need to be tackled and answered.

4.3.1 Key People/End-User Drivers for the IoT Lab Platform

During the start-up of a crowdsourcing platform, it is important to identify the lead-users, or initiators and interest organisations, which can be motivated to be the first to participate and contribute to the crowdsourcing effort, i.e. IoT Lab platform. For these lead-users, it is important that their needs and drivers correspond to the aim of the experiments and the platform. In the case of the IoT Lab platform, the aim is focused on crowdsourced driven research, or citizen science initiated by the citizens. In the IoT Lab platform, the main driver for the crowd has been identified as a will to contribute to a better society and having fun while doing it. Societal contributions, in this case, can take different forms. It can be either through the incentive system, focusing on giving to charity, or it can be about the experiments such as contributing to a better society by, for instance, measuring air pollution or noise pollution in a specific area. Hence, the crowd is driven by idealistic ideals.

For the experimenter end-user group, drivers are somewhat different than the crowds' even though there is some overlap since both end-user groups want to create value for a common good in some sense. Among researchers, drivers, such as getting in contact with the real world and getting access to context data as they emerge, are the most prominent. IoT Lab also facilitates new interesting and challenging research areas and questions. These also become a driver for the researchers' group, as they desire to challenge established knowledge, while they explore new questions and situations.

4.3.2 Key People/End-User Challenges

When it comes to end-user challenges and the IoT Lab platform, there are several aspects that need to be considered and grapple with. We need to come up with a solution that creates value for both use sides, while offering an opportunity to the generic users to become more actively engaged in experiments and research projects and thus strengthen democracy itself. We need to stress the fact that, the experiments, their quality, and their scope are of utmost importance. Consequently, at this stage the added value of the crowds' contributions needs to be explicit and well defined. Additionally, in IoT Lab platform, the experiments must be crystal clear, engaging, and often divided into micro-tasks that are easily managed, since the crowd is characterised by limited available time and attention span. As a result, the challenges of the platform emerge on two layers: first on the usability layer of the platform and the second on the experiment layer that the crowd should contribute to. At this point, it is important to understand what users are motivated by in order to design tasks that the crowd wants to engage in.

In relation to challenges related to privacy and participatory sensing, aspects such as time, location, pictures, videos, sound samples, acceleration, environmental data, and biometric data are important and require special handle. For example, time and location is data acquired by many applications and due to their nature, these two modalities have shown to lead to privacy sensitive information about the end-users, including home and workplace locations, routines, and habits. When it comes to environmental data, for instance gas and particle concentration, it may not be a threat of privacy in itself, but when it is combined, for instance, with temperature identification of location down to a room level in a building is possible, which might invade privacy aspects in a workplace.

Based on this, we conclude that privacy threats are an inherent challenge of any participatory sensing application, especially when different sensors are combined. Hence, addressing privacy threats in this field is a multi-dimensional problem that needs to be considered and designed into crowdsourcing and IoT solutions. For instance, participatory sensing solutions should have functionalities that facilitate tailored sensing, anonymous task distribution, anonymous and privacy preserving data reporting, pseudonymity, spatial cloaking, data perturbation, hiding sensitive locations, and access control and audit. In previous studies [41–43] it has been revealed that fundamental research in the field of privacy related to participatory sensing is still in its infancy. Thus, challenges such as including participants in the privacy equation, providing adaptable privacy solution, trade-offs between privacy, performance, and data fidelity, making privacy measurable and defining standards for privacy research are important to investigate further. As of today, many end-users are rather naïve and unaware of what can be done with their data both in the primary and secondary usage of it. Hence, to succeed with crowdsourcing solutions that are ethical in their character, privacy must be protected without putting the responsibilities and efforts on the end-user side.

As a conclusion the table that follows presents the major challenges in the design and development of the IoT Lab crowd-driven ecosystem (Table 3).

Table 3 Challenges in the design and development of the IoT Lab crowd-driven ecosystem

Requirements/Needs	Technical perspective	Business perspective	People perspective
	<ul style="list-style-type: none"> • Resource heterogeneity (static, mobile, portable) • Testbeds integration • Modularity • Overall performance and scalability • Federation of testbeds • Integration of complex databases and Resource Directory • Scalability of platform in terms of Mobile Users and IoT Resources • Degree of Freedom for experimenters • Simultaneous direct requests to database • Simultaneous requests to Resources via main platform server. • Localizing the resources 	<ul style="list-style-type: none"> • Market needs and data creation and co-creation of value • Quality of co-created value • Capturing value in an efficient way • Ecosystem sustainability 	<ul style="list-style-type: none"> • Engaging projects for the crowd to contribute to • Quality of data • Ease of use of the system • Usefulness of the system • Ease of learning • Satisfaction • Ease of remembering
Privacy/Security/Trust/Ethics	<ul style="list-style-type: none"> • Privacy by design • Right to be forgotten • Disassociation of data to its owner • Physical location (country) of server and main database due to legal issues 	<ul style="list-style-type: none"> • Reassuring users about the ecosystem privacy and security • Trusted environment & relationships/interactions • Intra-ecosystem reputation mechanisms 	<ul style="list-style-type: none"> • Perceived trustworthiness of the system • Opportunistic sensing (coming from people) and information security <ul style="list-style-type: none"> • Data from sensors is involuntary • No control over data collection (what, when, where) raises serious privacy concerns • Participatory crowdsourcing <ul style="list-style-type: none"> • Voluntary data sharing method—individual chooses what she/he wants to report to the system

(continued)

Table 3 (continued)

	Technical perspective	Business perspective	People perspective
Motivation/Engagement	<ul style="list-style-type: none"> ● Intrinsic motives (badges, etc.) ● Extrinsic motives (money, coupons, etc.) ● Implementation of incentive mechanisms—money versus badges 	<ul style="list-style-type: none"> ● Ecosystem and crowd evolution ● Maintain engagement and motivation 	<ul style="list-style-type: none"> ● Minimal privacy concerns but no control after reporting ● Important information security, integrity and availability to correct stakeholders ● Understanding what motivates the crowd and their participation ● Transparency in information security important for trustworthiness of the system and participation rate ● Engage the crowd ● Identify the right crowd ● Engage crowd over a longer period of time ● Transparency of data collection and use ● New type of research methods

4.4 IoT Lab Crowd-Driven Ecosystem Scoring

The examination of these distinct perspectives (*people-centric, technology-centric and business-centric perspectives*) in the context of IoT Lab enabled us to identify and describe key issues that were critical for the design and development of such an innovative crowd-driven IoT ecosystem. In order to extend our analysis further, we conducted a comparative analysis of each of the three perspectives under examination by introducing the “*Crowd-driven Ecosystem Index (CEI)*”. CEI measures the extent to which a holistic, multi-dimensional approach has been utilised in the design and development of a crowd-driven ecosystem initiative in the context of IoT/IoE; conveying this way its potential propensity of success (Table 4). The CEI is a quantitative measure, that acts as a self-evaluation tool and which varies between 0 and 1 and obtains its maximum value when all parameters and key thematic areas presented above exhibit highest coverage intensity. High values (CEI = 1 – 0.67) indicate that emphasis has been placed upon the different ecosystem elements, indicating a relatively high success potential of the ecosystem; moderate values (CEI = 0.66 – 0.33) indicate moderate potential; and low values (CEI = 0.32 – 0) indicate low potential.

The overall IoT Lab evaluation indicates that a balanced approach has been followed in the design and development of the ecosystem with a CEI of 0.87 (Fig. 6). The comparative results from each perspective indicated that although high coverage intensity has occurred, the values vary slightly between the different perspectives and the emphasis placed on each. As such “motivation and engagement” seems to be the thematic area with the highest coverage intensity followed by requirements and user utility. This is the case for both the people and business perspectives showing clearly the differences between the different perspectives.

Table 4 Crowd-driven Ecosystem Index (CEI)

Thematic areas	Perspectives			Thematic areas score
	Technical	Business	People	
Requirements/needs	5	4	4	13
Privacy/security/trust/ethics	4	4	4	12
Motivation/engagement	4	5	5	14
Perspective Scoring ^a	13	13	13	

^aScoring scale: 1–5, depending on the coverage intensity of each factor: (1) low coverage intensity, (2) moderate low, (3) moderate, (4) high and (5) extreme coverage intensity

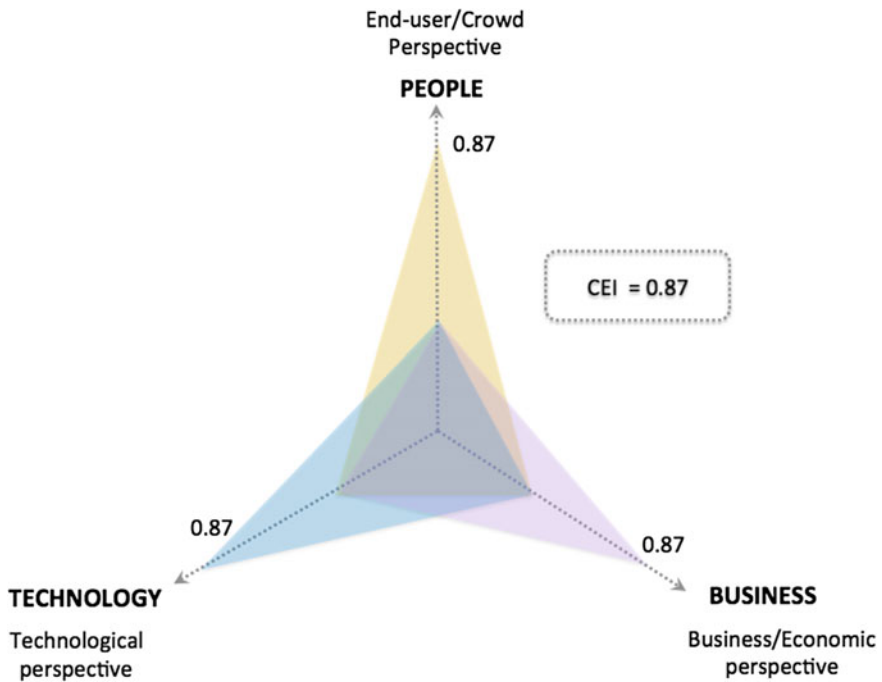


Fig. 6 The IoT Lab CEI

5 Conclusions

Crowd-driven ecosystems serve as a global meta-environment for leveraging the network effects, in order to harness the collective and distributed power and intelligence of our society. These open, collaborative, user-driven, and value creation ecosystems enable individuals to collaborate in innovative ways. Participants openly share their ideas, data, best practices, and create new knowledge that enhances our common innovation potential. This ability to exploit the capacity of the crowd has been fuelled by the Internet of Things (IoT)/Internet of Everything (IoE), introducing new user-centric paradigms, such as mobile crowd sensing (MCS). MCS goes beyond traditional sensing techniques (e.g., sensor networks, etc.,) leveraging both the power and the wisdom of the crowd in order to sense, observe, measure and make sense of real-world conditions (e.g., environmental, etc.,), and activities (e.g., personal activities and interactions, etc.,) using user-owned mobile and wearable devices. Such *crowd-driven IoT/IoE ecosystems* can exist in many forms concentrating on specific crowd-driven functions (crowdsourcing, crowdsensing, crowdfunding, etc.,) or they can be hybrid not tied to a specific mode.

Despite the promising participatory value creation paradigm and the numerous advantages it provides, crowd-driven IoT/IoE ecosystems are still in their initial

stages and face many challenges. One of the aforementioned challenges of these ecosystems relates to their design and development, acknowledging that their nature necessitates the adoption of multi-disciplinary perspectives. The existing literature emphasises the technical aspects for the development of such networks and provides useful insights of what needs to be addressed. However, the directions on the way one should design and undertake this process—accounting for non-technical ecosystem elements are limited. Consequently, there is a need for a unified framework that embraces a holistic approach to address different parameters, crucial for the design and development of such crowd-driven ecosystems. Aiming to fill this gap, this chapter provided a framework that adopts a holistic multi-perspective approach that facilitates the design and development of crowd-driven ecosystems. Our model provides three perspectives: (a) *the people-centric perspective* that encompasses the crowd views and needs for the creation and co-creation of value within a given network; (b) *the business-centric perspective* that emphasises the creation of economic and business value of a given crowd-driven ecosystems—factors that affect the network sustainability—and (c) *the technology-centric perspective* that focuses on the technological aspects (e.g., ecosystem architecture and components, technological resources, etc.) relevant to the design and development of a crowd-driven network.

