Communications and Control Engineering

Przemysław Ignaciuk
Andrzej Bartoszewicz

Congestion Control in Data Transmission Networks

Sliding Mode and Other Designs

Springer

Communications and Control Engineering

For further volumes:
http://www.springer.com/series/61

Przemysław Ignaciuk • Andrzej Bartoszewicz

Congestion Control in Data Transmission Networks

Sliding Mode and Other Designs

Springer

Przemysław Ignaciuk
Institute of Information Technology
Technical University of Łódź
Wólczańska St 215
Łódź
Poland

Andrzej Bartoszewicz
Institute of Automatic Control
Technical University of Łódź
Stefanowskiego St 18/22
Łódź
Poland

ISSN 0178-5354
ISBN 978-1-4471-4146-4   ISBN 978-1-4471-4147-1 (eBook)
DOI 10.1007/978-1-4471-4147-1
Springer London Dordrecht Heidelberg New York

Library of Congress Control Number: 2012944653

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Information sharing plays a key role in functioning and evolution of modern societies. The access to appropriate information frequently affects the choices we make, thus influencing our life both on the professional and private levels. It is of paramount importance then to provide this access as fast as possible and with appropriate guarantees.

The backbone of information sharing systems is formed by data transmission networks. The networks supply the means of communication and give crucial functionality for successful data exchange. A “successful” data exchange implies that a suitable service level has been achieved while delivering the data, since only then a particular application (or service) can be regarded useful for the partners in communication. In consequence, apart from the mere fact of relaying the data pieces from one point to another, the network should ensure that they are transmitted in an appropriate way. The service level, or Quality of Service (QoS), reflects the way the data is transported through the network. Various applications require different service level guarantees measured through performance indicators, such as error rate, throughput, or average delay. Since a data transmission network is inherently a distributed system, in order to establish good values of the performance indicators, efficient means of resource sharing and traffic management should be implemented. Among the traffic regulation mechanisms, congestion control (or data flow control) plays a key role in ensuring coordinated access to the available resources. In particular, it is vital for ensuring appropriate, *dynamical* load adjustment according to the changing networking conditions.

Despite the considerable research effort lasting for more than three decades now, efficient data flow regulation remains a challenging issue. This is mainly attributed to the intrinsic complexity of communication systems and unceasing need for adding new and more sophisticated services to the existing infrastructure. Nowadays, the traditional objective of efficient and fair resource usage is augmented by other QoS requirements that should be considered in the control strategy design, such as high reliability in banking information delivery, or low delay variation in video transmission.

Meeting the QoS requirements poses a serious challenge when the distance between the communicating partners increases. When the information exchange takes place on large geographical areas, as is frequently the case in the Internet, the data needs to pass through multiple nodes and interconnecting links. Therefore, in addition to higher risk of losses, the long-distance communication imposes substantial latency caused by the physics of signal propagation and information processing. The data pieces traveling through the network are subject to propagation delay along the links, processing latency at the nodes, and queuing delay in the buffers. Moreover, the overall latency for the data stream may change during the transmission depending on the buffer state and the path the pieces are directed along. As a consequence, a good flow control strategy for modern communication networks should explicitly incorporate suitable mechanisms for handling the effects related to delay. These mechanisms should provide compensation for delays spanning the range from several milliseconds (local flows) to even a few seconds (in global data exchange). They should also ensure robustness to latency variations and other perturbations affecting the control process.

This work is devoted to the application of control-theoretic methodology to the communication system modeling and design of congestion control algorithms. The primary objective of the monograph is twofold. First of all, we show how various networking phenomena can be represented in a consistent mathematical framework which is suitable for rigorous formal analysis. We differentiate between fluid-flow continuous-time traffic models, discrete-time processes with constant sampling rate, and sampled-data systems with variable discretization period. The second fundamental objective of this work is to provide appropriate control mechanisms which can handle the congestion and guarantee high throughput in various traffic scenarios (with different networking phenomena considered). We propose a systematic design approach using sound control-theoretic foundation. Since the robustness issues are of major concern in providing efficient data flow regulation in today’s networks, sliding-mode control is selected as the principal technique to be applied in constructing the control algorithms. The controller derivation is given extensive analytical treatment and supported with numerous realistic simulations.

Acknowledgments

Our work presented in this book was developed with the kind support of the National Science Centre of Poland provided under the research project "Optimal sliding-mode control of time-delay systems." This work was also possible due to the generous help granted by the Foundation for Polish Science (FNP). Therefore, financial assistance of the foundation, obtained by the first author of this monograph – Przemysław Ignaciuk – under young researcher support program, is gratefully acknowledged. Furthermore, a few projects sponsored by the Polish State Committee for Scientific Research and Polish Ministry of Science and Higher Education in the past 15 years also helped initiate and – to some extent – develop many ideas reported in this book.

The research reported in this monograph has benefited greatly from collaboration with a number of Ph.D. students and many friends at Technical University of Łódź, Poland. Therefore, we wish to thank all of them for their encouragement, cooperation, and stimulating discussions while we were preparing this book. In particular, we acknowledge much needed, constructive help of Dr. Antoni M. Zajączkowski and Dr. Michał Morawski who assisted us in many ways during the last couple of years when we have been working on this book. We also wish to thank all those individuals at Technical University of Łódź, Poland, who gave us the opportunity to undertake and continue the research presented in this book. We must also thank our families for their patience and continuous moral support during the period when this monograph was written.

Finally, we wish to thank the entire team of Springer publications for allowing the preparation of this book to proceed. We hope that the result of our joint effort will be of true interest to the control community working on various aspects of congestion control in data transmission networks, and also those working on variable structure control of time-delay systems.

Technical University of Łódź, Poland Przemysław Ignaciuk
Technical University of Łódź, Poland Andrzej Bartoszewicz

Contents

Abbreviations

ABR	Available Bit Rate
AQM	Active Queue Management
ATM	Asynchronous Transfer Mode
DT	Dead time
DTC	Dead-time compensator
ECN	Explicit Congestion Notification
IP	Internet Protocol
ISM	Integral sliding mode
LQ	Linear-Quadratic
PD	Proportional-Derivative
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
QoS	Quality of Service
RED	Random Early Detection
RM	Resource Management
RTT	Round-trip time
SM	Sliding mode
SMC	Sliding-mode control
SP	Smith predictor
TCP	Transmission Control Protocol
VC	Virtual circuit
VSS	Variable structure system

List of Symbols

$\mathbf{A}$	State matrix
$\mathbf{A}_{\mathbf{cl}}$	Closed-loop state matrix
$\mathbf{b}$	Input vector
$\mathbf{c}$	Vector of hyperplane parameters
$C(\cdot)$	Link capacity
$d(\cdot)$	Available bandwidth
d_{max}	Maximum available bandwidth
$\det(\cdot)$	Determinant of a matrix
$\mathbf{e}(\cdot)$	Error vector
$\mathbf{g}$	Optimal gain vector
γ, $\mathit{\Gamma}$	Controller gain
$h(\cdot)$	Utilized bandwidth
$J(\cdot)$	Dynamical performance index
k	Index of sampling instant in discrete-time systems
λ_p	Fraction of the overall rate allocated to connection p
m	Number of connections
n	State-space system order
$N(\cdot)$	Number of TCP sessions
$p(\cdot)$	Probability of packet mark (or drop)
$\mathbf{q}$	Output vector
$q(\cdot)$	Packet queue length
$R(\cdot)$	Feedback loop delay
$s(\cdot)$	Sliding variable
$\mathrm{sgn}(\cdot)$	Signum function
$\sigma(\cdot)$	Switching function
t	Time (continuous variable)
T	Discretization period
T_{B}^{p}	Backward delay in connection p
T_{F}^{p}	Forward delay in connection p
$u(\cdot)$	Controller command

$u_{\max}$	Maximum transmission rate
$u_p(\cdot)$	Transmission rate of source p
$u_{\mathrm{R}}^{p}(\cdot)$	Incoming rate from source p
$v_p(\cdot)$	Rate allocated for source p
$W(\cdot)$	TCP window size
$\mathbf{x}(\cdot)$	State vector
$x_i(\cdot)$	ith state variable
$y(\cdot)$	Output variable
y_{D}	Demand queue length
$y_{\max}$	Maximum queue length

Chapter 1
Introduction

In recent years, we have experienced rapid evolution of networking services. We can find widespread network traffic related to web browsing, e-mail, electronic trade and money transfer, telematics (vehicle positioning, crash notification, etc.), video streaming, remote visualization and steering, Internet telephony, etc. As a consequence, in addition to the increased intensity, the today networks need to handle also the diversity of data streams and meet their QoS requirements. Together with serving the applications demanding fast and reliable information interchange (e.g., banking transactions), modern data transmission networks are expected to provide high-throughput, low-jitter, end-to-end connectivity important for multimedia transmission. The connectivity needs to be available to both local and long-distance streams without violating fairness constraints. In the following sections, we will recall the fundamental concepts of organizing data transfer in communication networks and elaborate on a major obstacle to obtaining appropriate QoS – the congestion.

1.1 Data Transfer Concepts

The exchange of information in a data transmission network takes place between two end points. The end point injecting the data into the network is referred to as the source (or transmitter) and the other end point as the destination (or receiver). In bidirectional communication, both end points serve the role of source and destination, depending whether they are transmitting or receiving the data. Typically, the stream generated by the data source is partitioned into units of appropriate size – packets. In the network, packets pass through a series of nodes until they are delivered to their destination. Depending on how a particular stream is handled by the nodes, basically, we can differentiate between connection-oriented and connectionless types of networks.

P. Ignaciuk and A. Bartoszewicz, *Congestion Control in Data Transmission Networks: Sliding Mode and Other Designs*, Communications and Control Engineering,
DOI 10.1007/978-1-4471-4147-1_1, © Springer-Verlag London 2013

In connection-oriented networks, a virtual circuit (VC) is created between the communicating entities before the actual exchange of data commences. Thus, prior to the transmission of data, the network sets a path through a sequence of nodes for the entire data stream. Also in the connection setup phase, appropriate switching rules at the nodes are established to quickly move the incoming packets between the input and output node interfaces. Afterwards, the stream of packets follows the fixed path with no need to examine full address data of the ultimate destination by the nodes from the individual packets. As a result of reducing the time needed for taking the routing decisions, the transmission efficiency can be increased. Moreover, transferring the data along the preestablished, fast-switched path extends the possibilities for loss reduction, more efficient resource utilization, and QoS enhancements. In particular, it allows for decreasing the average propagation delay and delay variation, which are critical for multimedia applications. Some noticeable examples of connection-oriented technologies are, for instance, Asynchronous Transfer Mode (ATM) [1], Multiprotocol Label Switching (MPLS) [10], Generalized MPLS (GMPLS) [9], Pseudo Wire Emulation Edge-to-Edge (PWE3) [2], or UDP-based data transfer (UDT) [3, 4]; see also recent special issues reporting on the advancements in connection-oriented networks [5, 6].

In spite of the numerous advantages of connection-oriented technologies, the majority of data traffic in the today Internet is of connectionless type, regulated by the family of TCP/IP (Transmission Control Protocol/Internet Protocol) network protocols. In connectionless networks, the route for the stream of packets generated by a source is not established *a priori*. Each packet arriving at a network node is routed individually. The node reads full address data of the destination point from the packet header, determines the appropriate output interface from the routing table, and finally forwards the packet on this interface to the next node on the path towards the destination. Due to the necessity of extracting full address data from the header, the switching time – the time of information processing and directing the packet between the input and output interfaces – grows. However, a more serious drawback of the connectionless data transfer is the increased risk of congestion. Note that the lack of connection setup phase implies that the network has fewer means of regulating which flows should be accepted and which should be rejected due to insufficient resources. As a rule, a connectionless network accepts all the incoming traffic. Therefore, in addition to longer delays the packets experience at the nodes, in connectionless networks, there is a bigger chance of the traffic intensity growing beyond the available channel capacity which leads to congestion. Thus, the need for efficient control schemes that will *dynamically* adapt the traffic intensity to the changing networking conditions is perhaps more profound in connectionless networks than in connection-oriented ones. The advantage of connectionless technologies, in turn, is smaller initial signaling overhead and increased robustness to errors related to breaking the connection. If in a connection-oriented network the established connection is cut (for instance, due to a link failure), the setup phase needs to be repeated. Since in connectionless networks each packet is treated individually, the lack of possibility of sending a packet along the optimal route need not terminate the ongoing transmission. Typically, a link failure reduces to switching the traffic to an alternative (backup) path.