The examination of this model has been implemented in the context of a hybrid crowd-driven IoT/IoE ecosystem, namely *IoT Lab* (which integrates both crowdsourcing and crowdsensing elements). *IoT Lab* is an innovative crowd-driven IoT/IoE ecosystem that utilises the emerging participatory value-creation model and explores the potential of crowdsensing (opportunistic and participatory sensing) and crowdsourcing to extend the existing IoT testbed infrastructure. *IoT Lab* is the first experimenting facility designed to not only federate several IoT testbeds, but also to intrinsically integrate crowdsourced devices as experimenting resources. This new type of resources posed new, non-trivial challenges, which have not been addressed in the past in the context of experimenting facilities. In particular, their highly personal nature (i.e. the fact that each device is owned by a person) has raised significant reliability and availability issues. This type of resources cannot be provisioned as traditional IoT resources, as the owner of the device needs to consent to their use in the context of an experiment. This way, a new ecosystem of interactions has emerged in which an experimenter, a testbed provider, and the devices' owners need to synchronise with each other, while providing the necessary guarantees. For instance, on one hand the experimenter and the testbed provider need to provide sufficient guarantees to the crowd with respect to privacy, security, and trust, while on the other hand, the device owner needs to be trustworthy, in terms of availability.

This triangle of relations' paves the way for new modes of value creation and value capture within such ecosystems, as well as, new sources of value creators and co-creators. This, however, also hinders a number of challenges related to the effective design and the management of this co-creation process. However, the emerging crowd-driven IoT/IoE ecosystems, such as *IoT Lab*, demand that we not only revisit the way one creates/co-creates value, but also, how value is captured.

This constitutes another challenge for open ecosystems such as IoT Lab. This is due to the fact that portions of value are captured by different entities in addition to the ecosystem itself. Hence, the identification of the appropriate business model for this crowd-driven ecosystem has been critical for its success and sustainability. In addition, many challenges need to be tackled, in order to keep people's motivation and engagement towards this type of platform. Thus, by understanding the incentivising and the driving forces of people will allow us to develop innovative engaging techniques. This will help us to motivate people to contribute on our projects on regular basis.

Our examination of these distinct perspectives (*people-centric, technology-centric and business-centric*) in the context of IoT Lab enabled us to identify and describe key issues that are critical for the design and development of such innovative crowd-driven IoT/IoE ecosystems. This has been further extended with the introduction of the "*Crowd-driven Ecosystem Index (CEI)*", which measures the coverage intensity of each of the key ecosystem parameters, denoting this way the propensity of success of a given crowd-driven network. Our comparative analysis in the context of IoT Lab indicated that a balanced approach has been followed ($CEI = 0.87$) with relatively high coverage intensity of the model parameters.

Some future research areas, from a technical perspective, include the use of novel privacy preserving mechanisms in the context of crowdsensing (e.g. differentially private mechanisms), providing the services and the technological enablers to support new economic models (e.g. schemes of open and collaborative economies) as well as, providing services that would further leverage the usage of crowdsourced infrastructure (e.g. in network data processing, in-memory databases for smartphones, etc.). In addition, further empirical validation of the proposed model would be appropriate in order to fine tune its key thematic areas and account for additional micro-level factors. This could also be extended with an evaluation tool that will accompany the model and provide additional guidance in relation to the design and development of crowd-driven ecosystems in the area of IoT/IoE.

Acknowledgment The research reported in this paper has been supported by the EU/FIRE IoT Lab project—STREP ICT-610477.

References

1. IEEE-IoT 2015. Towards a definition of the Internet of Things (IoT). Revision No. 1—Published 27 May 2015.
2. J. Bradley, J. Barbier, and D. Handler D. Embracing the Internet of Everything to capture your share of \$14.4 trillion: More Relevant, Valuable Connections Will Improve Innovation, Productivity, Efficiency & Customer Experience. White Paper Cisco, 2013.
3. J.A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy; et al., Participatory sensing. Center for Embedded Network Sensing. 2006. UCLA: Center for Embedded Network Sensing.
4. R. K. Ganti, Y. Fan and L. Hui, Mobile Crowdsensing: Current State and Future Challenges, IEEE Commun. Mag., 49(11), 2011, pp. 32–39.

5. A.G. Tansley. The use and abuse of vegetational concepts and terms. *Ecology*, 16, 1935, pp. 284–307.
6. World Resources Institute (WRI). World Resources 2000–2001: People and ecosystems: The fraying web of life. Report Series by United Nations Development Programme, United Nations Environment Programme, World Bank and World Resources Institute - September 2000.
7. R.A. Frosch, and N.E. Gallopoulos, Strategies for Manufacturing. *Scientific American*, 261 (3), 1989, pp. 144–152.
8. J.F. Moore, Predators and prey: A new ecology of competition. *Harvard Business Review*. 71 (3), 1993, pp. 75–83.
9. J. F. Moore, *The Death of Competition: Leadership & Strategy in the Age of Business Ecosystems*. 1996, New York, Harper Business.
10. H.W. Chesbrough and M.M. Appleyard, Open Innovation and Strategy. *California Management Review*, 50(1), 2007, pp. 57–76.
11. P. Almeida and B. Kogut, Localization of knowledge and the mobility of engineers in regional networks. *Management Science*, 45, 1999, pp. 905–917.
12. R. Baptista, Clusters, innovation and growth: a survey of the literature. In: G. M. P. Swann, M. Prevezer and D. Stout, eds. *The dynamics of industrial clusters: international comparisons in computing and biotechnology*, 1998. Oxford: Oxford University Press 13–51.
13. D. Bray, Knowledge Ecosystems. In *Organizational dynamics of technology-based innovation: Diversifying the research agenda*, 2007 (pp. 457–462). Springer US.
14. T. Coughlan, Enhancing Innovation through Virtual Proximity. *Technology Innovation Management Review*, 4(2), 2014, pp. 17–22.
15. B. Clarysse, M. Wright, J. Bruneel, and A. Mahajan, Creating value in ecosystems: Crossing the chasm between knowledge and business ecosystems. *Research Policy*, 43(7), 2014, pp. 1164–1176.
16. G. Koening, Business Ecosystems Revisited. *Management*, 15(2), 2012, pp. 208–224.
17. K. Valkokari, Business, Innovation, and Knowledge Ecosystems: How They Differ and How to Survive and Thrive within Them. *Technology Innovation Management Review*, 5(8), 2015, pp. 17–24.
18. M. Iansiti and R. Levien, *The Keystone Advantage: What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability*. Boston, MA: Harvard Business School Press, 2004.
19. J.B. Andersen, What Are Innovation Ecosystems and How To Build and Use Them. *Innovation Management*, 2011.
20. M. Wright, Academic entrepreneurship technology transfer and society: Where next? *Journal of Technology Transfer*, 39(3), 2013, pp. 322–34.
21. E.G. Carayannis, Innovation, Technology, and Knowledge Management, 2010.
22. S.M. Lee, D.L. Olson and S. Trimi, Co-Innovation: Convergencomics, Collaboration, and Co-Creation for Organizational Values. *Management Decision*, 50(5), 2012, pp. 817–831.
23. D. Tapscott and A.D. Williams. *Wikinomics: How mass collaboration changes everything*. 2008, Penguin.
24. M. E. Porter and M. R. Kramer, Creating Shared Value, *Harvard Business Review*, 89(1–2), 2011, (January–February).
25. R.K. Rana, C.T. Chou, S.S. Kanhere, N. Bulusu, and W. Hu, Ear-Phone an End-to-End Participatory Urban Noise Mapping System. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, Stockholm, Sweden, 12–16 April 2010, pp. 105–116.
26. N. Thepvilojanapong, T. Ono, and Y.A. Tobe, Deployment of Fine-Grained Sensor Network and Empirical Analysis of Urban Temperature. *Sensors*, 10, 2010, pp. 2217–2241.
27. Ludwig, T., Siebigteroth, T., and Pipek, V., 2014. CrowdMonitor: Monitoring Physical and Digital Activities of Citizens During Emergencies. *Social Informatics*, 8852, pp. 421–428.
28. Bélanger, F. and Crossler, R. E. 2011. Privacy in the Digital Age: A Review of Information Privacy Research in Information Systems. *MIS Quarterly*, Vol. 35, No. 4, pp. 1017–1041.

29. Hong, W. and Thong, J. Y. L. 2013. Internet Privacy Concerns: An Integrated Conceptualization and Four Empirical Studies. *MIS Quarterly*, 37(1), pp. 275–298.
30. Muhdi, L. and Boutellier, R. 2011. Motivational Factors Affecting Participation and Contribution of Members in Two Different Swiss Innovation Communities. *International Journal of Innovation Management*, 15(3). pp. 543–562.
31. Brabham, D. C. 2010. Moving the Crowd at Threadless. *Information, Communication & Society*, 13(8), pp. 1122–1145.
32. Kaufmann, N., Schulze, T. and Veit, D. 2011. *More Than Fun and Money. Worker Motivation in Crowdsourcing – a Study on Mechanical Turk*. AMCIS2011.
33. Nov, O., 2007. What Motivates Wikipedians. *Communication of the ACM*, 50(11) pp. 60–64.
34. Stählbröst, A., Angelopoulos, C. M., Evangelatos, O., Krco, S., Nikolettseas, S., Raptis, T. and Ziegler, S., 2015. *Understanding Modes of Crowdsourcing and Related Crowd Motivators*. XXVI ISPIM conference, Budapest, Hungary.
35. IoT Lab project. Available at (access: 20.01.2016): www.iotlab.eu.
36. Greenough, J., 2014, How the Internet of Things market will grow, Business Insider. Available at (access: 25.01.2016): <http://uk.businessinsider.com/how-the-internet-of-things-market-will-grow-2014-10?r=US&IR=T>.
37. Dave Evans, The Internet of Everything – Cisco. Available at (access: 25.01.2016): <https://www.cisco.com/web/about/ac79/docs/innov/IoE.pdf>.
38. Payne, A.F., Storbacka, K. and Frow, P., 2008. Managing the co-creation of value. *Journal of the academy of marketing science*, 36(1), pp. 83–96.
39. Prahalad, C. K. and Ramaswamy, V., 2004. Co-creating unique value with customers. *Strategy & leadership*, 32(3), pp. 4–9.
40. Chesbrough, H., 2007. Business model innovation: it's not just about technology anymore. *Strategy & leadership*, 35(6), pp. 12–17.
41. Christin, D., Reinhardt, A., Kanhere, S. and Hollick, M. 2011. A Survey on Privacy in Mobile Participatory Sensing Applications. *The journal of systems and software*, 84. pp. 1928–1946.
42. Huang, K. L., Kanhere, S. S. and Hu, W. 2010. Preserving Privacy in Participatory Sensing Systems. *Computer Communications*, 33(11). pp. 1266–1280.
43. Weber, R. H., 2010. Internet of Things – New Security and Privacy Challenges. *Computer Law & Security Review*, 26(1). pp. 23–30.

Improving Quality of Life with the Internet of Everything

**Despina T. Meridou, Maria-Eleftheria Ch. Papadopoulou,
Andreas P. Kapsalis, Panagiotis Kasnesis, Athanasios I. Delikaris,
Charalampos Z. Patrikakis, Iakovos S. Venieris
and Dimitra I. Kaklamani**

Abstract The advent of the Internet of Everything, where things and data providers can connect not only to other things and data providers, but to human entities as well, and are enriched with intelligence, calls for sophisticated data handling, storing, and sharing mechanisms. In this chapter, we present an ecosystem built over the idea of Internet of Everything and featuring a collaborative, intelligent service bus, which gathers information from connected devices and uses it to improve quality of life. All aspects related to data collection, processing, protection of privacy, as well as collaboration between users and interfacing between humans and devices are addressed, while mechanisms supporting decision making towards health related goals are presented.

1 Introduction

The vast variety of smart entities, from mobile applications to wearable devices to smart home appliances, has started to surround us in our everyday life. This leads to an enormous amount of generated data that gets difficult to handle and take advantage of as it gets bigger and bigger. The interconnection of these entities, or “things”, in an Internet of Things, where exchange of information and combination of knowledge is possible, results to an even more puzzling pile of information. The convergence of human and things related activities and generated data into a single ecosystem capable of improving efficient operation of machines and processes, and supporting to human welfare leads to the next step: the Internet of Everything (IoE).

D.T. Meridou (✉) · M.-E.Ch.Papadopoulou · A.P. Kapsalis · P. Kasnesis
I.S. Venieris · D.I. Kaklamani
School of Electrical and Computer Engineering, National Technical University
of Athens, Heron Polytechniou 9, 15773 Athens, Greece
e-mail: dmeridou@icbnet.ece.ntua.gr

A.I. Delikaris · C.Z. Patrikakis
Department of Electronics Engineering, Technological Education Institute
of Piraeus, Petrou Ralli & Thivon 250, 12244 Egaleo, Greece

The Internet of Everything is spreading in all kinds of industries, from home equipment to car manufacturing, education, and healthcare. This chapter is focusing on the idea of leveraging the Internet of Everything to improve the well-being of a person. In this respect, an intelligent service-based platform provides its end users (patients or non-patients), healthcare professionals, and government health services the means to exchange information, collaborate and get advice and recommendations: the Wellbeing Service Bus (WSB). The WSB is a collaborative platform based on a social network and it is built on top of the Enterprise Service Bus (ESB) paradigm [1]. It is able to aggregate health and well-being data for a particular user/patient in raw format, as those used by code systems and standards, such as LOINC (Logical Observation Identifiers Names and Codes; <http://loinc.org>). Data are collected from smart mobile applications, wearable devices, smart domestic appliances and other systems (e.g., Decision Support Systems). They can be either generated through embedded sensors or inferred by combining information coming from different data sources via machine-to-machine communication. Upon entering the platform, data transforming services are responsible for the conversion of data to RDF, a semantic language that gives meaning to raw data. At a later stage of data processing, semantic reasoning rules are applied to the RDF data. As a result, new chunks of information, implied indirectly rather than stated in a straightforward way, are inferred. Semantic data structures are used for storing the RDF data and the inference rules (quad and triple stores, respectively). Due to the sensitive nature of health-related data, sophisticated security and privacy mechanisms are applied throughout the lifecycle of the data.

The above architectural characteristics require a scalable and extendable resource infrastructure. The amount of collected data as well as the number of devices sending data to the platform can be extremely large. This creates the need for mechanisms that guarantee the availability of the services and the integrity and security of the collected data. The best approach is offering the Wellbeing Service Bus platform as a service (PaaS) over a Cloud Infrastructure (IaaS). Modern Cloud infrastructures can provide effective load balancing and elastic resource management that responds to peak demands and allows for easy distributed data processing clusters. Also, by default, most modern Cloud IaaS stacks offer sophisticated isolation, virtual routing and firewall mechanisms in order to secure virtual instances. This feature is valuable in the case of the Social Avatar network, so that sensitive data stored in the Cloud are protected.

1.1 Motivation

Advances in wearable technology have introduced a variety of devices, capable of measuring and reporting a wide range of health and wellbeing related parameters, providing information about daily activities, possible health risks and even personal coaching towards fitness or healthy life goals. Frequently, though, information produced by a single device (or any other object) is one-sided; they do not combine

the data they produce with other pieces of information. For instance, fitness devices may suggest to their users to engage in intense physical activity without taking into consideration any heart conditions they might suffer from [2]. In this respect, the need for a paradigm that not only integrates information coming from various sources, but also processes it together and extracts safe conclusions, is identified. At the same time, this plethora of medical and well-being data that are available on the fly could be shared directly with the data owner's personal doctors, dietitians and trainers, allowing for timely reaction to severe conditions and remote provision of advice towards a healthier lifestyle.

These requirements are addressed by a single health-related platform that comes with its own social network.

1.2 Related Work

Up to this day, there is a wide variety of systems and applications that process health and lifestyle data coming from wearable devices; each one of them serves a different purpose. Similar functions to the WSB are offered by web applications, mobile applications and service-based platforms communicating with wearable devices; each to a certain extent.

SuperTracker [3] is a web application offered by the Department of Agriculture of the United States. Users can browse through the wide selection of daily meal suggestions, create their own dietary plans and combine them with physical exercise. *SuperTracker* also allows its users to set a number of goals with respect to their fitness status and tracks their progress into achieving the aforementioned goals by processing their daily food intake and physical activity. Finally, it gives access to a number of charts informing the user about changes on their weight over time, the physical activity they have undertaken in a given time period and the amount of food they have consumed per food group (grains, vegetables, fruits, dairy, protein foods and oils). A more restricted set of functions similar to those of *SuperTracker* is offered by *eat this much* [4], a web application that creates meal plans based on the user's diet goals. It gives the user quite many degrees of freedom with respect to the number of meals per day, the food ingredients (e.g., in case of personal preferences, allergies, or limitations, such as those inflicted by diabetes) and the budget per meal.

Glooko [5] is a platform that integrates with fitness trackers, biometric devices and applications with the purpose of providing useful insights to diabetes patients. It integrates data related to the glucose trends of a patient with their carbohydrate intakes, their insulin dosage and the duration of their daily exercise aiming at monitoring their health status and at providing adherence reminders. *Sen.se* [6] is an open platform that allows for combining and correlating data from different data sources so that joint processing and decision making is possible. In this respect, *Sen.se* provides several tools that gather sensor data, processes them and executes a special function real-time. *PolyglotHIS* [7] is a smart agent-based polyglot solution

for healthcare data management. It addresses the issue of combining data of multiple disparate data sources residing even in a single institution by delivering a system that utilises databases of different types at once according to need. PolyglotHIS facilitates the application of queries by end-users, such as doctors and nurses, to the integrated set of healthcare data aiding the diagnosis process of medical professionals.

Apart from web applications and data-mashup platforms, several health-related service-based systems have been presented, as well. IBM's solution for Healthcare enables the integration of heterogeneous data through an Enterprise Service Bus named *IBM Enterprise Service Bus for Healthcare* [8, 9]. This ESB platform allows outdated legacy systems of hospitals, medical centers, pharmacists and other medical institutions and professionals to communicate seamlessly, facilitating the construction of a thorough patient profile. The IBM ESB for Healthcare leverages a healthcare-specific Common Information Model (CIM), which defines a common vocabulary, allowing health-related disparate systems to exchange information. The *Health Service Bus* [10] is another health-related platform based on the Enterprise Service Bus paradigm. Data homogenization is accomplished by means of a translation service, which is responsible for transforming all health-related data in XML format. During the transformation process, from and to the standards HL7 V2, HL7 V3 and OpenEHR, an ontology-based mapping tool, named OWLmt, is used.

The above-described solutions are designed to solve a specific issue of the healthcare domain. Although the purpose each one serves may differ when each solution is compared to one another, their functions are all relevant to those of the Wellbeing Service Bus presented in this chapter. The IBM ESB for Healthcare and the Health Service Bus are both ESB-based solutions, as happens with the Wellbeing Service Bus. While the platform presented here is leveraging ontologies in the context of data transformation and inferencing, IBM's solution is using a UML-based data representation format named Common Information Model (CIM), whereas the Health Service Bus fosters ontologies solely in the data transformation process. Glooko [5] enables diabetes management, which could be connected to the Wellbeing Service Bus for joint decision making. Similarly, the health-related data gathered by Sen.se could serve as a valuable data source for the WSB. Finally, the diet plan functionality offered by the Wellbeing Service Bus could be combined with the functions of *SuperTracker* and *eat this much* in the context of a goal-based collaborative scenario.

The rest of this chapter is organized as follows. Section 2 presents our view of a health-related Internet of Everything and briefly describes the tools leveraged towards data and device interoperability in this respect. In Sect. 3, the Wellbeing Service Bus, an ontology-based platform following the Enterprise Service Bus paradigm, is presented. In the same section, the functionality of its components is also depicted, and a detailed description of the security mechanisms that are adopted and the cloud infrastructure that supports the platform is provided. Section 4 presents the Social Health Avatar Network, a social network that allows people to communicate anonymously with each other, seeking medical advice or social support. In Sect. 5, the Wellbeing Service Bus is applied in a welfare use case, where a

clear goal of weight loss has been set. Finally, Sect. 6 provides a brief evaluation of the infrastructure on which the WSB is based, while Sect. 7 concludes this chapter.

2 A Health-Related Internet of Everything

Cisco reports on an Internet of Everything connecting not only people, machines, computers and devices, but also home appliances, garments and medical supplies; any object, in general, that is big enough to carry a sensor. It is expected that more than 50 billion devices will be connected in an Internet of Everything by 2020 [9], bringing overwhelming advantages to the public sector, especially in the fields of Healthcare, Education, Transportation and citizens' Safety. In this chapter, we are focusing on a health-related Internet of Everything, where end-users, medical professionals, wearables, smart devices and other smart objects collaborate towards the improvement of life.

2.1 Who and What Constitutes the Health-IoE

According to Cisco, the Internet of Everything brings together *people*, *process*, *data* and *things* [11]. Figure 1 depicts the four components of the IoE, as they are addressed in our health-related platform.

People represent the group of end-users (i.e., data owners), medical professionals, such as doctors, nurses and pharmacists, dietitians and trainers (i.e., data users). As the Internet evolves, connected devices allow a doctor to be fully aware of their patients' medical status; a nurse to monitor the vital signs of a patient and collect their data without actually being at the patient's bedside; a person to swallow a smart RFID-enabled pill, which then sends information relevant to the acidity, pressure, temperature, and digestive activity of their intestines to their doctor. In addition, the presented platform allows people to organize in a health-focused social network, the *Social Health Avatar*, in order to keep their doctors up-to-date with their health status via tweet-like posts, monitor the condition of their loved ones or connect with people facing similar health issues.

The set of *Things* is made up of physical objects, such as sensors, smart applications running on mobile devices or smart appliances, such as smart scales and refrigerators. In the Internet of Everything, things produce data and not only do they transmit them to a central receiver, but also they exchange the underlying information with other things or people via the Internet. In other words, things are able to work together towards a common goal. This way, the issue of smart devices making contradicting suggestions to the user, such as advising a person with an elevated heart rate to engage in intensive exercise, is tackled.

So far, IoT data follow a log-like structure; it is plain raw data that are collected via a hub and analysed appropriately. In IoE, *Data* are enriched, as things gradually

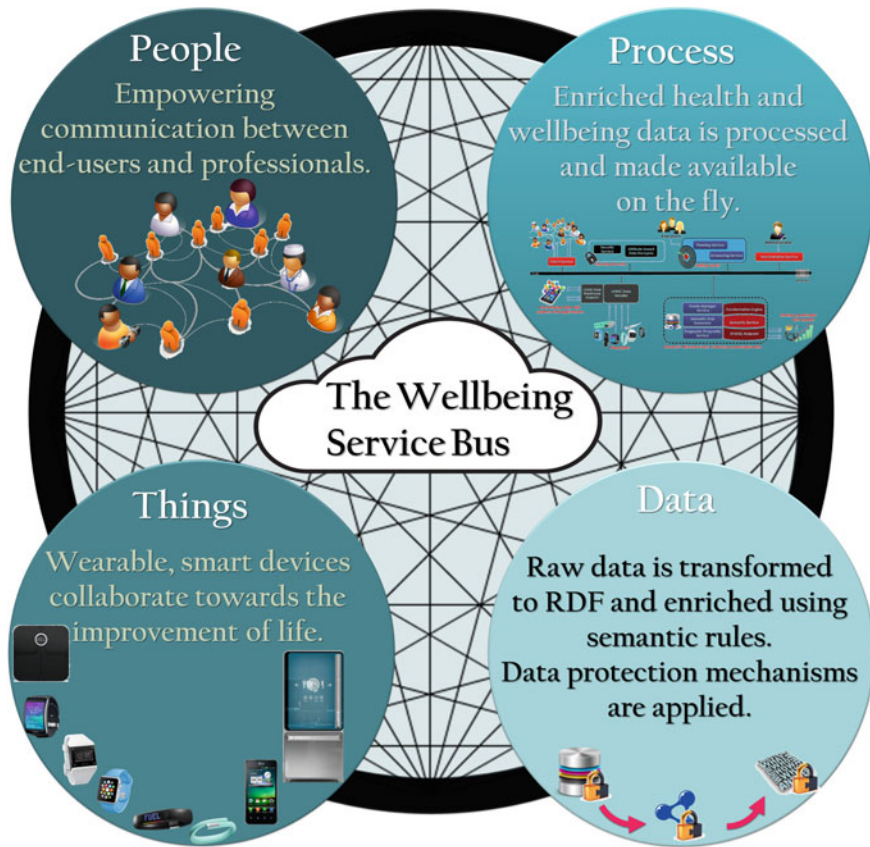


Fig. 1 The Internet of Everything in Health: a cloud- and ontology-based Wellbeing Service Bus

become context-aware and follow certain logic. In the Wellbeing Service Bus, raw, mostly XML-based data are semantically annotated through the use of ontologies. The semantic data, regardless of their content, are grouped under the same user ID, which makes it easier to perform more personalized processing. Semantic rules are then applied to the transformed data, resulting in the generation of reasoned data, which, utterly, go through a stage of further processing. Through all the aforementioned stages of data processing and analysis, strong security mechanisms and cryptographic tools used in the platform ensure privacy protection of the end users' sensitive data.