However, no matter the choice of the transmission technology, either connectionless or connection-oriented, in order to guarantee good network performance, appropriate means of controlling the access to network resources need to be provided. In particular, when high service level is required, the available resources need to be administered in a well-coordinated manner, according to the dynamically changing networking conditions (varying traffic intensity, number of active connections, buffer levels, etc.). If the network load increases beyond the channel capacity, the congestion occurs, leading to packet discards, retransmissions, and throughput degradation. If, on the other hand, the traffic intensity excessively drops, then only part of the available bandwidth at the intermediate links is used for the data transfer, and the revenue of the telecommunication service providers decreases. Therefore, in order to combat the congestion and at the same time ensure high throughput in the network, it is not sufficient to merely extend the physical channel capacity, introduce connections with higher bit rates, or install faster transceivers. A successful solution to the dynamical resource allocation problem must involve the use of appropriate flow control mechanisms that will guarantee stable and efficient network operation.

1.2 Congestion

Let us study the phenomenon of congestion in more depth. We will analyze the events related to transmission of data in the example network depicted in Fig. 1.1. The throughput associated with each discussed case is illustrated in Fig. 1.2.

The network shown in Fig. 1.1 consists of six nodes, numbered 1–6, which are connected by the links of equal capacity. There are two sources of data, S1 and S2, which send packets to destinations D1 and D2, respectively. If the sources are allowed to transmit packets at the maximum rate permitted by the capacity of the links between the nodes N1–N3 and N2–N3 (graph a in Fig. 1.1), then the incoming rate at node N3 is twice the outflow rate at the output connection N3–N4. Consequently, not all the packets can be immediately forwarded at link N3–N4, and the bottleneck is formed. Certain packets need to be stored in the buffer allocated to that link at node N3, where they wait to be forwarded towards the destination. However, since any physically realizable buffer can only be of finite capacity, at some moment in time, the buffer at node N3 will become entirely filled with data. The incoming packets that meet the buffer fully occupied are discarded, which leads to the throughput decrease (see Fig. 1.2, curve a). The network state when packets are discarded due to the insufficient bandwidth at the outgoing link and buffer overflow is referred to as the congestion. The link where the bottleneck is formed is termed the bottleneck link. If the situation of the buffer overflow persists,

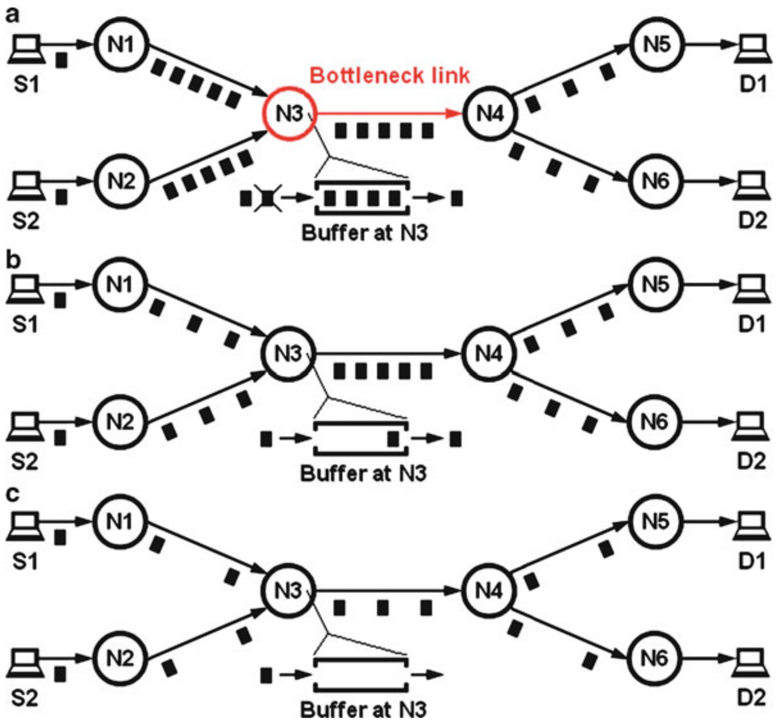
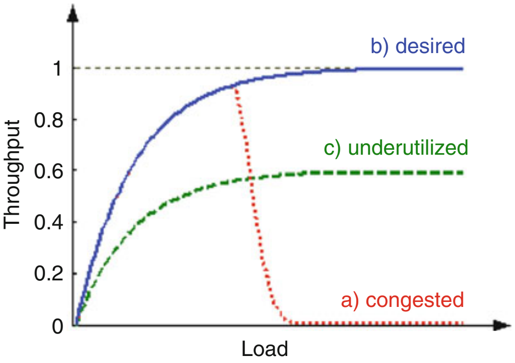


Fig. 1.1 Network states: (**a**) congestion, (**b**) desired conditions, and (**c**) system underutilization

Fig. 1.2 Throughput vs. network load: (**a**) congestion, (**b**) desired network state, and (**c**) system underutilization



a network breakdown called the congestion collapse may occur. In that case, the effective throughput tends to zero. To see how it happens, notice that the packets rejected at node N3 need to be retransmitted by the sources later in time. At some point, the data transfer will be dominated by the retransmissions of the discarded packets. In this way, the throughput will approach zero.

It follows from the preceding example that there are a few conflicting objectives to be met to obtain satisfactory network performance. One aims to achieve high bandwidth utilization, since it affects the revenues from using the existing infrastructure. On the other hand, keeping the links fully occupied with the transmission of data at all times may lead to excessive queue growth. As a result, the delay expands, downgrading the service level. Long queues also enforce the use of larger buffers and increase the risk of congestion which may cause devastating throughput decrease. In order to obtain high throughput, low risk of congestion, and small queues in the dynamically changing environment, one has to introduce appropriate control mechanisms.

The purpose of this monograph is to elaborate on the methods of efficient data flow control in communication networks. In the literature, some authors prefer to restrict the term "congestion control" to the actions related to recovering from the state of congestion and reserve the notion of "flow control" to a broader spectrum of mechanisms responsible for maintaining good network performance (including the congestion recovery). However, in many publications, the terms "congestion control" and "flow control" are used interchangeably. We adapt this paradigm, and whenever discuss the congestion control, we refer to all the actions necessary to achieve smooth flow of data and suitable service level in the network, not just the procedures for recovering from the ongoing congestion.

First of all, in this work, a mathematical framework is developed to cover a wide range of networking phenomena, including transfer rate updates at discrete-time instants, data source nonidealities, delay variations, packet reordering, etc. A number of flow control strategies are designed following rigorous, control-theoretic methodology. It is shown that the elaborated strategies ensure stable system operation in the presence of long, disparate delays in the communication system. Moreover, it is demonstrated that the risk of congestion can be eliminated solely with the use of the proposed control algorithms, i.e., without resorting to additional mechanisms, such as backpressure [8], or sophisticated switch service disciplines, for example [7]. It is also shown that the loss elimination can be obtained while the links are kept fully occupied with data transfer, which yields the maximum throughput in the communication system in each of the considered traffic scenarios. The maximum throughput is guaranteed even if the feedback information necessary for rate adjustment is accessible at irregular time instants, and the system parameters are estimated imprecisely. Furthermore, it is demonstrated that fairness criteria are satisfied, and the algorithm implemented at a node operates correctly in the presence of multiple bottlenecks and can inadversely coexist with other schemes active in the network. It is also discussed how the support for additional QoS enhancements (e.g., delay jitter reduction) can be introduced without violating the buffer capacity constraints or downgrading the bandwidth utilization.

The highlighted subjects are presented in subsequent chapters. The basic concepts and modeling framework are introduced in Chaps. 2, 3, 4, and 5, whereas more advanced topics are covered in Chaps. 6, 7, and 8.

Chapter 2 gives a historical perspective on the evolution of congestion control techniques. We begin with discussing various performance criteria. Then, the early, commonly *ad hoc* proposals for vendor-specific communication systems are presented. Afterwards, the ideas behind various algorithms proposed for standardized connection-oriented and connectionless networks are discussed. The primary emphasis is given to the formal approaches to establishing an appropriate congestion control strategy. The benefits of the application of sound control-theoretic methodology are discussed, and numerous examples of the classical and innovative techniques are provided. A separate section is devoted to the application of sliding-mode control to data flow regulation in communication systems, since it is chosen as the primary technique in the derivation of control schemes elaborated in this monograph.

In Chap. 3, we introduce the fundamental concepts of variable structure systems and sliding-mode control (SMC). We begin with the subject of sliding-mode (SM) controller design in continuous-time domain and discuss the importance of choosing appropriate sliding plane to achieve adequate system performance. Then, we move on to the design methods for discrete-time systems.

Chapter 4 covers the issues related to congestion control in systems with continuous feedback information delivery. The fluid-flow traffic approximation is used to construct relevant network models. The models explicitly incorporate the effects related to delay, both in the single and multisource network configuration. Continuous-time SM controllers are proposed for each of the analyzed cases. The switching functions are selected to stabilize the closed-loop system in the presence of nonnegligible input–output delay. The SM controller performance is compared with the classical control scheme employing the Smith predictor for delay compensation.

In Chap. 5, we elaborate on the discrete nature of crucial networking events, such as transmission rate updates by the sources at the instants of feedback information retrieval from the network. An appropriate network model, which covers the effects related to finite sampling rate and nonnegligible delay, is created. Using carefully chosen state space, the controller design is conducted directly in discrete-time domain. The sliding hyperplane of the fundamental controller is chosen by solving a dynamical optimization problem with quadratic performance index. The obtained closed-form solution of the optimization task allows for a detailed analytical treatment of the controller properties. Next, the issues related to various networking phenomena are discussed, and appropriate modifications of the fundamental strategy introduced to achieve good network efficiency are presented. In particular, we show how to cope with saturating transmitters, delay variations, and packet reordering. In each case, the conditions for achieving the maximum throughput in the network are formulated and strictly proved.

In Chap. 6, the basic concepts introduced in Chap. 5 are extended to the multisource network, in which the connections are characterized by different delays. The optimization problem of sliding hyperplane selection is again solved explicitly, which allows for analytical study of the derived controller characteristics. In addition to reformulating the concepts discussed in Chap. 5 in the context of multisource traffic scenario, a number of issues specific to multisource networks are also addressed. In particular, we elaborate on the fair bandwidth allocation and the existence of multiple bottlenecks. All the controller properties are proved analytically and illustrated in extensive numerical examples.

Chapter 7 is devoted to the data flow control in the networks in which the sampling synchronization cannot be maintained. Thus, in that chapter, we discuss the problems related to variable discretization period. A number of control algorithms are proposed, and each is given full consideration at the analytical level. In addition, QoS issues related to serving the multimedia traffic are addressed, and possible solutions are discussed.

In Chap. 8, we move on to the problem of data flow control restricted to TCP/IP networks with Active Queue Management (AQM) support. First, the choice of the modeling framework and various linearization options are discussed. The linearized model is discretized and represented in the state space, which is selected to explicitly incorporate the effects of delay. The SM controller design is performed directly in discrete-time domain, with the sliding hyperplane chosen for a dead-beat scheme. The proposed controllers are compared with the classical AQM packets marking schemes.

Chapter 9 contains a brief summary of the applied methods and presented results. In that final chapter we also give concluding remarks and discuss possible extensions and future research directions.

In the Appendix, the tools used for validation of the constructed models and testing the proposed control solutions are discussed. A reference to the web page with the online material associated with the tests is provided. On the indicated web page, one can find source files, sample scripts, and graphs illustrating the results of selected simulations performed in Matlab-Simulink and ns2.