The fourth and last component of the IoE, *Process*, guides the interaction of the rest of the IoE components, such that they work in harmony towards providing added value. Well-designed workflows lead to the transformation, enrichment and dissemination of wellbeing data, allowing, on one hand, for the timely prevention of various kinds of health conditions by providing recommendations on a user's lifestyle, and on the other hand, for instant notification of doctors and relevant specialists and in-time reaction to incidents.

By defining a proper IoE environment, we aim at addressing important challenges in the sector of Healthcare, especially those that concern home recovery and remote treatment.

2.2 *Data Interoperability*

The heterogeneity of the devices and smart appliances constituting the set of things in the IoE environment introduces the need for the transformation of the raw health data they produce into understandable, structured and standardised healthcare messages. Thus, for the standard-based message exchange between things and the WSB platform, the use of the emerging HL7 standard named as Fast Healthcare Interoperability Resources (FHIR) [12] is considered as the appropriate choice, as FHIR adopts the lightweight REST architecture and represents all the exchangeable entities as resources sharing a common set of characteristics [13]. In the WSB platform, due to the use of well-known HTTP methods, the collected healthcare data are easily propagated as FHIR messages through the device gateways to the platform's server-side, where they are parsed and forwarded to the services of the platform for storage and processing.

The use of FHIR is enhanced by its combination with a universal code system representing the observed and measured data types and their values. In order to ease the semantic interoperability of the proposed platform and allow for its integration with devices and other external platforms, we consider the use of the terminology database of LOINC (Logical Observation Identifiers Names and Codes) [14] for representing healthcare and wellbeing data in a standardized way. LOINC provides a common terminology made up of identifiers, names and codes for laboratory, clinical and physical observations, including clinical measures like vital signs, collections of laboratory test results, set of answers on a survey or form regarding, for example, physical activities of a person, etc. The received data types and values are transformed into their respective LOINC representation and embedded into FHIR messages by the middleware layer of the WSB platform.

Semantic differences between dissimilar types of healthcare and wellbeing data should be resolved and, thus, semantic technologies are applied. To be more specific, a well-specified semantic knowledge model implemented in the form of an ontology (i.e., a vocabulary of concepts of the health domain and their relations enriched with logical axioms), called HLD Ontology, is used to annotate semantically the raw data collected by sensors and devices, transforming them into meaningful data based on the schema of the aforementioned ontology. The semantic knowledge model constitutes the basis upon which logical inference and advanced reasoning techniques regarding the process and analysis of lifelogging and health-related data by the underlying mechanisms of the system occur. Moreover, in order to achieve expressiveness, the WSB Policy Model Ontology is also used for the specification of access control rules, based on the expressive stack of Web ontology languages that are used for the formal representation of rules,

inference and decision making over access operations. Both the abovementioned semantic models are thoroughly described further on.

2.3 Device Interoperability

The IEEE Standard Computer Dictionary [15] defines the term *interoperability* as “the ability of two or more systems or components to exchange information and to use the information that has been exchanged”. In this respect, the use of specific codes, terminology and formats in eHealth environments, limiting at the same time optionality, favours interoperability. IEEE 11073 Personal Health Device (PHD) [16] is a family of standards that specifies the communication between personal health devices. It defines the concepts of *agents* and *managers*. The agents correspond to personal health devices. Their main features are that they are inexpensive, battery-operated sensor devices with low power consumption and that the user interface that they offer is rather limited when it comes to its functionality. The managers exhibit greater computing capabilities compared to agents, representing small computers or smart phones. Their role is to receive the agents’ data, reacting on update events and triggering events on the agents. Usually managers only monitor and display the agents’ data, but in some cases they might control the agents’ actions as well. For a more detailed analysis on agents’ data, managers can transmit it to a remote supporting center. In most cases, the communication channel among private healthcare devices and managers is held by a point-to-point connection.

Table 1 introduces the active 11073 PHD standards for device communication [17]. For example, a normative definition of communication between personal

Table 1 The IEEE 11073 PHD family of standards

IEEE 11073 PHD standard	Device specialization
11073-10404:2010	Pulse oximeter
11073-10406:2012	Basic electrocardiograph
11073-10407:2010	Blood pressure monitor
11073-10408:2010	Thermometer
11073-10415:2010	Weighing scale
11073-10417:2014	Glucose meter
11073-10418:2014	International Normalized Ratio (INR) monitor
11073-10420:2012	Body composition analyzer
11073-10421:2012	Peak expiratory flow monitor
11073-10441:2015	Cardiovascular fitness and activity monitor
11073-10442:2015	Strength fitness equipment
11073-10471:2010	Independent living activity hub
11073-10472:2012	Medication monitor

tele-health glucose meter devices and computing devices (e.g., mobile phones, personal computers, personal health appliances, and set top boxes) is established by the IEEE 11073-10417:2014 standard enabling plug-and play interoperability.

3 An IoE-Enabled Intelligent Wellbeing Service Bus

In this section, an intelligent service-based platform, the *Wellbeing Service Bus*, is presented. The goal of this platform is to connect the plethora of health- and wellbeing-related devices, web and mobile applications and decision support systems into a single, powerful platform that enables processing of data in an integrated manner, which, in turn, makes joint decision making possible. The sophisticated infrastructure along with its intelligent services is presented in the following sections.

3.1 Architecture

The architecture of the intelligent platform is conceptually divided in three layers: (a) the data layer, (b) the middleware layer, and (c) the smart applications layer. Figure 2 depicts the way the services of the architecture are organized by means of

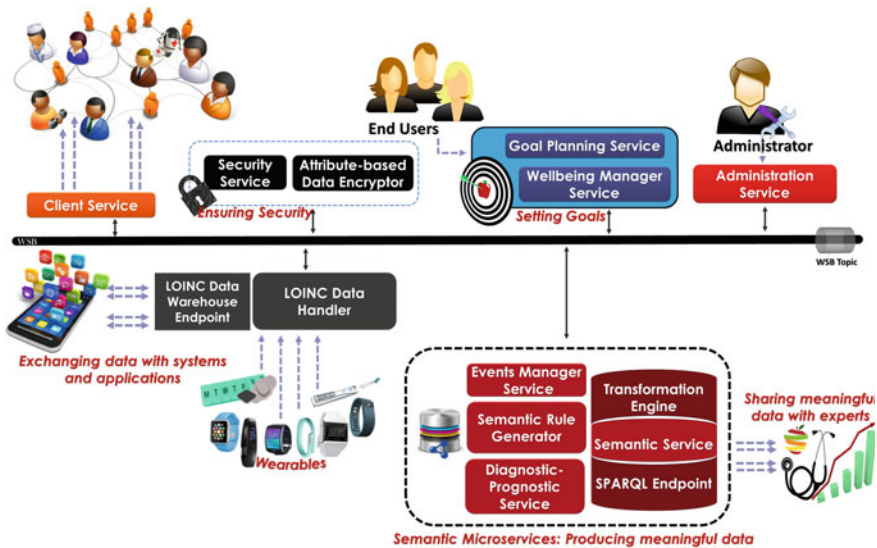


Fig. 2 The Wellbeing Service Bus

an Enterprise Service Bus. The following sections describe the role of the layers along with the functionality of the services of each layer.

Data and Data Sources

The *LOINC Data Handler* acts as an entry point of LOINC-compliant data into the WSB platform. It offers the necessary REST APIs to external data sources (devices, platforms, etc.) in order to post user monitoring data into the system. The REST interface expects JSON or XML formatted messages that contain LOINC encoded information in their body. After parsing and validation stages are completed, the LOINC Data Handler performs the following operations:

- it forwards the LOINC encoded data to the Transformation Engine that in turn transforms them into semantic data, and
- it stores the LOINC data into the LOINC Data Warehouse, which acts as a repository for the LOINC Data Warehouse Endpoint.

The LOINC Data Warehouse Endpoint provides access to third-party services and applications that would like to retrieve health and lifestyle data in its initial, raw form. Such kind of data are usually stored in the LOINC Data Warehouse, and are exchanged with the aforementioned external entities via messages leveraging the FHIR standard and LOINC encoding. The LOINC Data Warehouse stores the user monitoring data in their initial form, as they enter the WSB Platform. Specifically, LOINC Data are placed in the repository directly by the LOINC Data Handler service. For performance reasons the warehouse implements a NoSQL database to cope with read/write times of large amounts of data. In addition, data can be stored as is, in the form of either key-value stores or semi-structured documents. Most NoSQL implementations also offer REST APIs that can accept or return data and queries in XML or JSON format. For data redundancy and when combined with the features of a private Cloud testbed, the LOINC Data Warehouse can be scaled across multiple servers ensuring that there will be no single point of failure.

Data used by the WSB Platform can be acquired by multiple and heterogeneous sources. These data sources can either be smart personal devices that incorporate various communication standards or even integrated and independent health platforms. With respect to WSB, data can originate from anything that can be regarded as a data generator and is able to connect through the public WSB gateways.

The WSB Platform can connect to other integrated health related platforms, through the Data Endpoints (LOINC Data Handler, LOINC Data Warehouse Endpoint). Such platforms can either be a source or a consumer of user health and wellbeing data as long as they are able to use the REST APIs offered by the WSB Platform. In cases where users happen to be monitored by other health related platforms, the WSB Platform could analyse that data—provided that the user authorizes such an action—to perform a more thorough analysis and provide a more personalized profile according to certain situations (medical or psychological

conditions). In that same fashion, wellbeing data collected by the WSB platform could be of value to other client platforms and thus allow them to perform a better user-related evaluation on a broader range of monitored data.

Another source of wellbeing data can be various medical and wellbeing client applications. Such applications are usually targeted on gathering data from multiple devices that monitor the user's activity or on creating a personalized profile of a person based on their habits. Such applications—usually mobile or web based—do not necessarily perform any second level analysis of the collected data and are limited only to providing a convenient way for users to track certain aspects of their daily life. Client applications can be used in the same way by the WSB platform in order to pull data and offer a next level service for users. Additionally, the WSB platform could achieve a better integration with such applications so that they can in turn offer a more personalized type of service to the end users (i.e., allowing users to continue using their existing applications and at the same time receive a more sophisticated service).

The most common type of data originates from personal devices that are used on a daily basis by users. They range from weight scales (IEEE 11073-10415) [18] to cardiovascular activity monitors (IEEE 11073-10441:2013) [19] and smart insoles that track a person's walking activity. Most of these devices are able to use the IEEE 802.11 (Wi-Fi) communication protocol and thus post their monitoring data to various endpoints. The WSB platform is able to collect such data directly from the aforementioned devices through a user gateway or through an aggregation service implemented by the device manufacturer.

Middleware

The transformation of the LOINC data to their ontology-based format, the ontological data storage and the application of reasoning techniques is realized by the *Semantic Microservices* of the middleware layer. With respect to data transformation, the Health and Lifelogging (HLD) Ontology [20] is used as the target schema. The HLD Ontology models concepts relevant to the data that can be gathered from various sources (devices, platforms or applications) for a user. In particular, the HLD Ontology helps capturing demographic data, such as the Age and Gender of a user, as well as data relevant to their Body Measurements and their Role(s). In addition, any Medical Conditions along with their Symptoms are linked to the user (Person), as well as any Goals they may have set for health or wellbeing reasons. For each user, we keep track of Observations, i.e., information generated by a smart data source and sent to the LOINC Data Handler. Such observations are always accompanied by an Observation Type, which describes their semantic type and may include a value, a unit and a timestamp. Finally, the HLD Ontology allows linking the capturing device as well as any alert that may have been triggered to the corresponding observation. The concepts defined in the HLD Ontology and the relations between these concepts are depicted in Fig. 3.