References

1. ATM Forum (2002) ATM User-Network Interwork Interface (UNI) specification version 4.1
2. Bryant S (2005) Pseudo Wire Emulation Edge-to-Edge (PWE3) architecture. IETF RFC 3985
3. Grossman RL, Gu Y, Hong X, Antony A, Blom J, Dijkstra F, de Laat C (2005) Teraflows over gigabit WANs with UDT. Future Gen Comput Syst 21:501–513
4. Gu Y, Grossman RL (2005) UDT: UDP-based data transfer for high-speed wide area networks. Comput Netw 51:1777–1799
5. Habib I, Jajszczyk A, Awduche D (2006) Advances in control and management of connection-oriented networks. IEEE Commun Mag 44:58–59
6. Habib I, Jajszczyk A, Awduche D (2007) Advances in control and management of connection-oriented networks: part 2. IEEE Commun Mag 45:76
7. Izmailov R, Lee DS, Sengupta B (1997) Design and analysis of a congestion-free overlay on a high-speed network. IEEE/ACM Trans Netw 5:970–980
8. Karol M, Golestani SJ, Lee D (2003) Prevention of deadlocks and livelocks in lossless backpressured packet networks. IEEE/ACM Trans Netw 11:923–934
9. Mannie E (2004) Generalized Multi-Protocol Label Switching (GMPLS) architecture. IETF RFC 3945
10. Rosen E, Viswanathan A, Callon R (2001) Multiprotocol Label Switching architecture. IETF RFC 3031

Chapter 2
Congestion Control in Data Transmission Networks: Historical Perspective

The congestion occurs when the traffic generated by the network users exceeds the available bandwidth in the communication system. In such circumstances, not all the packets sent by the sources can be immediately relayed on the route towards their destination. Instead, they accumulate in the buffers at the intermediate nodes and wait for the bandwidth increase. If the incoming rate is not reduced (or stopped) before the queue of awaiting packets reaches its limit, typically defined by the amount of the reserved memory at the node, the new data pieces must be discarded. The lost fragments are retransmitted, which further deepens the congestion at the bottleneck point. At certain stage, the network becomes clogged with retransmissions and stops providing its services – this state is referred to as a deadlock or congestion collapse. In fact, the early communication networks frequently suffered from congestion collapse, until the development of the Jacobson's scheme [73] for the Internet flow control.

It is obvious that appropriate measures should be taken to recover from the congested state, or even better, to react to deteriorating transfer conditions before the congestion actually happens. The question arises whether one can solve this problem by eliminating the resource deficiency, i.e., by extending the node memory, introducing rapid links and faster processors, or using better signal processing and switching techniques, such as those envisaged in [50, 151, 154]. Unfortunately, it has been shown [78] that the application of modern, state-of-the-art technologies at the physical layer alone does not provide a satisfactory countermeasure to the congestion problem. In fact, the installation of physical layer enhancements may even downgrade the system performance at the interface of slow and fast networks [77]. Therefore, the necessary solution to the congestion problem must involve the use of efficient dynamical resource allocation algorithms operating at the logical level [77]. The technological advancements should be regarded as an imminent, yet long-term evolutionary improvement to the communication process, required for creation and implementation of new, possibly sophisticated and functionally challenging networking services, rather than an immediate answer to the congestion threat.

P. Ignaciuk and A. Bartoszewicz, *Congestion Control in Data Transmission Networks: Sliding Mode and Other Designs*, Communications and Control Engineering,
DOI 10.1007/978-1-4471-4147-1_2, © Springer-Verlag London 2013

In this chapter, a summary of various solutions to the congestion control (or alternatively flow control) problem in telecommunication networks considered so far in the literature is presented. The chapter is organized in the following way. First, in Sect. 2.1, the key comparison criteria of flow control algorithms are discussed. Next, in Sect. 2.2, preliminary approaches to congestion control are described, contrasting in particular the rate and credit-based concepts. Section 2.3 gives an overview of various heuristic rate-based solutions. More recent proposals, involving popular analytical tools, are discussed in Sect. 2.4. Section 2.5 summarizes the developments in sliding-mode controller design for traffic regulation in data transmission networks. Finally, this chapter concludes with the discussion of the benefits of employing formal methodology in the design of flow controllers in modern networks.

2.1 Comparison Criteria

The technological diversity of telecommunication systems together with variety of traffic types and the resultant panoply of data transfer related phenomena greatly complicate the evaluation of different flow control algorithms. A properly designed controller for an application requiring small variations of transfer delay (e.g., video streaming) may not be appropriate for handling a loss-sensitive service (e.g., stock exchange feeds). On the other hand, the favored loss elimination may downgrade the network utilization – greedily desired by the telecommunication operators – if the system is overprovisioned. Therefore, in order to provide a comparison among various data flow control techniques, numerous aspects need to be considered [44, 79, 109, 127, 157]:

1. Telecommunication system modeling

   - *The level of details* – the practical usefulness of the developed control scheme depends on how closely the model applied in the design resembles the behavior of the real object. According to [73], the description of core communication process, which consists of data emission at the sources, transmission through the network, and reception at the destination side, to a large extent can be approximated by means of linear blocks, such as gains, integrators (which represent the packet accumulation in buffers), and delay elements (which model the link propagation latency). However, the other phenomena, such as transmitter saturation, time-varying delay, data segment granularity, or packet dropout, usually require the use of nonlinear blocks, which complicates the design and property analysis [124, 153]. A good compromise applied by many researchers, for example, [21, 46, 70, 148], is to reduce the number of details for the purpose of the mathematical derivation and verify the results obtained for the (simplified) nominal model in more realistic simulation scenarios with nonlinearities and element nonidealities included.

- *Parametric uncertainty* – the exact system modeling does not guarantee that parameters describing the object, and influencing the operation of the designed flow control algorithm (e.g., propagation delay or number of connections), are measured or estimated precisely. Moreover, the feedback information – vital for the proper rate adjustment – may be lost or corrupted by errors. Consequently, the congestion control algorithm should function correctly despite possible inaccuracy in parameter estimation and the disturbances of the feedback information delivery [33, 131]. In order to protect the legitimate sources (those which respond precisely to the controller commands), it should also offer a possibility for isolating the nonconforming ones (those which violate the traffic agreement and rate intensity adjustments) [79].
- *Continuous- vs. discrete-time modeling* – the network variables (e.g., packet queue length in a link buffer, transmission rate of a source, or link bandwidth) may be represented as signals either continuous or discrete with respect to time. The continuous-time representation of data transmission process (especially fluid-flow approximation) demonstrates enormous potential in modeling complex network behavior in a manageable way [42, 116, 158]. However, any communication network, in essence, is a discrete event system – the state of network variables changes in response to certain events, for example, emission of a packet, control unit reception, and rate adjustment – and is more accurately described by discrete-time functions. Consequently, since certain signals may be accessible at the communicating entities at discrete instants only (e.g., the feedback information passed to the data sources in control units by the network nodes), it is desirable to consider the effects of sampling in the flow controller design either explicitly [29, 106, 160] or through discrete-time simulations [42, 46, 127].

2. Implementation issues

   - *Implementation cost* – the developed scheme should respect the existing hardware limitations and the standardization of the transmission protocols and equipment. Installation of new hardware and infrastructure rebuilds as well as complex architectural design and software coding increase production cost and time to market. Therefore, the designed congestion control strategy should be easy to understand and straightforward in implementation, possibly within the existing framework.
   - *Operational efficiency* – expenditures for installation and maintenance of networking equipment and cabling are economically justified only if the available resources are governed efficiently, thus ensuring return on the invested capital [128, 144]. The developed scheme should achieve high level of bandwidth utilization (ideally, the entire link capacity should be employed for data transfer) [24], with reasonable processor power and node memory consumption.
   - *Administration simplicity* – large number of difficult to understand parameters and complex performance tuning discourage technical experts from using a particular protocol in the administrated network. In addition to the

configuration burden, numerous coefficients and scaling factors tend to reduce the system robustness to parametric uncertainty and make the algorithm prone to errors [44]. Consequently, a desirable congestion control solution should be simple and intuitive in installation and should allow for automation of the management tasks.

3. Steady-state characteristics and transient response

   - *Controller dynamics* – data transmission networks are dynamical systems and, as such, require careful investigation of both the steady-state and transient behaviors. The designed flow control algorithm should promptly react to the changes of networking conditions and quickly reach steady state, possibly without oscillations and overshoots [111]. Short settling time and smooth state transition enhance transmission consistency (vital for multimedia applications which, in principle, do not tolerate abrupt rate modifications) and improve memory management policy.
   - *Signal constraints* – any flow control algorithm developed for communication networks should guarantee that the generated transmission rates are always nonnegative and upper-bounded, and the queue length is finite both in steady state and during transient periods.

4. System-related features

   - *Scalability* – the developed congestion control solution should demonstrate potential for effective operation in the case when the telecommunication system expands and the number of regulated flows increases, and it should be applicable to both the local and long-distance traffic.
   - *Fairness* – since a communication network provides the transport services to multiple users, it requires appropriate mechanisms of resource allocation among the contending flows. In particular, this concerns the method of memory and bandwidth distribution, which should be efficient, yet fair. However, fairness itself is not a precise term and can be interpreted in a variety of different ways [58, 78, 107]. Intuitively, a fair allocation can be perceived as the one depending neither on the topological (e.g., the source relative position in the network or the path established for a data stream) nor on timing factors (the moment when the transmission commences). In practical terms, quantitative measures are needed to assess fairness and avoid flow discrimination (or preferential treatment). For this purpose, appropriate mathematical criteria are formulated, usually in the form of a utility function, and the resource allocation vector for competing sources is obtained from an optimization algorithm [44, 132]. One of the most commonly applied criteria of fairness in communication networks is the max-min allocation [74], whose objective is to maximize the overall throughput in a multiuser multi-node system by satisfying the needs of the least demanding flows and reducing the load created by the most congesting ones.
   - *Interoperability* – a telecommunication network is a system of interconnected nodes, which implies that the designed traffic regulation protocol should be

prepared to operate in a distributed environment (to be implemented at multiple nodes) [31]. Another complication resulting from the distributed nature of computer networks, which, in principle, are governed by autonomous organizations and business entities, is the diversity of functioning technologies and transmission protocols. Consequently, a successful strategy for heterogeneous multilayer networks should safely coexist with other, possibly different flow regulation schemes [55].

- *Management overhead* – telecommunication operators focus on maximizing the economic gains from their infrastructure and are primarily interested in using their network to transfer the customers' data, not the management information [24, 128]. Consequently, the amount of the exchanged control information necessary for the flow regulation should be kept at the minimum in relation to the invoiced traffic.
- *QoS support* – the worldwide proliferation of applications and networking services, demanding specific treatment of the generated data stream (interactive telephony, video on demand, remote sensing), stimulates the search for new and better methods of handling various traffic types in the traditionally best-effort communication paradigm [15, 32, 39, 53, 57, 170]. Consequently, modern flow control strategy should not only avoid the transmission bottlenecks and keep the throughput high, but it should also improve the quality-related requirements of various traffic kinds, such as loss rate, reliability, average transfer delay, latency variation (jitter), and buffer queuing time [51].

The complexity of the developed flow control algorithm grows with the number of problems addressed in the system modeling and design procedures. In addition, the design objectives do not always coincide with each other (some goals are typically achieved only at the expense of others), and trade-offs typically need to be applied in search for an optimal solution, for example, versatility vs. simplicity or bandwidth utilization vs. fairness. Probably, no single scheme can adequately answer all the issues [127], yet it should at least satisfy the fundamental requirement of bounded transmission rates and finite queue length. Researchers argue about the other features that a successful scheme should provide. Usually, the importance of efficiency, simplicity, robustness, scalability, fairness, and enhancements for QoS support is stressed. These measures will constitute the key comparison criteria for various congestion control mechanisms discussed in the further part of this chapter.

2.2 Early Concepts of Congestion Control

The lack of standardization in the early communication protocol development, especially in the core part of the existing networks, made the initial flow control approaches heavily dependent on the actual transmission technology for which they were designed. The research area was primarily limited to the proprietary solutions of various interest groups and companies. Different concepts were applied to connectionless and connection-oriented networks. Severe inconsistencies could be observed in the interaction of various layers of the communication process, even if a protocol was designed and implemented within the single-vendor equipment. A detailed, comprehensive study of those primary flow control strategies is given in [58]. The authors of [58] formulate basic evaluation criteria of throughput, delay, and power and compare the operation of the existing congestion control schemes and emerging solutions with distinction of the operational range (access, end-to-end, or hop-by-hop) and switching technique (datagram connectionless vs. VC based). Numerous predictions about future research, such as the concurrent control of integrated voice and data services, combined routing and flow management, or focus on congestion avoidance rather than congestion recovery solutions, proved extremely accurate and actually dominated scientific investigation in the field of data traffic regulation for more than a decade [77, 79, 115].