With respect to the overall data processing workflow, the initial stage involves the transformation of the LOINC data to their semantic format, which is realized through the *Transformation Engine*. In this respect, the LOINC Data Handler

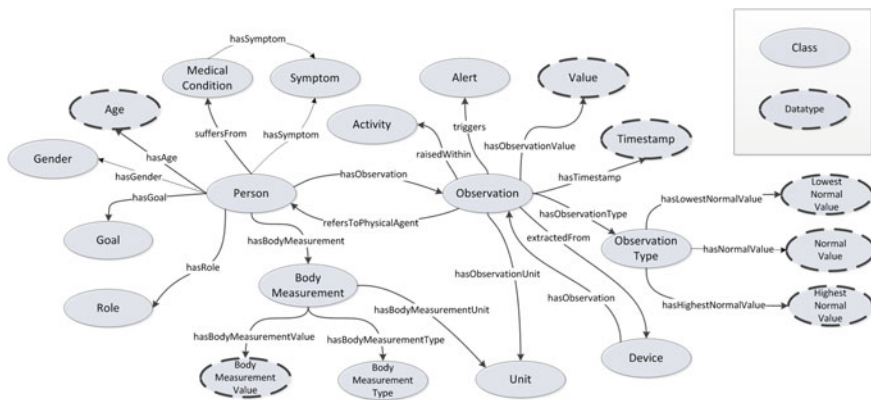


Fig. 3 The Health and Lifelogging Data (HLD) Ontology

pushes the newly-fetched data to the Transformation Engine, which is responsible for generating the semantic data. The Transformation Engine leverages predefined static mappings between the LOINC codes (source schema) and the Health and Lifelogging Data (HLD) Ontology (target schema). The underlying API of the Engine receives the LOINC data as input and outputs the respective RDF data, which are then sent to the Semantic Service. At the same time, the Transformation Service forwards the transformed data to the *Events Manager Service*, which wraps them in an event message and sends them to the appropriate WSB Topic (Publish/Subscribe implementation of the platform). From that point on, any service of the WSB that is subscribed to the aforementioned WSB Topic can be notified of the arrival of the new data.

Once the Semantic Service has received the new information, it has to store it in the *Quad Store*. The Quad Store, also known as a *Named Graph* [21], is a graph-based data structure, designed to hold RDF data. In such a database, statements (triples) are organized in a subgraph, to which a name in the form of a URI is assigned. In the case of the Wellbeing Service Bus, statements are organized in patient-oriented subgraphs, this way allowing the processing and querying of the data to be carried out more efficiently. Then, queries to the Quad Store can be applied in two ways: either via a predefined API, which is provided by the Semantic Service or in the form of SPARQL queries through the *SPARQL Endpoint*. In the former case, the API offers methods for a set of queries that are expected to be applied frequently by the WSB services (e.g., query for the allergies or medical conditions for a given user). Finally, in order to enhance the overall performance of the Quad Store with respect to responding to queries, sophisticated semantic caching mechanisms are applied.

The final stage of data processing involves the application of reasoning techniques to the semantic data with the intention of inferring health-related information that is not explicitly stated. The process of reasoning takes place in the Semantic Service by leveraging ontological rules, stored in the *Rules Repository*, a special

database for semantic rules. The aforementioned rules are partially formed following recommendations of health-related organizations, such as the American Heart Association [22]. Such rules can be used to deduce if the blood pressure of a user is low or high, if the amount of food and vegetables consumed within the day conforms to the ideal daily quantity based on the experts’ recommendations, if the amount of time spent on an activity qualifies them as “active”, etc. Examples of rules formed based on experts’ recommendations are the following:

- The normal blood pressure values for an adult over 20 years old are 120/80 mm Hg (less than 120 systolic AND less than 80 diastolic) [23].
- A user should take up at least 30 min of moderate-intensity aerobic activity at least 5 days per week for a total of at least 150 min of activity [24].
- A user should be involved in an activity for more than 10 min in order for this activity to be considered as exercise to boost their well-being [25].

The rules that are already stored in the Rules Repository are helpful for tackling the cold-start problem, but they present a lack of personalization. In this respect, apart from formulating semantic rules based on experts’ recommendations, a set of personalized rules can be generated by applying an internal logic to the acquired data. Specifically, the *Semantic Rule Generator* is responsible for creating personalized semantic rules. In this respect, the Semantic Rule Generator observes a specific data attribute and extracts a normal range of values for a specific user, i.e., it classifies the user by taking advantage of classification machine learning techniques (e.g., Support Vector Machine). The duration of the training period of these classification methods is predefined according to the monitored attribute and user. The output is translated into semantic rules, which in turn can act as triggers of events whenever values of the monitored attribute deviate from the normality range. As an example, we could regard the case of sleep duration and sleeping hours. Whenever a user begins to monitor their sleeping habits, the Semantic Rule Generator gathers all data regarding sleep duration and sleep/wake hours. Over the course of the training period, it will be able to determine how long this specific user normally rests and also which their preferred sleeping hours are. Then, the output will be translated into semantic rules. Figure 4 shows a personalized semantic rule concerning the sleeping hours of a user with ID “user6998447210”. The semantic rules extracted by the Semantic Rule Generator are then used by the Semantic Service and by the Diagnostic-Prognostic Service or the Wellbeing Manager Service at a later stage.

Finally, the reasoned data undergo a final stage of processing; the one that goes through the Diagnostic-Prognostic Service, which transforms it to *wellbeing* data. In particular, the Diagnostic-Prognostic Service’s objective is to reason under

Fig. 4 A personalized rule concerning the sleeping hours of a user with ID “user6998447210”

```

Person (user6998447210) ^ hasSlept(?p, ?d)
^ Duration (?d) ^ lessThan (?d, 6) ->
SleepDeprivedPerson (user6998447210)
    
```

uncertainty, in an attempt to predict the health condition of a user through observations. While the aforementioned semantic rules exploit deductive reasoning and the process of inference is deterministic, the Diagnostic-Prognostic Service is a nondeterministic service, which estimates the conditional probability that a user suffers from a disease and alerts them, if so. The Diagnostic-Prognostic Service contains two main entities: (a) the *Bayesian Network*, and (b) the *Agents' Ecosystem*.

1. The *Bayesian Network* (BN) denotes the dependencies between the observed variables and the target variables under uncertainty. The BN addresses all the relationships between the nodes (symptoms, diseases, activities and observations) of the Probabilistic Graphical Model. Exploiting the semantics of the HLD Ontology and the reasoned data, the BN updates the conditional probability tables (CPT) over the nodes. For example, given the fact that a person who suffers from high blood pressure gains weight or does not exercise as much as they should, the conditional probability of having a heart attack increases. It should be noticed that each user is associated with their own BN.
2. The *Agents' Ecosystem* (AE) consists of unique personalized agents, named UserAgents, which constitute the cyber-physical representation of each user. After acquiring the required data from the Semantic Service through a WSB message, the UserAgent gains evidence about reasoned health data (e.g., tachycardia) relevant to the corresponding user. Subsequently, the Conditional Probability Distribution (CPD) of the user's BN is updated. Having hard evidence on a group of variables, information about a disease can be extracted.

The result of the operations of the Diagnostic-Prognostic Service on the semantic data is their transformation to wellbeing data; health and wellbeing information for each user is enriched through the application of semantic rules and through intelligent processing by the DPS. The wellbeing data are updated by the UserAgents whenever a change in the state of a variable occurs. Figure 5 illustrates a simple example of the way the user's BN is updated according to observations on a given set of variables. Specifically, the example involves a Bayesian Network, which contains an Activity node (Activity1), two Disease nodes (Disease1 and Disease2) and two Symptom nodes (Symptom1 and Symptom2). As depicted in Fig. 5, the conditional probability $P(D2 = T | A1 = T)$ of Disease2 being true is increased when Activity1 is also true. This implies that Activity1 is a rather harmful action or habit. Consequently, the user or their personal doctor updates the state of the symptoms and the activity of the example, as shown in the right part of Fig. 5. This update action (i.e., event) awakens the UserAgent so that it can solve the BN's conditional probability tables. After obtaining hard evidence on these variables, the user's BN is updated. Finally, with the probability of the patient suffering from Disease2 being higher than 0.5, the system is responsible for notifying the user and their doctor.

The flow of data, from the moment they enter the system through the LOINC Data Handler until they take their final form, is depicted in Fig. 6.

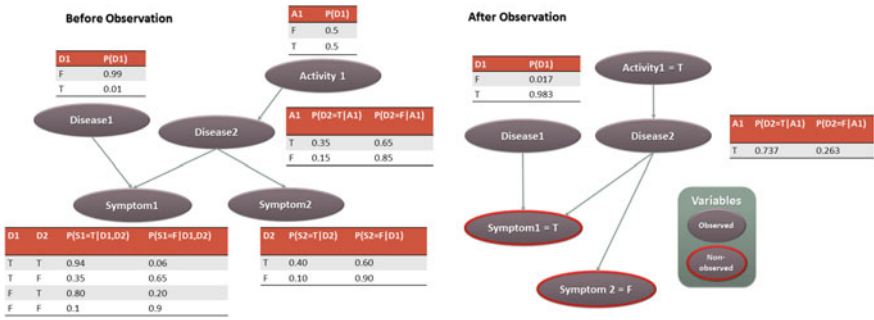


Fig. 5 The Bayesian Network of a user before observation (left) and after belief updating (right)

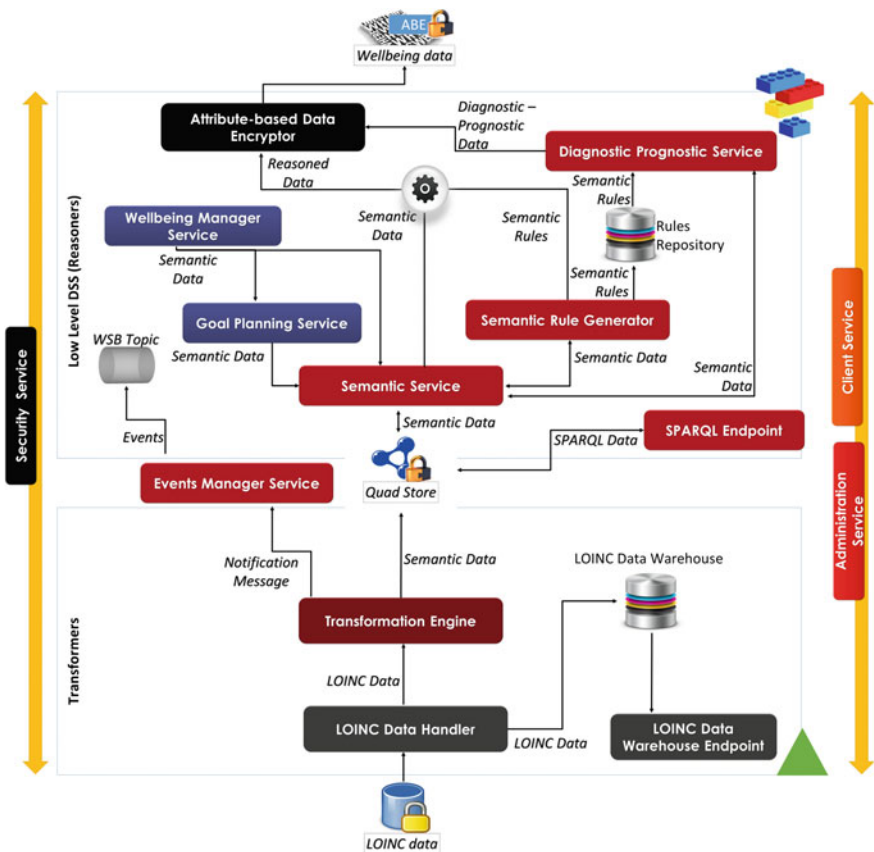


Fig. 6 The flow of data in the Wellbeing Service Bus

Smart Applications

The main users of the wellbeing data that are generated by the WSB platform are the Client Service and the Goal Setting Services that is, the Goal Planning Service and Wellbeing Manager Service. Each of the aforementioned WSB services utilizes the data for specific processing purposes, as described below.

The *Client Service* acts as the connection point between the Wellbeing Service Bus and any other service, application or platform that is not implemented following the JBossESB guidelines. It provides data relevant to progress updates, notifications (alerts) and combined knowledge, as extracted from the analysis of the collected data. The main user of the Client Service is the Social Health Avatar Network, a social network that aims at projecting the physical status, the living conditions and lifestyle habits of data owners through an abstraction, named “Social Avatar”. The Social Health Avatar network is described in detail in Sect. 5.

The goal-enabling duet of services, the Goal Planning Service and the Wellbeing Manager Service, has been designed to sketch abstract plans and user-specific schedules, respectively, towards the achievement of goals set by the user through sophisticated user interfaces offered by the WSB. On the one hand, the *Goal Planning Service* is based on the available data for the user that has set a goal in order to produce an abstract sketch of the actions that need to be performed to achieve the previously-mentioned goal. Besides the type of the goal itself, the Goal Planning Service takes into consideration, among others, the user’s living conditions, their physical status and their medical history. Then, it suggests general ways of achieving the defined goals, respecting the available resources (i.e., desirable budget in case of medical or dietary supplies) and time limits, if any. For instance, as described later in this chapter, in a weight loss use case, the role of the planner is to define the amount of calories a user should take in daily and the duration of daily exercise a user should engage in so that the goal is achieved in a predefined time period.

On the other hand, the *Wellbeing Manager Service* is responsible for generating user-specific guidelines towards the achievement of a goal. It leverages the wellbeing data maintained in the Quad Store, as well as the abstract plans produced by the Goal Planning Service. In particular, the Wellbeing Manager Service transforms the abstract suggestions of the Goal Planning Service to concrete guidelines, such as a specific medical treatment the user should follow in case of an illness, a daily diet program they should pursue or the exercise they should engage in to strengthen given parts of their bodies. At all times, the service takes into account any restrictions that may have been set.

The aforementioned goal-enabling services are implemented as intelligent agent-based systems. Each user that decides to set a goal via the WSB platform is assigned an instance of the Goal Planning Service and multiple instances of the Wellbeing Manager Service, based on the number of defined goals. In this respect, all the instances of the Wellbeing Manager Service are following the abstract plan generated by the GPS and, if need be, cooperate with each other towards the joint achievement of the user’s targets. In both cases, the user’s goals are treated as

Constraint Satisfaction Problems (CSPs), i.e., as a set of objects whose state must satisfy a number of limitations.

3.2 *Securing the System*

As the whole WSB platform incorporates different devices and appliances that use a variety of technologies, the retrieved sensitive data may become vulnerable to a variety of either internal or external threats and attacks due to the combination and interaction of the aforementioned entities. Thus, for ensuring the integrity, availability and confidentiality of the data collected, transmitted and processed, the latest security practices are considered. An overall security system architecture designed in a way that fulfils the special security requirements of the infrastructure adopts security standards for secure communication among the system components, data protection and access control.

The Security Service of the platform is a service mainly responsible for user authentication and authorisation, managing, processing and evaluating any request for access to a resource based on explicit policies formalised in the Web ontological format, obtaining and enforcing authority decision. To be more specific, any software or hardware component's request to act in the WSB platform, driven either by its own logic or by a human directly or indirectly, is intercepted by the Security Service. The latter evaluates the request against a matching policy retrieved from the policies' repository and enforces the decision made, i.e., either allows or denies the access to the desired service or data.

The access control mechanism deployed is scalable, attribute-based as well as privacy- and context-aware, so as to ensure data authenticity and system trustworthiness. It is based on the Attribute-Based Access Control (ABAC) model [26], taking into account the high degree of data sensitivity involved in transactions between the services of the WSB platform. As mentioned above, for the control of access requests, policies are defined, which determine how the access is given to data and services as well as how end users' personal data is used based on the underlying purpose. Personal data should be kept secure from any potential threat, and, as a result, their protection is a major issue for us.

The administrator of the system is able to define rules that secure the data from being accessed by malicious users. This is accomplished by taking into account attributes of the involved entities, contextual parameters (like time or space), events that may occur or actions required to be done as a prerequisite or in consequence of the authorisation (e.g., log activity). A user-friendly Graphical User Interface exposes all functionalities provided by the Security Service to the System Administrator in order to define access control rules and control, based on the system requirements, the access to the resources. The underlying mechanism responsible for access control in the WSB platform combines the logic of RDF triples (subject-predicate-object) and takes advantage of the deductive reasoning. Going a step further, the proposed modelling approach interworks with a

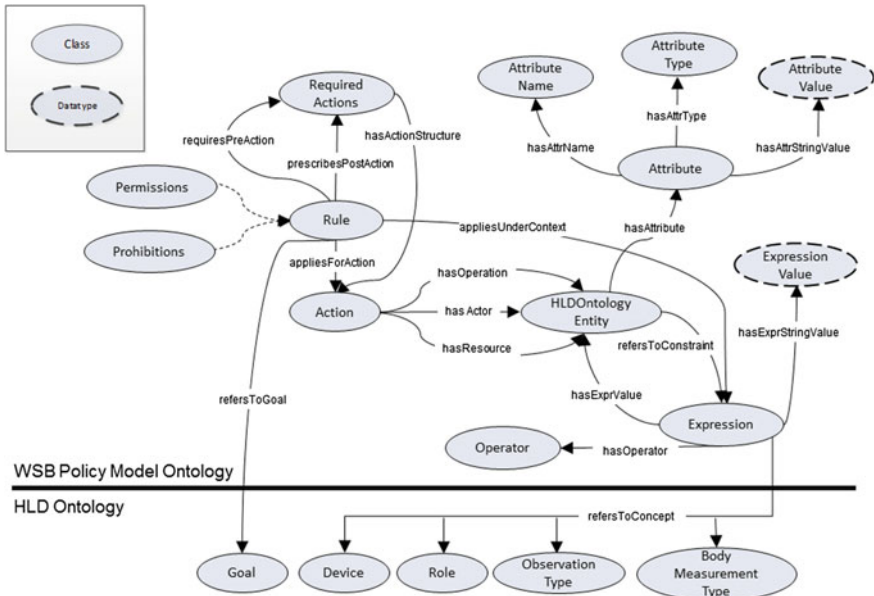


Fig. 7 The WSB Policy Model Ontology

state-of-the-art privacy-aware access control model [27] supporting semantic definition of the privacy principles and policies.

In order to achieve expressiveness, the Policy Model Ontology is used for the specification of access control rules. As shown in Fig. 7, a rule may be either a Permission or a Prohibition. Each rule applies to an Action, where an Actor (access requestor) is allowed or denied to perform an Operation (e.g., view, edit, execute, etc.) on a Resource (e.g., tools, services, data, etc.). The object properties hasActor, hasOperation and hasResource associate each rule with the aforementioned involved rule entities represented by the HLDOntologyEntity class. Based on the attributes of the aforementioned entities and taking into account contextual parameters (if needed), the system administrator may define this action.

The Expression class models either a concept of the HLD Ontology (as depicted in Fig. 7, this concept refers to an instance of one of the following classes of the HLD ontology: Device, Role, ObservationType or BodyMeasurementType) or a logical relation between the aforementioned concept and a value, modelled as the datatype property hasExprStringValue, by using an operator which is defined as an individual of the Operator class (i.e., greaterThan). The Expression class is associated with the HLDOntologyEntity class via the refersToConstraint and hasExprValue object properties. Furthermore, the underlying purpose of an action is reflected through the refersToGoal object property while contextual parameters, such as events, under which the Actor is allowed to perform an Action, are defined as logical relations via the Expression class.

The individuals of the `HLDOntologyEntity` class are characterised by `Attributes` via the `hasAttribute` object property. The types and names of the access requestors' (Actors) and Resources' attributes are modelled through the `AttributeName` and `AttributeType` classes and their possible values are specified through the `hasAttrValue` datatype property. A type of an Actor attribute could be any feature typical of a user, such as their role. Resource attributes may refer, for example, to the type of the resource that the actor wants to access (such as data or a service) or the owning person. In addition, actions that should have preceded or follow the enforcement of a rule are represented by the `RequiredAction` class, which is linked to a rule through the `requiresPreAction` and `prescribesPostAction` object properties, respectively, and have the same structure with the actions. A `postAction` could be, for example, the logging of the access by the system. With the adoption of such a flexible access control model, which is based on user and resource attributes, the need of explicit authorizations assigned directly to individuals is avoided and real-world needs of an IoE environment for multi-factor decision support are satisfied.

Furthermore, to protect data collected, processed and shared between the entities involved in the WSB platform, Attribute-Based Encryption (ABE) [28] is used, which extends the Identity-Based Encryption (IBE) and the user's private key and the cipher depend on various attributes, assuming the identity of the user as an attribute. This way, data, involved in these transactions, incorporate access policies, thus reducing the need for trustworthy storage systems and complex schemes for maintaining access policies across the WSB services, enabling also information confidentiality. The Attribute-based Data Encryptor of the platform is responsible for encrypting the end users' semantic personal data, including diagnostic-prognostic data and reasoned data, in a way that is efficient to process transmission of such data, having as primary objective the protection of privacy. As far as the decryption of the data is concerned, the ciphertext depends only on the attributes of the user.

Concerning the messages exchanged between the different services and components of the system, those are encrypted in order to ensure trust and confidentiality of the information transmitted among them. All communicating components have to use correlation IDs that are generated by the request initiator and then travel along with all consequent requests to other services and data sources, as all access attempts are logged for the detection of any possible threats from and misbehavior of the platform components. Finally, as far as the physical security of the whole infrastructure is concerned, it is taken for granted that it is ensured and protection from destruction by physical phenomena or unauthorized human access is provided by the data center, where the infrastructure is deployed.

3.3 Cloud-Based Deployment

The WSB Platform as an architecture of interconnected and independent services needs to assure that at any given moment the Quality of Service (QoS) is

exceptional mostly because of the constant incoming streams of data. Cloud solutions and specifically the IaaS (Infrastructure as a Service) resource delivery model has become a necessity among health related platforms as it offers the required features out-of-the-box. Currently, two control models are offered as an IaaS; private and public cloud infrastructures. In the WSB Platform, the private cloud approach is mainly preferred for control, security and privacy reasons. That means that from the owning organizations' perspective, the allocated resources (e.g. Virtual Machines) must not be exposed at public networks and must be protected behind a corporate firewall. This requirement complies with the need for data privacy and security among the various interconnected services of the WSB Platform. In the proposed architecture the only thing that could be exposed to the public internet are the service endpoints for user incoming data and third-party applications or platforms. Below we describe the most common features of IaaS stacks and how the WSB Platform can profit from them. As a reference technology stack the WSB Platform uses the Openstack [29] suite, as a pioneer in the domain of private cloud solutions.

Load Balancing and Elasticity

The elastic nature of clouds allows the continuous monitoring of services and VMs and is scalable with respect to the offered resources according to the currently monitored workload and the user SLA (Service Level Agreement). Elasticity is a more abstract term and a characteristic of cloud infrastructures that tries to tackle situations where certain services experience a sudden increase to their workload. The term *Load Balancing* as an elasticity implementation method refers to a cluster of VMs that are allocated to a specific service. One node of the cluster usually acts as a request forwarder to other nodes that perform the actual service logic. Load balancing only applies to stateless services that usually perform basic data-oriented operations. Since the WSB Platform is service-oriented, load balancing can be applied to most of its services, whether they implement REST APIs (HTTP) or maintain message queues to communicate with the main service bus (JMS). Load balancing can ensure that requests can be evenly forwarded to nodes in order to achieve both high service availability and acceptable request processing time. The elastic nature of the cloud paradigm allows for dynamic addition or removal of nodes to a cluster according to the current workload that is monitored at certain periods. For instance, it is expected that most user activity data will be generated during morning hours and, thus, the WSB Platform is more likely to experience higher workloads at these hours.

Apart from the Load Balancing mechanisms that are offered out of the box by most private Cloud vendors, elasticity needs to be applied in cases that the workload cannot be distributed in more than one node. This becomes apparent when we consider the case of the service bus, which is the central communication channel of the WSB Platform. In order to cope with an increased workload, the VM that holds the service bus must be able to scale according to its needs. Fortunately, lately this can be achieved as modern cloud IaaS technologies support live resizing so that a

certain VM acquires or releases computational resources with minimal to no downtime.

Infrastructure Monitoring

As mentioned above, in order for the infrastructure to be able to offer effective elastic and load balancing solutions, it needs to monitor the services and take precautionary actions to prevent service downtimes and data loss. Monitoring mechanisms are nowadays offered as a part of the underlying IaaS and can be customized to match the user's needs. Monitoring can be performed in an active fashion by measuring the number of requests that arrive for a specific service or the amount of data that is exchanged between services. A sudden increase in requests or in the amount of exchanged data means that the infrastructure should take action and ensure that the service will be able to cope with the increased workload. However, monitoring can be also performed passively by monitoring the VMs' utilization of resources. Based on analysis of historic data, the monitoring service can take appropriate action when it detects certain peaks in the utilization of CPU or RAM, something that indicates that the workload for a VM has increased.

Networking as a Service

The term *Networking as a Service* includes the provisioning of virtualized network resources to the user. This includes the creation of virtual appliances (routers, firewalls) as well as services (NAT, VLAN). Virtualized and user specific networking can be essential for the deployment of the core blocks of the WSB. Virtual networks can add another layer of security as they can isolate even more sensitive VMs from the rest of the network as well as the outside world and, thus, prevent side-channel attacks. Two-phase NAT is widely used in private cloud deployments, where VMs are secured behind a virtual router instead of sharing the same private network with the physical infrastructure.