One of the first congestion control schemes for connection-oriented networks was developed for TYMNET [143], centrally supervised international communication system, and its improved version TYMNET II [164]. In TYMNET, during the connection setup, a throughput limit is calculated for each VC (according to the terminal speed) which is enforced at all the nodes along the established route. Flow control is obtained by assigning memory quota at each intermediate node and sending transfer permits based on the quota exhaustion between the neighboring switches. A transmitter sets a counter equal to the maximum buffer size (quota), which is decremented with each data piece (character in TYMNET) relayed on the transmission path. Periodically, each node sends backpressure vector to its neighbors, containing a binary flag for each VC passing through the node. The flag is set to zero if the assigned buffer is entirely filled with data (the maximum permitted allocation is reached). The transmitter stops data transfer when the counter reaches zero and resumes it once the received backpressure bit for the corresponding VC is equal to one. Backpressure propagates from node to node back to the source and finally slows it down or turns off. Although efficient and cost-effective for low-speed terminals, the backpressure mechanism may introduce unsatisfactorily large delays when transmitting bigger volumes of data, especially real-time ones [59]. Moreover, fairness with backpressure flow control is guaranteed only when per-VC queuing is applied (in common-buffering scheme, sources not using the congested resource may be blocked due to backpressure cascade effect) [77], which does not scale well with the increase of the VC number [79].

In TRANSPAC [48], the French public data network, the throughput class concept of the X.25 internode protocol was employed to control the congestion. At the connection setup, each VC declares its peak instantaneous rate that is used by the nodes to estimate the maximum aggregate throughput. By monitoring the average buffer occupancy and the actual throughput in relation to the declared one, each node dynamically adjusts a set of thresholds used to limit the load. Depending on the severity of congestion, the current flows are slowed down (by delaying the return of acknowledgments at the VC level), new VC requests are rejected, or the existing ones are terminated. Again, the necessity of per-VC queuing required for fair operation of the proposed algorithm raises serious scalability issues for growing number of connections. Moreover, the efficiency of the analyzed scheme highly depends on the appropriate choice of thresholds, which may not be a straightforward task in the distributed environment.

The use of a combined approach of the low-level deadlock avoidance and high-level flow control was investigated in GMDNET [60, 138], experimental network created in Darmstadt, Germany. The deadlock avoidance is resolved by creating a class-differentiated pool of buffers at each node – structured buffer pool (SBP). The packets arriving at a node are divided into classes according to the number of hops they have traversed, and stored in the buffers of the corresponding level, or a level with lower priority. If an incoming packet meets all the buffers available for its class fully occupied, it is discarded. Consequently, the older packets (for which the network has invested more resource in delivery) are given preference over the junior ones. The flow control on each VC is performed with variable size windows, defined as the number of packets a sender can transmit before the receipt of an acknowledgement or a permit is received, which are adjusted both locally (on the hop-by-hop basis) and globally (end-to-end). The drawback of the proposed scheme lies in the necessity of a balanced, well-tuned cooperation of both SBP and windowing flow regulation. Either of the mechanisms working alone cannot prevent throughput degradation, deadlock, and unfairness [58].

Systems Network Architecture (SNA) [2] developed to provide distributed communication services for IBM systems used a similar hybrid of the hop-by-hop and end-to-end flow control strategies. The traffic regulation in this type of networks is exercised individually on each VC through dynamic window resizing at the end point as well as at the intermediate communicating entities. Besides subtle differences, for example, the IBM solution enforces the sender to always ask for permission to transmit another set of packets (sent in the first packet of the current window), both GMDNET and SNA control mechanisms behave alike and are subject to analogous limitations.

With the advent of the ATM technology and recommendation of the Consultative Committee on International Telephony and Telegraphy (CCITT) – an international organization responsible for telecommunication standards (replaced in 1993 by International Telecommunication Union (ITU)) – to use it for Broadband Integrated Services Digital Network (B-ISDN), the research in the field of data flow control significantly intensified. ATM employs the benefits of packet and circuit switching to provide high-speed data transfer both locally and on large distances. It was designed to provide strong support for QoS of class-differentiated traffic, thus eliminating limitations of the traditional best-effort networks. In order to accomplish this ambitious goal, ATM was built on the concept of a connection-oriented network (where the path for a data flow is established prior to the actual data transfer and usually does not change during the transmission unless a node or link failure occurs), combined with stream partitioning into short, fixed length cells (53 bytes). As a result, low-jitter long-haul communication was possible at the rates exceeding 155 Mb/s. For the purpose of QoS provision for different kinds of applications, several service categories were specified in ATM [18]. Among the proposed categories, a particular attention attracts available bit rate (ABR) one, intended for the ordinary data traffic (file transfer, e-mail delivery, web browsing, etc.), as it is the only category, which responds to the network feedback. Basically, two feedback mechanisms are available for this service type. The first uses a bit in the ordinary data cells, which is set by the nodes in the case of an overload. Once the destination detects a congestion indication (CI) bit marked in the received cells, it issues a command to the source to throttle the data emission rate. The command is sent in a special control unit – a resource management (RM) cell – served with priority by the nodes. The second mechanism involves periodic generation of RM cells by the sources, which travel interleaved with data cells to the destinations collecting the feedback information from the nodes, and are directed by the receivers back to their origin. The nodes incorporate specific, more accurate information about the congestion level (such as the current buffer occupancy), which is used by the transmitters to appropriately modify the cell transfer speed [19].

The standardization of ATM was mainly conducted by ATM Forum – a joint task force of international companies and institutions. To resolve congestion control issues, critical for proper network operation, a special traffic management group was started in 1993. The group elaborated a number of comparison criteria and evaluated various proposals with respect to scalability, efficiency, fairness, robustness, and implementation. An excellent survey of the most important results presented at the group meetings, together with short rationale for their potential success or rejection was given by Jain in [79]. The main congestion control concepts reported in that period are briefly discussed below.

Fast Resource Management [36], the scheme proposed by France Telecom, assumed that each source sends an RM cell requesting the necessary bandwidth before transmitting any data. If the demand cannot be fulfilled at a certain node along the data path, the RM cell is dropped. Once the timer (started when the RM cell is emitted) expires, the request is repeated in another RM cell. In the case of successful allocation, the RM cell is returned by the destination to the source, which can then deliver its data. This means that each transmission is preceded by at least one round-trip time (RTT) idle period. To eliminate this delay, the awaiting burst can be sent immediately after the RM request cell, at the risk of being discarded together with the RM cell when the expected resources are not granted. The described proposal was not accepted by ATM Forum due to significant latency in normal networking conditions and excessive loss during the congestion.

Another proposal, Backward Explicit Congestion Notification (BECN) [119–121], was based on the buffer occupancy monitoring by the network switches. Once the queue length in the switch buffer exceeds a threshold, RM cells are sent to all the sources contributing to the queue buildup. Upon the receipt of a BECN cell, the transmitter reduces its rate by half. If no BECN cells arrive at the source within a recovery period, the rate is doubled once every period until the peak value is reached. In order to achieve fairness among multiple virtual connections, the recovery period was made proportional to the current rate so that the smaller the rate, the shorter the period and faster recovery towards optimal transfer conditions. To cope with the system inertia resulting from the delay in the feedback loop, a filtering function was introduced into BECN mechanism at the nodes which prevents excessive BECN cell generation. The BECN proposal was considered unfair, as the sources receiving overload notification were not always the ones causing the congestion [133], and was rejected by ATM Forum. However, the idea of network nodes sending the explicit information about deteriorating traffic conditions to transmitters, based on buffer occupancy monitoring, proved applicable for other networks [4]. In particular, AQM combined with Explicit Congestion Notification (ECN) form a valuable supplement to the fundamental open-loop control provided within TCP end point specification (see Sect. 2.4 and Chap. 8).

Early Packet Discard [149], the solution presented by Sun Microsystems, exploited the fact that all the fragments of a data segment partitioned into cells must be received correctly at the destination to reconstruct the message. Therefore, it is convenient to drop the entire segment at a node, instead of selective, random discard of cells belonging to (in general) different segments. A big advantage of this approach is the short time to market, as it requires neither the standardization of the source-switch nor interswitch communication. It can also be implemented as a supplement to other control schemes and operate in the case of severe congestion when cells need to be dropped due to the buffer overflow. It was revealed, however, that Early Packet Discard could not guarantee fairness since the cells arriving at the fully occupied buffer were discarded even though they belonged to the VCs that had not contributed to the switch congestion [135].

Another approach not requiring internode standardization, delay-based rate control [94], was put forward by Fujitsu. It involves periodic generation of RM cells by the sources. The RM cells are sent back by destinations towards origin, and the congestion state estimation is performed via latency measurements. Each emitted RM cell contains a timestamp, which is used to calculate the RTT propagation delay upon the reception of the returning RM cell. Since no feedback from the network is required, the proposed algorithm can operate in heterogeneous networks (consisting of various technologies), similarly as another delay-based scheme developed by Jain [76]. Unfortunately, no precise details concerning the usage of latency estimation to react to changing networking conditions were specified following the concept presentation, and the proposal was abandoned [79].

Fair queuing with rate and buffer feedback [110] employed RM cells generated periodically by the sources to collect the feedback information from the network nodes on the data route of their VCs. Each node maintains a separate queue for each VC and schedules cell transmission in the order of the increasing service time. Two values are recorded in RM cells: the queue length and fair share of the available bandwidth. The queue length is updated by a node only if it exceeds the value written by other nodes on the forward data path. Bandwidth share in an RM cell, in turn, is modified only if it is smaller than that already allocated by the nodes on the forward path. Consequently, in the returning RM cell, the maximum queue length and the minimum bandwidth for a VC is recorded. The biggest disadvantage of this control strategy stems from the complexity in implementing per-VC queuing.

One of the most popular solutions to the congestion control problem in the ATM networks was called credit-based approach. The algorithm, elaborated by Kung and Chapman [103, 104], was based on the idea of hop-by-hop window control [58], where the nodes maintain a separate queue for each VC. A sender (source or switch), communicating over a link with the receiver (switch or destination), transmits only as many data cells as permitted by the recipient. The number of currently allowed cells – a credit – is determined by the receiver on the basis of the queue lengths of each of the active VCs. The credit is chosen large enough to obtain full bandwidth usage of the link at all times. This initial scheme, flow control virtual circuit (FCVC), had two important drawbacks: first, it did not provide any protection against losing credits; second, it required excessive buffer reservation for the controlled VCs. The first problem was solved by incorporating a credit resynchronization algorithm. It assumed periodic exchange of sent and received cell counts between a transmitter and its recipient and augmenting the credit by the number of lost cells. The second issue was solved by changing buffer allocation procedure. In the modified version of the control scheme, adaptive FCVC [102], the assigned buffer capacity was related to the number of VCs and their activity in using the credit. Highly active connections were granted a larger credit, and those less demanding were given a reduced fraction of the available resources. Although adaptive FCVC improved the buffer capacity management, the level of bandwidth utilization deteriorated due to the additional delay in allocating the entire capacity.

The second most favored solution, ultimately recommended for the ATM standardization, rate-based approach, was proposed in 1994 by Hluchyj [62]. Unlike credit-based approach, it assumed that a flow control mechanism should concentrate on regulating the transmission rate of the sources rather than directly imposing limits on the number of cells transferred across the links. Since rate-based approach allows the controllers to influence not only the amount of data injected into the network but also the way the data is emitted, it brings the potential of alleviating the traffic burstiness (which is typically obtained with traffic shapers, such as the leaky bucket algorithm [161]).

Among the initial concepts of flow control in ATM networks, these two approaches – credit- and rate-based – gained the most popularity and actually divided the research community of the involved scientists and telecommunication engineers into two opposing camps. Either faction supported only one of the schemes and was unwilling to seek compromises to the extent which precluded adaptation of integrated solutions (attempting to combine the benefits of both the credit- and rate-based control, e.g., [134, 165]). This led to the intensive debate in the traffic management group, which lasted for over a year and finally resulted in choosing the rate-based proposal as an official ATM Forum recommendation. In the course of multiple workshops, presentations, and discussions, numerous aspects were considered [63, 64, 77, 134]. The recognized distinguishing feature of the credit-based solution is the requirement of implementing per-VC queuing (to avoid deadlocks and unfairness) and making the buffer reservations even for inactive connections. This leads to excessive complexity of switches and was considered not scalable for larger number of flows. The rate-based solution, on the other hand, can be realized with common buffering, thus eliminating scalability concern.