Persistent Storage and Data High Availability

Data is the most sensitive aspect in the WSB platform and it must be available at all times regardless of the workload that the platform or the underlying infrastructure might be experiencing. To ensure that data are always available, every system must eliminate potential single points of failure. Single points of failure can occur in the case of hardware failures (hard disk failures is the most common case) or VM errors (downtime, migration, hypervisor failures). Modern cloud solutions tackle such issues by employing services that offer persistent storage features. Persistent storage blocks can be attached to VMs as an external storage medium and data written to them stay there even after the destruction of the VM and the release of its resources. One of the most sophisticated mechanisms Cloud infrastructures employ is the efficient replication of these data blocks into multiple and -if possible- geographically scattered servers.

Image snapshots are another way of ensuring that the offered service and data are available at all times. Redundancy techniques are very common when platforms perform sensitive data operations. As in the case of the WSB platform, data handling and transformation services need to be constantly available. VM snapshots that are stored and deployed can ensure that, in the unfortunate case of VM or hardware failure, a backup VM that is identical can replace it and pick up the work almost immediately.

Modern Cloud technologies also offer Databases as a Service (DaaS). These feature the allocation of database management systems to users through the use of APIs. Apart from the convenience and the efficiency offered, these services also ensure that data are always available by creating clusters for distributed management. Most NoSQL database systems already incorporate distributed architectures in order to improve performance and ensure that data are protected and available in case of software or hardware failures.

4 Interfacing the Users: The Social Health Avatar Approach

Users participate in the platform through the use of a privacy protecting anonymous avatar, and a health and wellness monitoring framework, where information from personal, wearable devices is transparently fused into the platform, linked to the owners' health profile, and visualized and accessed by the end-users through a special dashboard. The data collected and inferred for each user form their *health avatar*, which is used as the electronic equivalent of a human and features a dynamic life profile corresponding to the human owner's physical status, living conditions, and habits.

The Wellbeing Service Bus platform brings the health avatars of the subscribed users together with the view of forming a network of people; simple users and medical experts or other professionals of health and wellbeing-related domains. The avatars of these users will present their health and wellbeing information in the form of tweet-like posts, which could be visible to their family and their doctors, always based on the permissions that have been set by themselves (the data owner). This way the health avatar of a data owner is "upgraded" to a *social health avatar* [20, 30].

The advantages offered by such a social network are multiple and multisided. To name a few, users (either suffering from a particular illness or just being concerned about their fitness) can communicate with their doctors or dietitians virtually and instantly. This way, unnecessary visits to the professional's site of practice are avoided, while home recovery and tele-health are embraced. At the same time, users of the Social Health Avatar Network can connect with each other and exchange their experience of a certain condition or disorder they suffer from, as in group therapy.

4.1 *Categorising Users*

A user, and consequently their social health avatar, may belong to one of the following categories. The *Health and Lifelogging Data Owners* (HLD Owners) are the ones that generate the data that are posted to their health avatar profile. These pieces of information are collected through personal wearable devices, such as activity-tracking armbands or wristbands, health-related mobile applications, such as Apple's Heart application, and public devices/appliances, such as smart weight scales. Data Owners have the ability to define which devices and applications will be used as sources of data through their profile. This information is stored in the platform, so that the appropriate interfaces are defined between the LOINC Data Handler and the devices. Then, any new pieces of health and lifestyle information derived from the selected devices are acquired in a push fashion through the Client Service of the Wellbeing Service Bus and posted to the user's profile.

The *Health and Lifelogging Data Users* (HLD Users) represent the group of medical professionals, such as doctors, nurses and pharmacists, dietitians and fitness coaches. HLD users can monitor the status updates of an HLD owner either because they are linked to the latter in real life (i.e., by being the owner's doctor or coach) or for reasons of statistics. In the first case, HLD users may send advice to an HLD owner in the form of notifications. At the same time, a specially-designed UI is offered so that medical professionals can create rules that will be later used during the semantic reasoning process that takes place within the WSB platform. The rules creation process is based on the IFTTT (IF This Then That) [31] methodology by leveraging the concepts and attributes defined in the underlying ontology. It has to be noted that HDL owners are responsible for providing access to specific HLD users or make their profile public to everyone.

Finally, a third, conceptual category of users is the one of software agents that crawl the data owners' profile with the intention of processing the produced data in order to provide automated prognosis or extract statistical data. While all other (human) users can hold an HLD owner and an HLD user profile at the same time (a doctor can produce data relevant to their own health and lifestyle), software agents are linked to a single HLD user profile.

Connections between avatars are not limited to the HLD owner—HLD user relationship. In fact, multiple HLD owners can interact with each other, exchange their opinions on a common health issue, provide support to each other, etc. Similarly, HLD users are presented with the ability to work together in the context of providing remote care to common patients in a comprehensive way. While the identity of HLD owners can remain a secret due to the sensitive medical information being shared, this is not necessary in the case of HLD users.

4.2 *User Interfaces*

Each HLD profile comes with a special user interface that enables the owner of the avatar to perform certain actions. These actions differ according to the role of the avatar in the social network. HLD owners are given the opportunity to set their own goals, such as weight loss or improvement of their physical condition. Through the dedicated user interface, they can select one of the predefined types of goals and adjust their parameters so that their preferences are covered.

After a user has set a goal, it is disseminated to the platform through the Client Service. The appropriate processes are triggered and a schedule relevant to the goal set by the user is produced. This schedule is made known to the user through the Goals UI. In this course, the user is informed of the schedule they must follow, the activities they must engage in and any other actions they have to take in order to succeed in their goal. While a goal is active, any relevant information produced by the user's wearable devices or provided manually by the user is taken into account by the platform's tools. This way, if the user deviates from their goal, the platform can produce a new schedule from scratch or apply some changes to the already existing one and present it to the user through the same UI.

The main functionality of the data users' profile is the ability to create rules for their patients/trainees. These rules are stored in the Rules Repository of the Wellbeing Service Bus and exploited by the Semantic Microservices in order to deduce valuable information or define the occasions that require a notification to be sent to a data owner.

Rules are shaped according to the IFTTT logic, as described above. Based on the IFTTT paradigm, a service (i.e., a function) is triggered when an action has been performed. In this course, if a data owner has completed a set of actions, thus fulfilling a goal, or if they have failed to perform an action that has been part of a goal, an appropriate notification is sent to them via their social profile.

Finally, as far as access to personal data by professionals (regardless of their profile) is concerned, because of the application of the ABE mechanism, the data owner is responsible for managing access to their data by defining data encryption policies, describing the requirements that any user should meet so as to be allowed to access the requested data. The same applies for information, such as diagnosis, produced by professionals about a specific user. For example, a user may select to allow only Internists to have access to their blood glucose value measurements, and particularly the one that they have identified as their personal doctor, while the latter can set an access policy for their diagnosis that allows only a group of treating doctors for the particular patient to have access to them and the corresponding prescriptions.

4.3 Social Network Functionality

Apart from the Goal and Rules UI of the data users and the data owners, respectively, each user of the Health Social Avatar Network is able to send a friend request to another user, make a post to their personal profile, create or join a group and perform a search based on certain attributes. Each of these functions is described below.

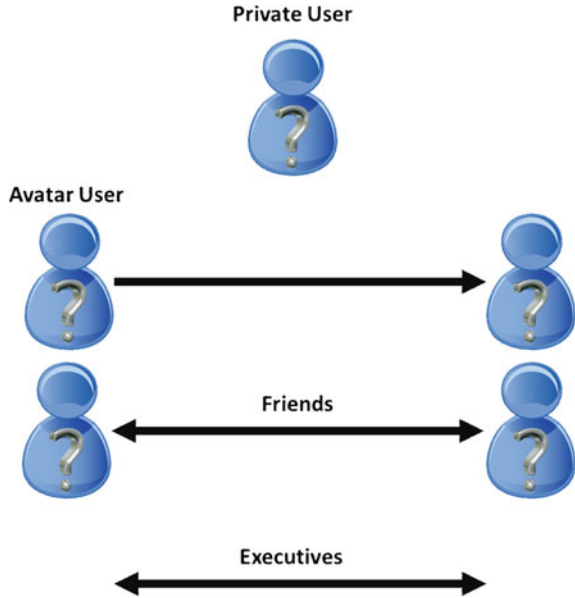
As in all major social networks, users are able to make *posts* either manually or through their wearable devices or mobile applications. The content of the posts may refer to the results of a new measurement, the mood of an avatar user, a medical article, etc. The visibility of such posts can be set individually (for each post) or globally (for the user's profile in whole).

Users can also participate in *groups* created either by their avatar or other users. This way, data owners are able to discuss with other data owners facing similar issues or having the same health- and lifestyle-related interests. They can also engage in group therapy activities, retaining their anonymity. Similarly, data users can join the same groups and provide advice to people facing medical issues. At the same time, they can be part of groups of professionals so that they can discuss, for example, about the health condition of a common patient.

The *search* functionality of the Social Health Avatar Network does not resemble the one of the well-known social networks. In fact, due to anonymity constraints imposed by data owners, search based on a person's name is not always effective. For this reason, the social network allows for performing search functions based on attributes, such as age, condition, goal or lifestyle interests. This way, users with common conditions or interests can connect to each other, while their identity is kept a secret or not, based on the choice they have made through their profile settings.

Friend requests can also be sent from one avatar user to another, this way establishing a friendship. There are four friendship levels, as shown in Fig. 8. A data owner or user that has no connections to other owners or users is considered a *private* avatar. Posts made by such a user are not visible to anyone else in the social network. At the same time, such users are not included in the search results of other users. An *avatar* relationship is established when a user follows another user. The follower is allowed to view posts of the user being followed if the latter has defined so in their settings, but their identities are kept anonymous. As in all social networks, the user being followed is not obliged to connect to their followers. Two users are considered to be *friends* if they are mutually connected through a friend request. A user may have access to the posts of their friends, always based on their settings and exchange private messages, with their identities remaining hidden by default. Lastly, connected users that wish to reveal their identities to each other are considered to be *executives*. Such a relationship is common in case of a data owner and the data users (e.g., doctors, coaches, etc.) that are linked to the former in real life.

Fig. 8 Friendship levels between Avatars



Overall, by adopting the successful model of online social network interaction through the *Social Avatar*, the presented platform provides a flexible communication medium that integrates and connects practitioners, patients, and virtual entities such as decision-support systems into a community for improving the quality of personal health.

5 The Wellbeing Service Bus in a Welfare Use Case

To demonstrate the Wellbeing Service Bus platform and its functionality, we have selected a welfare-related use case, linked to weight loss. In this use case, the user is able to define the parameters of the weight-loss goal, such as the amount of weight they desire to lose, the duration of the diet or of the daily or weekly exercise. At the same time, the user can state any culinary restrictions, due to their dietary habits or due to limitations caused by health issues (e.g., diabetes, allergies, lactose or gluten intolerance).

As discussed in Sect. 2, already existing solutions take the user's preferences into consideration and provide a well-defined dietary plan but most of them present certain limitations with respect to their functionality. For instance, some of them require manual data entry and do not provide the necessary interfaces for interacting with smart devices and others do not combine this dietary plan with exercise.

In our proposal, the welfare platform is using data collected through wearable devices, such as watches or armbands, smart scales, or mobile applications. Given a well-defined user target, the platform processes the collected data so that it can

generate a proper nutrition plan. In particular, the Goal Planning Service and the Wellbeing Manager Service undertake the definition of and collaborate within a user-set goal of losing an amount of weight. Given a set of historical data for a user, the Goal Planning Service creates a long-term, abstract dietary plan. The purpose of this plan is to define how many calories the user should take in daily and the amount of exercise that they should engage in, in order for the user to lose the desirable amount of weight during the predefined time period. The Goal Planning Service also defines the number of servings per food group (basic food groups; grain, protein, fruit, vegetables, dairy products and oil) a user should consume daily, without providing concrete meal suggestions. In turn, the Wellbeing Manager Service provides a specific dietary schedule, offering a number of alternatives of daily meals, taking into consideration the culinary preferences of the user and any food limitations, as discussed above. Specifically, the Wellbeing Manager Service produces daily meal schedules, taking into consideration the plan extracted by the Goal Planning Service and, at the same time, respecting the preferences of the user. While the Goal Planning Service identifies the necessary amount of each food group that should be consumed daily (e.g., 48 gr servings of whole grain¹), the Wellbeing Manager Service allocates the aforementioned amount to all or some of the meals of the day, providing specific products and recipes (e.g., 3 slices of whole-grain bread (see footnote 1)).

From the moment the goal has been set and the WMS has produced a dietary schedule, any new pieces of data arriving from the wearables to the service bus are considered to be events, which are handled by the Events Manager Service. The latter implements a special kind of internal logic, which is able to process the incoming events and infer if they are relevant with the goal (diet). If this is the case, they process the event data in combination with the goal-related data and update the latter accordingly. For example, if the event contains information regarding a meal that has been consumed by the user, the Events Manager Service updates the amount of the remaining quantities of each food group, accordingly. If the Events Manager Service comes to the conclusion that the diet has not been respected by the user (e.g., a meal has been skipped, excess quantity of a food group has been consumed or daily exercise has not been performed as planned), it can produce further events that will either be disseminated to the user in the form of warnings or will lead to the generation of a new diet schedule from scratch, through the Wellbeing Manager Service.

6 Evaluating the Service Bus

The Wellbeing Service Bus is currently under development, so its evaluation with respect to performance is not feasible at the time being. However, the WSB follows the architectural paradigm of the intelligent Enterprise Service Bus (iESB)

¹The recommendations are extracted from the non-profit consumer advocacy group Whole Grains Council (<http://wholegrainscouncil.org/>).

Table 2 The average execution times of the data access use case

	Average execution time (msec)			
	F1	F2	F3	F4
Triple store	163,38	134,78	140,58	186,20
MySQL database	160,52	150,40	140,64	138,50

developed within the European project ARUM² (Adaptive Production Management). The iESB [32] is an agent-based platform that relies on semantic technologies in order to plan, schedule and handle the manufacturing processes of highly-customized products, such as aircraft. Special focus is given to small-lot production and production ramp-up, which present a high frequency of disturbances (Table 2).

In the context of evaluating the semantic data managing services, namely the *Ontology Service* and the *Events Manager Service*, their performance, along with the performance of the underlying triple store, was compared to the use of a MySQL database maintaining part of the triple store data. In this respect, two use cases were considered; the first one concerned the evaluation of the Ontology Service through four representative functions, whereas the second one aimed at depicting the proficiency of the Events Manager Service through the generation of unexpected disruptions within a production process. At this point, it has to be noted that any details on the actual functionality of the above services are omitted, since they are considered to be out of context of this study. The aforementioned use cases are summarized as follows.

The first use case refers to the communication of the Ontology Service, the semantic data provider, with the Factory Network and Scenario Designer (FNSD), a tool of the iESB that is used to create models of the involved production processes. The use case simulates four consecutive requests for data, with each one requiring the application of a certain type of logic to the triple store data by the Ontology Service. Table 2 depicts the average execution times of this use case. In particular, the average execution times of the four functions of the Ontology Service (F1, F2, F3 and F4) are presented both with respect to the use of a triple store and a MySQL database. Similarly, Fig. 9 presents a graphical representation of the aforementioned execution times. The results show that, despite the fact that a triple store is a rather complex data structure compared to a relational database, it behaves better in two out of four functions. The fact that, at the time of the experiments, the triple store held 13 times more data than the MySQL database is quite important, as well.

The second use case describes the creation and publishing of production events. It refers to the communication of the Operational Scheduler, a tool that is responsible for designing detailed schedules of the production processes required to build a certain product, with the Events Manager Service. This use case shows the

²<http://www.arum-project.eu>.

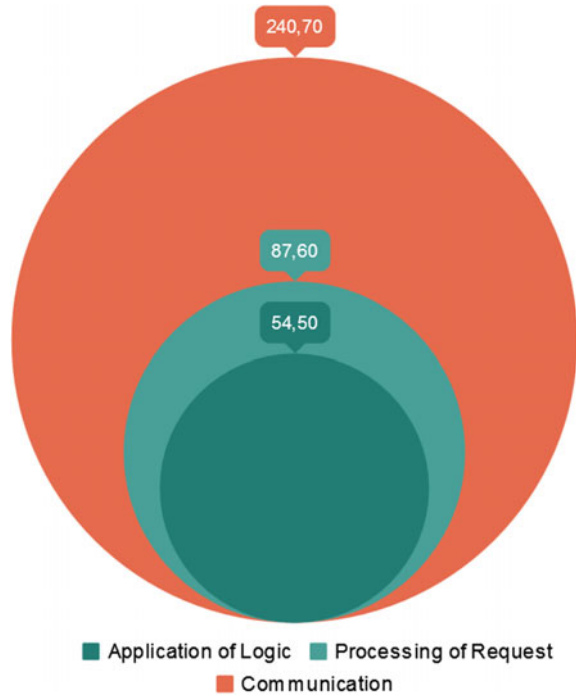


Fig. 9 Graphical representation of the average execution times of the four functions of the Ontology Service: comparison between the use of a MySQL database and a triple store

individual parts, to which the total execution time is divided within a single event creation and sharing. As shown in Fig. 10, it takes 382.8 ms in average to create and publish an event of a certain type. This particular use case involves the communication of the Operational Scheduler with the Events Manager Service in the scope of an event publishing request, the application of logic to the semantic data, the publication of the event and, finally, the response which is sent to the former by the latter. In this case, the logic applied by the Events Manager Service to the underlying data requires the generation of an additional event. In this respect, the communication of the two services took 240.80 ms, whereas the processing of the request and the application of logic required 87.60 and 54.50 ms, respectively.

The evaluation of the iESB was conducted through experiments ran on a personal computer with an Intel Core i5 2.80 GHz processor, 8 GB RAM, on a Microsoft Windows 7 Professional operating system and a Java Runtime Environment (JRE) v. 1.7. The iESB followed the JBossESB paradigm, which was deployed to a JBoss v. 6.1 Application Server. MySQL v. 14.14, distribution 5.7.9 was installed to the same computer. Each experiment was executed ten times and the average times were calculated for each case. During the aforementioned experiments, the triple store of the iESB maintained 94 MB of data, which corresponded to 137,198 RDF objects and 680,968 RDF statements. Similarly, 30,520 records were stored to the MySQL database and its size was equal to 6.3 MB, which made it almost 94 % smaller in comparison with the triple store.

Fig. 10 Graphical representation of the average execution times of creating and publishing a production event within the iESB



7 Conclusions

In this chapter, a health-related, service-based intelligent Wellbeing Service Bus is presented. The purpose of this service bus is to provide an ecosystem, where smart devices, applications and systems can integrate through a platform following the Enterprise Service Bus paradigm. The intelligent services of the platform exploit the power of ontologies and deductive reasoning in order to produce meaningful data that are not directly stated through existing real-life capabilities. The extracted data is processed by agent-based services that are capable of providing structured advice towards the achievement of user-defined goals and producing prognostic or diagnostic results with respect to the health status of a user. The dissemination of meaningful data is made easier through a health-related social network, which unites patients, medical professionals, athletes, and trainers. This social network, the Social Health Avatar Network, is paving the way towards the improvement of remote treatment and home recovery.

As regards the implementation of the collaborative platform described in this chapter, the individual components such as the Social Health Avatar [30], the Health and Lifelogging Ontology [1], and the Wellbeing Service Bus [1] have been implemented and tested individually. A full scale test of the integrated platform in order to extract conclusions on how to effectively link knowledge and expertise of

health professionals and make use of it in order to improve quality of life is the next step towards validation of the approach and ideas presented here.

Finally, our future plans include following IDSECOM [33, 34], an innovative ID-based approach for connecting IoT objects.

References

1. Meridou, D. T., Kapsalis, A., Kasnesis, P., Patrikakis, C. Z., Venieris, I. S. and Kaklamani, D. I.: An Event-driven Health Service Bus. In: Proceedings of the 5th EAI International Conference on Wireless Mobile Communication and Healthcare - Transforming healthcare through innovations in mobile and wireless technologies (MobiHealth 2015), October, 14–16, 2015, London, Great Britain.
2. Wortley, D.: How Wearable Devices Can Impact Corporate Health and Competitive Advantage. Cutter IT Journal, Vol. 28, No 9. Cutter Consortium, 2015.
3. SuperTracker, <https://www.supertracker.usda.gov/default.aspx>.
4. Eat this much, <https://www.eatthismuch.com/>.
5. Glooko, <https://www.glooko.com/>.
6. Sen.se, <https://open.sen.se/>.
7. Kaur, K., Rani, R.: A Smart Polyglot Solution for Big Data in Healthcare. In IEEE IT Professional Smart Systems, vol.17, no. 6, pp. 48–55, IEEE (2015).
8. IBM: IBM Enterprise Service Bus for Healthcare. Solution Brief, IBM Software Group (2010).
9. IBM: IBM’s Healthcare Integration Solution. Solution Brief, IBM Software Group (2012).
10. Ryan, A., Eklund, P. W.: The Health Service Bus: An Architecture and Case Study in Achieving Interoperability in Healthcare. In: Proceedings of the 13th World Congress on Medical Informatics, pp. 922–926, IOS Press (2010).
11. Evans, D.: The Internet of Everything: How More Relevant and Valuable Connections Will Change the World. Cisco IBSG (2012).
12. FHIR® – Fast Health Interoperable Resources. <http://www.hl7.org/implement/standards/fhir/>.
13. D. Bender, and K. Sartipi, “HL7 FHIR: An Agile and RESTful approach to healthcare information exchange”, in Proceedings of the IEEE 26th International Symposium on Computer-Based Medical Systems (CBMS), pp. 326–331, 2013.
14. LOINC, <http://loinc.org/>.
15. Geraci, A.: IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries. IEEE Press, 1991.
16. IEEE Standards Association (PHD - Personal Health Device), <https://standards.ieee.org/develop/wg/PHD.html>.
17. ISO, <http://www.iso.org>.
18. IEEE: ISO/IEEE 11073-10415:2010, Health informatics – Personal health device communication – Part 10415: Device specialization – Weighing scale (2010).
19. IEEE: ISO/IEEE 11073-10441:2013 - Health Informatics – Personal health device communication Part 10441: Device specialization–Cardiovascular fitness and activity monitor (2015).
20. Meridou, D. T., Papadopoulou, M.-E. Ch., Kasnesis, P., Patrikakis, C. Z., Lamprinakos, G., Kapsalis, A. P., Venieris, I. S., Kaklamani, D.-T. I.: The Health Avatar; Privacy-Aware Monitoring and Management. IEEE IT Professional Wearable Computing. IEEE, 2015.
21. Carroll, J. J., Bizer, C., Hayes, P., and Stickler, P.: Named graphs, provenance and trust. In: Proceedings of the 14th international conference on World Wide Web, pp. 613–622. ACM, 2005.
22. American Heart Association, <http://www.heart.org/HEARTORG/>.

23. American Heart Association: Understanding blood pressure readings, http://www.heart.org/HEARTORG/Conditions/HighBloodPressure/AboutHighBloodPressure/Understanding-Blood-Pressure-Readings_UCM_301764_Article.jsp#.VoVLKFJcuVA.
24. American Heart Association: American heart association recommendations for physical activity in adults,;: http://www.heart.org/HEARTORG/GettingHealthy/PhysicalActivity/FitnessBasics/American-Heart-Association-Recommendations-for-Physical-Activity-in-Adults_UCM_307976_Article.jsp.
25. Mobihealthnews: Fitbit changes the way it tracks active minutes, <http://mobihealthnews.com/42241/fitbit-extends-minimum-time-frame-for-active-minutes/>.
26. Hu, V. C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K.: Guide to Attribute Based Access Control (ABAC) Definition and Considerations. NIST Special Publication 800-162, Computer Security. 2014.
27. Papagiannakopoulou, E., Koukovini, M., Lioudakis, G., Garcia-Alfaro, J., Kaklamani, D. I., Venieris, I. S., Cuppens, F., Cuppens-Boulahia, N.: A privacy-aware access control model for distributed network monitoring. Computers and Electrical Engineering, 2012.
28. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-Policy Attribute-Based Encryption. In: Proceedings of the 2007 IEEE Symposium on Security and Privacy (S&P 2007), Oakland, USA, 2007.
29. Openstack, <https://www.openstack.org/>.
30. Delikaris, A. I., Patrikakis, C. Z.: My Social Net-Clone. In: Proceedings of the 10th International Scientific Conference eRA-10. Technological Institute of Piraeus, 2015.
31. About IFTTT, <https://ifttt.com/wtf>.
32. Meridou, D. T., Kapsalis, A. P., Papadopoulou, M.-E. Ch., Karamanis, E. G., Patrikakis, C. Z., Venieris, I. S. and Kaklamani, D. I.: An Ontology-based Smart Production Management System. IEEE IT Professional Smart Systems, November-December, 2015, pp. 36–46.
33. Mongay Batalla, J. and Krawiec, P.: Conception of ID layer performance at the network level for Internet of Things. Springer Journal Personal and Ubiquitous Computing, vol.18, issue 2, pp. 465–480, 2014.
34. Mongay Batalla, J., Gajewski, M., Latoszek, W., Krawiec, P., Mavromoustakis, C., Mastorakis, G.: ID-based service-oriented communications for unified access in Io. Elsevier Computer & Electrical Engineering Journal, 2016.