However, the simplicity of switches usually implies relegating complexity to the end points, which in the credit-based approach could be significantly reduced (e.g., network interfaces may always send at the peak rate without cell scheduling and speed adjustments). Additionally, per-VC queuing creates possibility of isolation of misbehaving users, especially in the static credit-based scheme, which is difficult to obtain with the common-buffering solution. Credit-based approach can guarantee loss elimination irrespective of the number of connections, traffic patterns, buffer sizes, node number, link bandwidths, or propagation and queuing delays, as the queue lengths can never increase beyond the granted permits. This generally cannot be achieved with the rate-based schemes since in the case of serious overload and long feedback propagation latency, the queues can exceed the buffer limitations resulting in cell loss. The argument in defense of the rate-based control was the inherent necessity of dealing with error-originated losses. Consequently, as a certain number of cells will have to be retransmitted even though no congestion has been experienced in the network, a small degree of overload-originated cell discards can be tolerated. The rate-based concept allows for more flexibility in switches in deciding how to allocate resources. Communicating nodes can use different mechanisms yet interoperating in the same heterogeneous network. In contrast, the credit-based scheme required all the switches to use per-VC queuing with round-robin service. The benefit of credit-based approach, in turn, was the ability of rapid, usually immediate capture of the available bandwidth, thus eliminating the ramp-up time (the time necessary to adjust the transfer rate to the currently optimal operating point). This was particularly true for the static version of the credit-based algorithm. The initial rate-based schemes and the adaptive credit-based solution usually required several RTT delays to converge to the modified networking conditions.

The presented summary gives an overview of the most important issues raised during the debate. According to Jain [79], the single biggest objection to credit-based approach was the necessity of applying per-VC queuing, which was regarded intricate for switch implementation and not scalable for networks supporting large number of high-speed connections. Rate-based approach, in turn, seemed to offer more design flexibility and better evolutionary prospects and ultimately won the competition. It was included in the official ATM specification as a standard of feedback information interchange and a basis for future development of flow control techniques [17]. In the next section, we will concentrate on various solutions to the congestion control problem built on the rate-based concept.

2.3 Initial Rate-Based Proposals

The first algorithms in connection-oriented networks, which can be classified as rate-based ones, were in fact appropriately modified binary feedback schemes previously designed for connectionless networks, for instance, DECbit algorithm [85]. The binary feedback assumes that a single bit is sufficient to adjust the flow rate and to control the congestion. Network nodes monitor the queue lengths and set explicit forward congestion indication (EFCI) bit in the traveling cells when the congestion is detected. Destination periodically checks for the presence of the EFCI bit in the received cells and sends an RM cell to the source, which uses an additive increase multiplicative decrease algorithm to adapt its rate. However, if severe congestion is experienced on the backward path, RM cells are lost and source keeps increasing its rate, actually aggravating the network overload. The problem was solved by applying a positive feedback approach. In the improved version, called proportional rate control algorithm (PRCA), sources mark EFCI bit in every cell except the *n*th one. If a destination receives a cell without EFCI bit set, it sends an RM cell to the source requesting the rate increase. The introduced amendment did not help with another important issue with PRCA, which is the possibility of unfair treatment of long-distance flows. In PRCA, the cells traversing more switches exhibit higher probability of having EFCI bit set than those belonging to local connections. Possible solution to this problem involves selective feedback [136], which assumes that the congestion indication should be sent only to the sources consuming more resources than the calculated max-min fair share, or intelligent marking [25], where the current flow rate of a particular VC and its influence on the congestion is used to decide about setting the EFCI bit.

The binary feedback schemes did not allow for exploiting the full potential of connection-oriented networks. They were considered too slow for high-speed traffic (requiring multiple cycles to reach optimal operating conditions) [41] and did not offer enough flexibility for the algorithm designers [79]. Faster convergence, simplified policing, and improved robustness could be achieved with explicit rate indication. This argument led to the development of Massachusetts Institute of Technology (MIT) scheme [40]. In the MIT algorithm, each source sends an RM cell every N data cells containing its current and desired rates. The nodes monitor all the flows and compute fair share of the rate in an iterative procedure. If the expected rate is bigger than the calculated one, it is replaced by the computed share, and a reduced bit is set in the control cell. The destination returns the received RM cell to the source, which adjusts its rate to the value assigned by the nodes. If the reduced bit is present, the source sets the desired rate equal to the current transmission speed. Otherwise, it is allowed to request a higher rate. It was demonstrated that the proposed solution conforms to the max-min fairness criteria [40].

The drawback of the MIT scheme was computationally intensive fair share calculation. A simplified procedure was proposed in enhanced PRCA (EPRCA) [145, 146], an improved version of PRCA combining the features of binary and explicit rate control mechanisms. The sources send RM cells every N data cells with their current and expected rates indicated. The nodes compute fair share as a fraction of exponential weighted average of mean allowed cell rate and appropriately adjust the explicit rate (ER) field in RM cells. Additionally, destinations set the congestion indication bit in RM cells if EFCI bit is set in the previously received data cell. The sources reduce their rate after every cell transmitted and increase the transfer speed (if allowed) upon the reception of an RM cell, taking into account the assigned rate, congestion indication bit, and increase factor. In the initial proposal, the queue length threshold was used as the congestion indicator. This was shown to be unfair as the sources joining the active pool at a later stage used to get smaller throughput than those starting earlier. The problem was alleviated by changing the congestion indicator from the queue length threshold to the queue growth rate [155]. The obvious advantage of EPRCA was its backward compatibility with PRCA, which allowed for efficient coexistence of older switches and smooth software and hardware upgrade to modern versions [122, 146].

The fairness issues with EPRCA stimulated further search for improvements of rate-based controllers for connection-oriented networks. The scheme proposed in [45], dynamic max rate control algorithm (DMRCA), employs a function of both the queue length in the node and the maximum rate of all connections to signal the congestion. The use of the maximum rate of all the active connections instead of the mean rate applied by EPRCA eliminates fairness shortcomings occurring when the mean is significantly different from the fair share. Such approach allows for rapid rate increase, which temporarily remains above the fair level and converges to the fair share in steady state. The problem with using the maximum rate, however, is large oscillations in transient period, which may render the system unstable. Therefore, a practical implementation of DMRCA requires rate smoothing typically realized by filtering out the short-term variations. Additionally, to solve the fairness issues arising due to excessive rate increase beyond the fair allocation and slow convergence to steady state, a reduction factor can be applied, which is a function of the degree of congestion of the switch measured with queue thresholds. However, to ensure a proper operation of the thresholding procedure, several parameters need to be set, which were shown to be sensitive to RTT and feedback delay [90].

The researchers from Ohio State University (OSU) elaborated a series of congestion avoidance algorithms to address the problem of fair yet efficient flow control for the ATM networks [80, 81]. In the original OSU strategy [80], the rate allocation was based on the load measurement performed by the network nodes. Each node calculates the incoming rate (averaged over a certain interval) and compares it with a target value, which is typically set as 85–95% of the link bandwidth. The resulting fraction (the input rate divided by the target one) constitutes a quantitative measure of congestion and is used for rate adjustment. In the case of severe congestion, or small bandwidth utilization, each source is requested to divide its rate by the load factor. In turn, when the load factor is close to one, the principle of selective feedback is applied. The node computes the fair share (the target rate divided by the number of active VCs) and allocates different rate values for overloading and underloading flows. This solution was demonstrated to be fair [80]. The mechanism used by the OSU strategy, being a congestion avoidance scheme (the target rate kept close to but below the capacity), allowed for high throughput and low queuing delay in the system. Additionally, as compared with EPRCA, it achieved faster convergence to steady state and was easier to tune having fewer parameters.

The problem of the first OSU strategy was large control overhead related to the periodic RM cell generation at fixed intervals by all the sources. To address this scalability issue, the method of feedback information distribution was adapted from

EPRCA. In the improved algorithm, each transmitter sends an RM cell every N data cells and not every T seconds, thus making the overhead independent of the number of served connections. In order to maintain the advantages of the original strategy, the rate allocation procedure of this new, count-based scheme [81] was modified to promote faster convergence to the fair share. Nevertheless, in certain configurations, fast convergence could not be guaranteed, especially if large errors appeared in the traffic measurements (e.g., in estimating the number of active VCs) [82].

Further improvements to the OSU scheme resulted in the development of congestion avoidance using proportional control (CAPC) algorithm [26] and its enhanced version CAPC2 [27]. This approach, proposed by A. Barnhart, supplements the OSU load factor measurements and congestion avoidance (keeping the utilization target slightly below the capacity) with queue threshold monitoring. Whenever the queue length exceeds the threshold, the CI bit is set in RM cells, which prevents the sources from increasing their transmission speed. As a result, the queue dynamics improves and the buffers can be depleted more rapidly. However, the truly distinguishing and highly desired feature of CAPC was its oscillation-free steady-state performance.

Interesting approach to congestion avoidance with rate-based flow control, sharing some similarity with CAPC, was proposed by Afek et al. [1]. The key idea of their scheme, called phantom, is to bind the rate of sessions that share a link by the amount of unused bandwidth as though there would be an additional imaginary session – a phantom – on that link. The bandwidth is fairly distributed among all the sessions, including the phantom, and the residual unused capacity can be used to accommodate new connections without the queue buildup. The disadvantage of this scheme lies in the unused bandwidth measurement, which is difficult and prone to errors. Moreover, in order to eliminate misestimation due to instantaneous residual capacity variations, filtering needs to be applied. This, however, introduces extra parameters and requires additional tuning.

The authors of the OSU scheme, encouraged by the success of their initial proposal, developed a newer and more efficient version of the congestion control algorithm called explicit rate indication for congestion avoidance (ERICA) [83], further enhanced to ERICA+ [84]. It has become the subject of substantial research and numerous publications [1, 90, 99, 111, 163] and was incorporated into the ATM traffic management official standard [18]. Its refined and consolidated description together with performance analysis is given in [91].

ERICA is based on the load and capacity measurements performed in consecutive equal-sized time slots called “switch averaging intervals.” During each averaging interval, the network node estimates the available capacity (which varies according to the intensity of high-priority traffic) and total input rate. These quantities are used to determine the load factor calculated as the total input rate divided by a fraction of the available capacity (further referred to as the target capacity). The fraction depends on the implementation and is usually set as 0.9 or 0.95. Its purpose is to keep the resource utilization close to one while preventing excessive queue length growth and resulting delay increase. The algorithm computes two values of transmission rate to be assigned to the sources. The first one

– load-based – is determined by dividing the current rate (read from the arriving RM cell or estimated through measurements) by the load factor. Consequently, the rate is increased if the load factor is bigger than one and throttled down otherwise. The second value is the maximum of the first rate and a fair share, computed as the target capacity (fraction of the available capacity) divided by the number of active connections. Assigning the maximum of the fair and load-based rate values favors high link utilization over fairness and allows an unconstrained source to proceed towards its max-min rate. This step is one of the key innovations of the ERICA scheme because it improves fairness even under overload conditions. Finally, the determined rate is recorded in the ER field of the management cell, if it is smaller than the value assigned by other nodes on the forward path (obviously, the rate of already bottlenecked connection should not be increased).

The ERICA algorithm operates efficiently and is straightforward in implementation. It quickly adapts to the changing networking conditions and usually achieves fair resource allocation (although in certain circumstances, it may fail to guarantee the max-min criteria [90]). One of the biggest problems with ERICA is its dependency on the measurement of metrics, in particular the estimation of the overload factor, available capacity, and number of active sources. If the measurement is significantly distorted by errors and the target utilization is set to very high values, ERICA may diverge, i.e., the queues may become unbounded and the capacity allocated to drain the queues may be insufficient. The solution under such cases is to set the target utilization to a smaller value allowing more bandwidth to empty the queues. However then, the steady-state utilization downgrades since it heavily relies on the target utilization parameter. To address this issue, an enhancement was introduced in ERICA+, which replaces the fixed utilization parameter (fraction of the available capacity) with a function of the queue length and delay. Allowing the target capacity to vary according to the queue length and latency fluctuations gives ERICA+ flexibility needed to obtain high throughput and limited delay.

The control schemes discussed so far employ mainly computer-algorithmic solutions and empirical observations to achieve the desired level of operational efficiency of the controlled network. Such approach, although successful in a majority of typical situations, poses severe difficulties in conducting formal analysis and assessing the full potential and limitations of the developed algorithms. Further in this chapter, we will focus on various proposals addressing the problem of congestion in data transmission networks from a theoretic, analytical perspective.

2.4 Evolution of Congestion Control Techniques

Earlier control schemes favored heuristic, ad hoc solutions. Noticeably, such intuitive design with the algorithm parameters tuned through simulation or experimental analysis have demonstrated astonishing performance in a variety of real-case scenarios. However, the empirical, ad hoc approach usually leads to severe nonlinearities and makes the systematic analysis of the proposed solutions cumbersome, if not impossible. This, in turn, causes difficulties in identifying elements truly responsible for their good performance and, more importantly, obfuscates the investigation of improper or erratic behavior in the case of failure [127]. The lack of systematic design methodology (e.g., involving control engineering tools) slows down the research on new algorithms and obstructs or even inhibits the evolution of the existing ones. Consequently, many researchers began seeking the answer to the congestion control problem in communication networks in structured modeling and application of various mechanisms that were proved effective in other fields, such as control engineering, stochastic analysis, game theory, or neural networks.

From the point of view of data traffic regulation and resource management, a data transmission network can be perceived as a feedback system. Therefore, it seems to be a natural choice to apply the concepts of control theory to the design of congestion controllers for network traffic control. One of the first successful and highly influential approaches to flow regulation using the ideas from classical control theory was proposed in 1993 by Benmohamed and Meerkov [30]. They provided a detailed mathematical description of the model of a packet-switched network, in which the sources respond to the feedback from the nodes. In the model, the feedback information was considered to be delivered with priority over the users' data and with constant delay. In the initial scheme [30], it was assumed that a single link constitutes the bottleneck for the set of controlled connections. In the fundamental derivation, the paths established for the data transfer remained unchanged during the transmission (although a comment about adaptability to routing modifications was included), and stochastic random input traffic can be approximated with the deterministic fluid flow (validity of this simplification was discussed). The obtained control law was based on the idea of standard proportional-derivative (PD) control with higher-order derivative terms incorporated to account for the propagation delay. Parameters of the proposed controller were chosen on the basis of the closed-loop pole placement so that the local stability of the nominal model could be achieved. However, due to the nonlinearities of the complete network model, the authors failed to solve the global stability problem. To address the issue of changing networking conditions (caused by time-varying input traffic), two approaches were presented: the static (also referred to as the robust) and the adaptive one. In the static scheme, the controller parameters are established prior to transmission considering the worst-case uncertainty scenario, while in the adaptive one, parameters are adjusted dynamically according to the varying traffic intensity and number of connections. Both solutions were evaluated analytically and compared in simulations using the queue length overshoot, degree of oscillations, and settling time as metrics. In majority of circumstances, the adaptive scheme outperforms the static one despite increased computational complexity. However, neither strategy can avoid oscillations and overshoots, which hinders efficient buffer allocation policy. The presented idea of using PD control to combat the congestion in a network with a single congested node was later generalized by the same authors to the case of multiple bottlenecks [31]. Still though, the biggest drawback of the proposed strategy remains in its substantial intricacy of rate calculations.

In order to mitigate the complexity of calculations in the PD scheme [30], which mainly originates from determining the controller gains each time the number of VCs changes, Kolarov and Ramamurthy [100] proposed to use a dual controller consisting of two simpler PD regulators (whose parameters are computed off-line) instead of a single complicated one. The first is a low-gain PD controller (LGC) which is designed for the case when the number of bottlenecked sources at an output port is very large and is used under steady-state traffic conditions (it stabilizes the system for small deviations from the equilibrium point). Since the slow rise time of LGC may result in poor utilization of the output port during transient periods, a second controller, called high gain PD controller (HGC), is applied. HGC brings the operating point quickly to equilibrium, where the control action is taken over by LGC. As HGC is designed for a smaller number of bottlenecked VCs (with fewer number of filter taps), it does not ensure stability of the closed-loop system. However, it is used only for a short period of time before switching to LGC. The decision to switch between HGC and LGC is taken by a filter, which compares the link utilization with a preset threshold. Finally, to handle the problem of queue buildup (and cell losses) when a large overload occurs, the third mechanism is used, called Initial Recovery Rate Selector (IRRS), which quickly brings the rate to the normal operating condition. The IRRS action is triggered if the queue length exceeds a predefined threshold or if it grows at an excessive rate. Although providing good steady-state and transient properties with low computational overhead, the use of the dual PD scheme requires careful tuning of numerous parameters, in particular the filter coefficients and thresholds, for proper switching between LGC, HGC, and IRRS controllers. The parameter adjustment can be time-consuming and prone to errors.

In literature, we can find other examples of applying the concept of PD control to regulate the flow of data in communication networks [108, 148] as well as proportional (P) [46, 147], integral (I) [21], proportional-integral (PI) [61, 89], classical proportional-integral-derivative (PID) [33], or fuzzy PID [159]. Apart from oscillations and overshoots, the characteristic limitation of the majority of those schemes when applied to feedback systems with delay, as discussed in [114, 147], is the necessity of throttling the controller dynamics (mainly reducing the closed-loop gain) to maintain stability.

As it has already been discussed, the need for reducing the computational overhead resulting from hardware limitations in the early protocol development (scarce processing power and slow memory) turned the attention of some researchers towards the binary feedback control schemes. However, those schemes, although simple in implementation, proved inefficient in the network resource management (they used to excite oscillations in the traffic intensity and required large buffers to accommodate the resulting burstiness of packet arrival). A number of research efforts were undertaken which tried to explain the specific behavior of binary feedback algorithms using systematic analytical methods.

Kawahara et al. [95] considered a two-stage queuing network where the cells from multiple sources (the first queuing stage) appear with delay at a node constituting the second queuing stage. The burstiness of cell arrival at the sources was modeled as a stochastic process (two-state Markov-modulated Bernoulli process). The sources simply stop/start transmission according to the CI bit in the control unit received from the node or additionally monitor the local queue length and push out backlogged cell from the full buffer when a new cell arrives within a time slot. The latter mechanism allows for cell loss reduction at the expense of increased waiting time at the node. The analysis of system properties was limited to the steady-state case only due to significant computational complexity of the dynamical case.

Bonomi et al. [35] faced a similar problem of excessive derivation complexity in the analytical design of binary feedback controllers. To address this issue, they proposed a simplified fluid approximation of the network model (composed of a set of first-order delay-differential equations) as the basis for the mathematical calculations and provided extensive simulations for a more realistic scenario. The authors discussed various design trade-offs and showed how an appropriate use of rate damping can improve stability and fairness without significant downgrade in responsiveness during the transient phase.

The biggest disadvantage, obstructing the systematic analysis of flow control schemes employing binary feedback, is the existence of significant nonlinearities within their operating regions. These algorithms tend to demonstrate serious problems of stability, exhibit oscillatory dynamics, and require large amount of buffer in order to avoid cell loss. Rohrs et al. [148] proposed to modify the interpretation of the CI bits and in this way linearize the system so that the standard techniques of linear control theory could be applied to binary feedback framework. The controller proposed in [148] estimates the level of congestion with parameter p, determined using the node buffer occupancy. The CI bit in RM cells is marked with probability p and left unset with probability $1-p$. The source deduces the value of the p parameter by calculating the fraction of RM cells with the CI bit marked to the total number of RM cells received in certain time interval. Once the parameter is estimated, the source can smoothly adjust its transfer rate. The proposed method of the CI bit interpretation creates the effect of multivalued feedback and allows for significant reduction of the degree of oscillations. Moreover, it let the authors employ the standard frequency domain techniques (Bode plot analysis) to tune the algorithm performance and satisfy various design requirements. A similar analysis by means of the classical linear control tools was later conducted for the network supporting explicit rate multivalued feedback [147].

The exposed problems of binary control, supported by strict analytical argumentation, ultimately directed the effort of research community towards the domain of multibit feedback schemes. Consequently, in the remainder of this section, we will concentrate on explicit rate solutions that employ multivalued feedback information to control the flow of data in communication networks. We will briefly return to binary and implicit feedback systems in the final paragraph of this section while discussing the congestion control concepts in TCP/IP-based and general-type networks.

Izmailov [71] considered a single connection serviced with constant capacity by a distant network node. The transmission rate of the connection is controlled by a linear access regulator whose output signal is generated according to the several states of the node buffer measured at different time instants. The asymptotic stability, nonoscillatory system behavior, and locally optimal rate of convergence were proved. However, the transient properties and responsiveness to the changing networking conditions were left unexplored, which limits the assessment of applicability of the proposed scheme in real, highly dynamical telecommunication networks. In his next paper [72], Izmailov extended the analytical investigation of the designed control algorithm to the case of multiple connections.

To cope with difficulties of the analysis of multisource traffic scenario, Zhao et al. [179] proposed to decouple different control loops and in this way reduce the controller design to a set of single-input single-output systems. The design task was formulated as a standard disturbance rejection problem where the available bandwidth acts as an external perturbation in the system. The source rate is adapted to low-frequency variation of the available bandwidth and H_2 optimal control is applied to design a controller that minimizes the difference between the source input rate and the available bandwidth. The principal disadvantage of the developed scheme is that the design procedure and the controller performance depend on the characteristics of the interacting high-priority traffic and on the measurements of the available bandwidth, which is cumbersome to be obtained in practice (as discussed for instance in [111]). Moreover, the decoupling of mutli-input single-output system based on the assumption of uniform disturbance and capacity distribution, and the subsequent optimization performed for each flow separately, may result in an oversimplistic design for a multisource network serving flows with different propagation delays [68]. Direct generalization of the result obtained for a single flow to the system with multiple connections characterized by disparate feedback delays may lead to undesirable oscillations in traffic intensity, which adversely affect the system dynamical properties.

Chong et al. [46, 47] proposed and thoroughly studied the performance of a simple queue length-based flow control algorithm with a dynamic queue threshold adjustment. The authors demonstrated that with properly chosen gains, the system is asymptotically stable, even for the case of multiple VCs with long and diverse propagation delays, and in steady state, the max-min fairness is achieved. Nevertheless, the transient analysis was performed for a homogeneous network (equal delays), and global stability was investigated via simulations only. The dynamic queue threshold adjustment in the case of time-varying available bandwidth and changes in the number of connections was shown to effectively prevent the closed-loop system from going to undesirable equilibrium point. In addition, the authors demonstrate that the adaptive queue thresholding allows for decreasing the sensitivity of the queue length to the bandwidth and VC number fluctuations.

The frequent variations of networking conditions in communication systems suggest the use of adaptive techniques for data flow regulation [3, 10, 34, 98]. One of the first algorithms employing the concepts of adaptive control was proposed by Keshav [98] in 1990. He assumed that a node (server) distributes available bandwidth equally among the set of active connections passing through one of the output ports. A source deduces its current service rate by sending two back-to-back connected packets and taking the inverse of the time spacing of the received acknowledgments. The noisy estimate of the current rate and the trace of the queue length are used to predict the future service rate (modeled as a random walk with the step being random variable). Using these values, the proposed control law drives the state (queue length) to a predefined set point. Following the discussion of the limitations of the traditional Kalman filter solution, a novel state estimation scheme based on fuzzy logic was developed. Another adaptive algorithm for congestion control in communication networks, employing a self-tuning regulator, was presented by Bolot [34] in 1992. He described the queue dynamics of a single bottleneck node with auto-regressive moving average with exogenous input (ARMAX) model and applied a predictive controller with least-squares model parameter estimator to regulate the source input rate. As investigated by simulations, the obtained self-tuning regulator provides high throughput and low loss probability over a wide range of delays and overload conditions. In the next paper [10], written jointly with Altman and Baccelli, a stochastic discrete-time approach to the design of flow controllers in data transmission networks was presented. The proposed regulator calculates transmission rates taking exponentially averaged estimate of the available bandwidth and queue size. The time-varying bandwidth is modeled as a truncated auto-regressive moving average (ARMA) process, in which the truncation ensures that the bandwidth cannot exceed certain bounds, thus eliminating the drawback of the random-walk model applied by Keshav [98]. The authors demonstrate that a pure rate-matching algorithm may lead to unacceptably long queues. Therefore, in order to guarantee stability, the combined approach of the bandwidth and queue length estimations should be applied. Although, as discussed in [3], the stability of the pure rate-matching scheme can also be satisfied if only a fraction of the available service rate is used as the utilization target. In consequence, excessive data accumulation in the buffers can be prevented.

In the further research, Altman et al. [7] and Pan et al. [123] exploited the ARMA model introduced in [10] to design several H^{∞}-norm-based controllers for a single connection bottlenecked at a network node. The work of Altman, Basar, and Srikant [8, 9, 11, 12] on multisource networks, in turn, led to the application of elements of game theory to the congestion control problem. In that approach, the flow control process can be perceived as a game, where the sources sharing a common bottleneck are the competing players. In the noncooperative game, the players try to achieve the best possible result (e.g., to capture the available bandwidth), which influences the performance of other players and creates conflicts. The objective of the control algorithm is to drive the system to a state, the users have no incentive to deviate from (they gain nothing by changing their strategy according to some performance measure). Such state is referred to as the Nash equilibrium [117]. In the cooperative game approach [9, 11, 12], on the other hand, the users sharing the bottleneck link form a team, which attempts to optimize some common objective (expressed with appropriate cost functional) instead of aggressively competing with each other. The popularity gained by the concept of game theory applied to congestion control, and performance optimization of distributed systems has led in recent years to an intensive research and a number of valuable publications, mainly in the field of traffic regulation in Internet-style networks [6, 14, 65, 150], and optimization of distributed computing systems [92].

Seeking the optimal solution to the congestion control problem is obstructed by the presence of action delay, which is the time from the moment the control information is sent to the source, until the action is taken by it and until subsequently that action affects the state of the node that issued the command. In order to avoid extensive numerical computations necessary to solve high-dimensional discrete-time algebraic Riccati equation in time-delayed stochastic systems, several authors [13, 70] proposed the use of certainty-equivalence principle. This principle adopted for communication networks allows for separation of the controller design into two stages. In the first stage, the exact optimal solution is obtained for a system with no delay. Afterwards, the delays are incorporated into controller through estimators predicting the future values of the queue length and bandwidth. The resultant suboptimal controller achieves similar performance as the exact, optimal one (determined numerically). However, both the optimal and suboptimal solutions presented in [13, 70] are derived under the assumption of perfectly known and static propagation delays. If the latency varies with time, an adaptive controller can be applied. However, the one proposed in [70], although demonstrates robustness to delay fluctuations, is developed for an idealized case of constant available bandwidth. The derivation of optimal solution in the system with delay can also be simplified with the use of instantaneous cost index [38]. The method advocated in [38], while avoiding the complex forecast problems (which require *a priori* knowledge of appropriate models of the future system state) encountered in [13, 70], allows for minimization of the amount of discarded data under the constraints imposed by the QoS contract of high-priority traffic. We show later in this monograph (Sects. 5.1.2 and 6.1.2) how the optimal control problem with quadratic performance index can be solved analytically with explicit consideration of the effects related to delay. A closed-form solution is obtained without recurring to certainty-equivalence principle.

The majority of systematic control solutions discussed so far assume that feedback delay is constant. Obviously, this assumption facilitates formal analysis but is seldom well justified from the perspective of the actual application in data transmission networks. The important problem of latency variation in a delayed feedback system has been addressed in a number of papers [33, 101, 124, 131, 153]. The authors of [101] modeled the delay fluctuations as well as the source activity periods and instances of rate calculation and feedback information delivery with random variables. They studied the influence of the described variations on the system performance with parsimonious (binary) and multivalued feedback mechanisms and formulated a number of observations, some being counterintuitive, on the relative importance of the investigated factors. First, it was shown that the time scales (the source activity period in relation to the propagation delay and frequency of bandwidth changes) dominate both the steady-state and transient performance of the system. According to the authors of [101], the longer the scale (smaller frequency of fluctuations) of a particular network variable, the bigger is its influence on the system performance. Secondly, it was demonstrated that a two-level controlled feedback system (with two queue thresholds), due to the extension of the overload and underload periods, is inferior to a single set-point solution in terms of the obtained throughput. In the presence of underlying high-priority traffic, performance comparison of the system using binary feedback with queue thresholding vs. the system using explicit rate notification shows that better throughput can be achieved in the latter one. This is especially true when the underlying traffic has slowly varying time scales. The most important observation is that asynchronism in the feedback information delivery (arising due to delay variations or generation of feedback carriers at irregular time instants, e.g., once every *N* data packets) actually results in better system performance and helps achieve stability, although may potentially lead to some unfairness. While the equivalent deterministic system may never converge (or converge very slowly to the desired equilibrium), the stochastic system surprisingly converges at a faster rate. However, it should be stressed that this counterintuitive remark about variable feedback delays improving stability of the oscillating system is formulated based on stochastic analysis of the plant and control signals. Consequently, the results of evaluation of any designed feedback scheme depend on the selected framework – deterministic or stochastic – and may be contradictory to each other.

Sichitiu et al. [153] developed a model for a rate-based congestion control system, considering rapidly changing buffer levels, which accounts for both the time-varying delays between the congested node and the data sources and the mismatch between the time-varying RM cell arrival period and the fixed controller cycle time. The modeling also includes the effects of buffer and rate saturation without the simplifying assumption of linearization around the equilibrium point. The resulting time-varying linear feedback system (nonlinearities are modeled through time-varying sector gains) is analyzed with regard to its stability using the theory for uncertain time-varying systems. It is shown that no stable equilibrium point exists if the delays in the forward path vary with time. Intuitively, even if the available bandwidth is constant, changes in the delay cause fluctuations in the output rate, which feed the queue integrator disturbing any previously established equilibrium level. Consequently, the authors of [153] claim that, under the time-varying delay conditions on the forward path, set-point control of the congested buffer cannot provide the desired queue length. Despite careful and detailed description of a multitude of networking phenomena, no formal controller design was performed, and the presented model should be regarded as a framework for future algorithm development. The discussed properties of time-varying feedback system are illustrated with an example of a simple proportional controller. A later work of Sichitiu and Bauer [152] on the stability of time-varying plants led to a proof of an important feature of congestion control systems with linear controllers, namely, the stability of such systems with a single source is equivalent to the one with multiple transmitters. Consequently, if the stability can be shown for a system with one source, the stability of the system with multiple sources follows automatically, which may simplify complex analysis in the multisource networks.

The work on effective control methods for feedback systems with time-varying delays was continued by Quet et al. [131], Blanchini et al. [33], Ünal et al. [168], and Pietrabissa et al. [124]. Quet et al. [131] elaborated a stable controller regulating the traffic in multiple connections with uncertain time-varying propagation delays and time-varying node service rate. The controller, obtained through the minimization of appropriate H^{∞} norm, satisfies a weighted fairness condition and guarantees asymptotic stability of the queue length at the bottleneck node. However, the time-varying forward delay induces steady-state oscillations, which cannot be avoided unless some information about the forward delay uncertainty is available to the controller. Moreover, the controller performance deteriorates as the real delay diminishes (in relation to the uncertainty bound), which is a feature that may inflict undesirable consequences on the control process and transmission consistency, as commented in [33]. An improved version of the H^{∞}-norm-based controller has been later reported in [168]. Blanchini et al. [33], on the other hand, proposed a classical control design based on PID controller to provide a stable solution to the network traffic regulation problem in the presence of time-varying delays. As pointed out by the authors, the more sophisticated controllers proposed earlier in literature, in particular the optimal ones, achieve better performance as long as the delays can be determined accurately and remain constant. The fragility of the optimal controllers manifests itself even in the simplest case of a single link, where any variation of the latency from the nominal one may render the closed-loop system unstable. The PID controller proposed in [33] remains stable even though the delays are determined imprecisely, vary with time, and the sources do not always follow the feedback signal. The stability analysis for a single connection was proved analytically using the Nyquist criterion and extended to multisource case through the equivalence property, which states that the mutlisource scenario in the proposed framework can be reduced to the single-source one, in which the maximum admissible delay is equal to that of the source at the maximum distance. The theoretical investigation was supplemented with simulation examples, comparing the elaborated PID solution with ERICA+ algorithm. The results show that the theoretically derived controller can achieve, in some respect, superior performance to that of the performance-oriented heuristic algorithms.

In paper [124], an adaptive controller effectively combining the benefits of control theoretic and fuzzy-logic approach was also proposed to address the issue of uncertain and time-varying delays and source saturation limits. The basic controller developed in [124] drives the queue length to a reference value, while the fuzzy element adjusts the adaptive multiplicative gain to compensate for the transmitter saturation. The designed regulator was compared with the H^{∞} scheme [131] and, according to the presented simulation results, demonstrates improved performance. It exhibits smaller degree of overshoots and oscillations and thanks to the incorporated fuzzy logic correctly reacts to the problem of nonpersistent data sources (even in the situation when all the sources are saturated and send the data at a rate lower than the assigned one). Fuzzy logic has also been exploited by Ren et al. [139] for the flow regulation problem in an ATM/ABR network with multiple uncertain time delays. In addition to good dynamical properties, the fuzzy immune-PID controller presented in [139] can handle a number of inopportune networking phenomena, for example, saturation nonlinearities. An extended version of the results reported in [139] has been recently published in [141], where the sufficient condition for the closed-loop system stability was presented and weighted fairness issues were discussed in detail. For a similar network model, the authors of [139], together with X. Zheng, proposed a controller based on integral sliding mode (ISM) [167]. This ISM controller [140], while maintaining the weighted fairness, demonstrates better dynamical performance and improved stability than the one described in [139].

In fact, the concept of applying fuzzy logic to traffic management, as well as other methods of artificial intelligence, such as neural networks or genetic algorithms, is not a new one (an excellent review of the initial as well as more advanced proposals is given in [49] and [125]). The lack of appropriate models in the early congestion control protocol development, on one hand simple enough to be analytically tractable, on the other hand retaining enough complexity to afford attractive performance properties, incited the search for alternative design paths. The systems based on fuzzy logic make decisions and perform calculations on imprecise quantities and, hence, are particularly useful in the situations in which the exact mathematical models are impractical or unavailable. In a sense, the fuzzy-logic-based systems emulate the decisions taken by an expert human operator. The schemes based on neural networks (NN), on the other hand, exploit the NN ability of capturing complex nonlinear relationships and predicting future system behavior from the acquired patterns [49]. Keshav [98] used fuzzy variables to construct a state estimator, which gracefully responds to changes in system behavior. Cheng and Chang [43] designed a traffic controller based on fuzzy rules, which simultaneously performs congestion control and connection admission functions, yielding improved performance over the traditional double-threshold queue length control. The optimal parameters of the membership functions and the control rules were determined from a genetic algorithm search. Pitsillides et al. [126] defined a set of linguistic rules for fuzzy ABR congestion control using the queue length and the queue length changes. The proposed scheme was compared with EPRCA and concluded to be superior in both maximizing the network utilization and achieving faster transient response. Tarraf et al. [162] discussed the application of NNs to solve a number of traffic management-related issues in ATM networks. According to the authors of [162], the learning capabilities of NNs can be effectively employed for predicting the forthcoming traffic intensity, connection admission control, flow policing, and congestion control. Jagannathan and Talluri [75] modeled the buffer dynamics at a network node as an unknown nonlinear discrete-time system and used NN-based controller with adjustable weights to predict the explicit values of the source transmission rates and to prevent the node overload. The proposed method of weights tuning guarantees stability of the closed-loop system and the desired QoS in the presence of unpredictable and statistically fluctuating network traffic with bounded uncertainty. Broad prospects of application of artificial intelligence elements in controlling complex telecommunication systems come at a price. Probably, the biggest drawback of those schemes as related to highly dynamical networks is the time lag of the learning process, which can be unacceptably long if good estimation accuracy of the model and its inferred behavior is to be obtained. Another disadvantage of artificial intelligence methods is the difficulty to execute sound stability and performance analysis using formal techniques.

Since the feedback information used for rate adjustment is typically available at discrete-time instants only, it seems reasonable to partition the time scale of the flow control process into discrete intervals reflecting the spacing between subsequent feedback information carrier arrivals and rate updates. The discrete nature of delayed feedback distribution, in turn, promotes the use of popular discrete control tools, such as digital filters, in system modeling and controller design. Laberteaux et al. [106] modeled the communication network as a finite impulse response (FIR) filter with unknown coefficients and proposed an adaptive control scheme based on the inverse FIR filter to efficiently regulate the data traffic. Several enhancements to the principal strategy allow for reducing the convergence time and improving the queue length management as compared with other schemes previously described in the literature. Tan et al. [160] designed a class of controllers based on recursive digital filter, whose parameters are selected so that fairness and the desired dynamical properties can be obtained. The closed-loop stability of the controlled system was analyzed using Schur-Cohn criterion and strictly proved. Aweya et al. [21] discussed in a tutorial fashion the analogy between the feedback control problem in communication networks and the classical discrete-time regulator problem employing digital filters. They proposed an integral control law, which allocates the current transmission speed for multiple sources using the comparison between the target and measured link utilization. The feedback delay was identified as a major cause of potential system instability and, hence, should be explicitly considered in the controller parameter selection. The authors of [21] propose a method of stability analysis using Routh-Hurwitz criterion. First, they determine the characteristic polynomial of the discrete-time closed-loop transfer function. Afterwards, they perform bilinear transformation and apply Routh-Hurwitz test on the coefficients of the modified polynomial. The resultant approximate relation between the controller parameters and the network latency allows for estimating the biggest admissible gain for a stable system. As discussed and verified through extensive simulations [21, 22], the proposed conservative approximation guarantees stability without excessive degradation of the transient properties. Moreover, as pointed out by the authors, in real systems, usually more restrictive constraints are applied than those obtained from the theoretical derivations (in practical implementation, it is not advisable to set the gain so that the system operates at the margin of stability, e.g., due to unmodeled uncertainties affecting the control process). In a later publication [20], Aweya et al. calculate exact stability limit for the developed control scheme, which occurs to be nearly the same as the approximate one (the error does not exceed 6% and decreases for longer delays). Although the comparison with a number of similar control algorithms [23], such as EPRCA, CAPC2, or ERICA, demonstrates superiority of the integral controller, its dynamics are rigorously bounded by the gain inversely proportional to the longest propagation delay, which is a serious limitation in WAN (wide area network) environment.

One of the most promising approaches to congestion control in connection-oriented networks, eliminating the problem of poor dynamics dictated by the stability requirements in long-haul transmission (which was a serious limitation of

PID schemes), employed the Smith principle [156]. The basic idea of the Smith principle is to create a projection of the model of the controlled system inside the controller and use it to establish appropriate command signals. As applied to communication networks [28, 29, 61, 66, 111–114, 129], the Smith predictor (SP) internally compensates for the feedback delay in the (outer) control loop. Once the problem of long feedback latency is eliminated, the system stability can be ensured without throttling the dynamics and degradation of the desired fast transient response.

In paper [111], Mascolo considered a single connection congestion control problem in a general packet-switching network. He used the deterministic fluid model approximation of packet flow and exploited transfer functions to describe the network dynamics. The designed continuous-time controller was applied to the ABR traffic control in ATM network and compared with the ERICA standard. The same author extended the idea of the SP to control the network supporting multiple data flows with different propagation delays in [112] and introduced feed-forward compensation of the available bandwidth in [113]. The proposed control algorithm guarantees no cell loss, full and equal network utilization, and ensures exponential convergence of the queue levels to stationary values without oscillations or overshoots. Gómez-Stern et al. [61] further studied the flow control using the Smith principle. They proposed a continuous-time PI controller, which helps reduce the average queue level and its sensitivity to the available bandwidth. On the other hand, the application of the Smith principle for satellite networks was considered in [129]. In that work, similarly as in [61], the saturation issues in the system with proportional continuous-time controller were handled using anti-wind up techniques. In a more recent paper [114], Mascolo demonstrated that also the TCP flow control mechanism implements the Smith predictor to handle the congestion. The result presented in [114] was supplemented with the analysis of the performance of the SP-based solutions as compared with the traditional PID controllers. It was shown that in the time-delay systems, the stability requirements significantly limit the dynamics of the PID-based schemes, and the Smith principle provides faster reaction to the varying networking conditions.

Although the SP-based controllers demonstrate outstanding performance both in steady state and during transient periods, they bear a serious flaw. The operation of those schemes largely depends on the accuracy of the system representation in the internal loop. In other words, the SP works correctly as long as the model of the regulated plant precisely coincides with the one used by the controller. Otherwise, for example, when the plant parameters are estimated with errors or are subject to severe uncertainties, the effectiveness of the Smith principle rapidly drops. To address this problem, a nonlinear algorithm exploiting the idea of the SP for the flow regulation in time-delay systems was proposed in [28]. The described continuous-time control mechanism guarantees congestion alleviating features and full resource usage even though the propagation delays (which constitute the principal model parameter) in a multisource network can be determined only with a limited degree of accuracy. The on-off controller [28] retains the propitious features of the earlier SP-based schemes with smaller buffer capacity requirements and simpler signaling (it can be realized as a binary feedback algorithm). A possible limitation of this scheme, as revealed in [67], is the required fast switching of the source rate, which can downgrade the transmission consistency. A discrete-time version of the SP-based nonlinear controllers, addressing the robustness issues related to parameter uncertainty in addition to the variations of the number of active connections, was presented in [29]. On the other hand, the SP-based control in the networks with finite, nonuniform sampling was addressed in [66].

The majority of the results discussed so far are intended for the networks with explicit, multibit feedback provision. There is also a great body of literature devoted to the network modeling and control related to the general-type networks and networks with implicit feedback. By no means, the review which follows should be treated as exhaustive. We will limit the discussion to highlighting the popular ideas and main research trends. For a more detailed treatment of the subject, the interested reader can refer, for example, to [5, 96, 109, 118, 157].

Among the systematic approaches to efficient network modeling and design of traffic regulation algorithms in general-type networks, one should certainly consider the utility-based optimization framework [97], further elaborated on in [96, 109, 157], and the control of heterogeneous networks, for example, [118]. In this class of problems, one may also find solutions of the combined control of elastic and inelastic traffic given within the DiffServ framework, for example, presented in [127]. However, still, the prevailing control mechanism deployed in the Internet is the TCP congestion control with the source transmission rate regulated according to the Jacobson's algorithm (and a number of its enhancements [157]), which essentially constitutes an implicit feedback system. Apart from the changes of this fundamental mechanism, such as delay-based TCP Vegas [37], various authors proposed the use of network-assisted control leading to an explicit feedback system within the TCP/IP framework. The key regulatory mechanisms in the network-assisted control for the TCP/IP networks are ECN and AQM. ECN [54, 137] is based on the idea of signaling the information about the network state to the sources by the network nodes in the form of marking additional bit(s) in the TCP header. AQM schemes, in turn, generate the explicit feedback information for regulating the inflow of data by observing the state of the node. Among the marking schemes, one can point out random early detection (RED) [56], random exponential marking (REM) [16], Blue [52], adaptive virtual queue (AVQ) [105], and many others (see [5]). Typically, an AQM scheme applies the signaling properties provided by ECN bits. Although the single packet marking provides satisfactory performance in many traffic scenarios, further improvements are sought in using two-bit [172] or multibit fields [93, 171] to inform the sources in a more exact way about the rate updates. In certain works, one may also observe a tendency of adapting the results developed for the traditional communication systems (such as the solution to the ATM/ABR flow regulation problem) to marking schemes present in the TCP/IP networks. Good examples of such approach in the network evolution are the works of Quet et al. [130] and Xia et al. [172], which can be perceived as TCP adaptations of the ideas presented in [131] and [91], respectively.

2.5 Sliding-Mode Congestion Control

In modern telecommunication networks, high throughput and robustness are of primary concern for handling the diversity of services and meeting the traffic demand of the users. The critical issue for achieving this desirable property remains in the use of efficient congestion control (or flow control) algorithms. On the other hand, numerous publications discussed in this review demonstrate the benefits of applying systematic theoretical approach in the design of such algorithms. Among the available techniques, one with particularly appealing robustness properties and efficiency in stabilizing complex nonlinear systems is sliding-mode control (SMC) [166]. However, so far, very few approaches have appeared in the literature regarding the application of SMC to regulate data traffic in communication networks. Several examples are discussed below.

The majority of SMC applications for data flow control reported so far in the literature are intended for the networks with implicit feedback. We can point out several successful design examples for AQM flow control in continuous-time domain, for example, [142, 173], yet those results are obtained without explicit consideration of the delay in the feedback and data paths. Also, a fuzzy SM controller [88] combining the benefits of linear and terminal SMC for improving the error convergence has been proposed for a simplified delay-free network model. Interesting combination of fuzzy and integral SMC has been reported for DiffServ networks [177], where the premium traffic is regulated by a fuzzy SMC algorithm and the ordinary service by an ISM controller. Also, in the DiffServ framework, an adaptive SM controller has been designed (using the backstepping procedure) for a model which neglects the feedback latency [180]. On the other hand, to regulate the flow of data in a DiffServ network with delay, the second-order SM technique was applied by Zhang et al. [178]. The three second-order SM controllers proposed in [178] outperform the standard one in chattering reduction both in the control of ordinary and premium traffic. As a result, more feasible controller command is obtained.

The work on flow control schemes for TCP/IP networks using the principles of SMC continues in [87, 169, 174, 176, 181], which give explicit consideration to the issues of nonnegligible latency except for [174]. In [176], the time delay of input signal is taken into account in the design of an AQM control algorithm, but stability is considered only with regard to matched uncertainties. For a similar model with input delay accounted for, the effects of mismatched uncertainties have been analyzed in [169]. On the other hand, both the input and state delay have been taken into consideration in [87], and the maximum allowable value of delay necessary for the system stability has been established. A discrete-time SMC approach to AQM controller design has been presented in [174], but the result is derived without explicit consideration of input or state delay. In that work, the sliding surface is selected to obtain robust asymptotic stability in the presence of parameter uncertainties by the linear matrix inequality (LMI) method. The LMI method has also been employed in the design of the observer-based AQM controller for a system with uncertainties, input delay, and saturated input [181]. The observer is used to estimate the average transmission window at the controlled source and drive the queue length to the target value. The authors show that the observer-based controller [181] achieves faster response and less oscillatory transient behavior in comparison with the algorithm described in [176].

Recently, a few publications appeared in the field of SMC and utility-optimization network traffic regulation. A discrete-time SM algorithm for adapting the source transfer rate in the utility max-min flow control framework was proposed in [86]. The dynamics of the source transmission rate in this algorithm is governed by the comparison of the source utility function and the value generated from the binary feedback information obtained from the network. The authors show that the utility max-min fairness and asymptotic stability are achieved in the considered framework. However, the entire analysis is performed for a delay-free system. A potentially impactful result in the context of general topology networks with delay and utility-based optimization has been reported in [175]. In a discrete-time setting, the authors show that any max-min fair system with a stable symmetric Jacobian remains asymptotically stable under arbitrary directional delays. This means that if a congestion control algorithm is designed so that the networking system has stable symmetric Jacobian, the stability of a delayed linear system may be examined based on the coefficient matrices of the corresponding undelayed system.

Clearly, the application of SMC in the problems of data flow control in communication networks received some attention in recent years. However, the important issues related to nonnegligible delay in systems with finite sampling rate [67–69] remain to much extent unexplored. In this work, we intend to fill some of the gaps in the systematic application of SMC to the traffic regulation problem in data transmission networks. The effects of delay are given thorough consideration, both in the analytical and numerical studies.

References

1. Afek Y, Mansour Y, Ostfeld Z (1996) Phantom: a simple and effective flow control scheme. In: Proceedings of the ACM SIGCOMM, Palo Alto, USA, pp 169–182
2. Ahuja V (1979) Routing and flow control in systems network architecture. IBM Syst J 18:298–314
3. Ait-Hellal O, Altman E, Basar T (1996) Rate based flow control with bandwidth information. Special Issue on ABR Eur Trans Telecommun 8:55–66
4. Akujobi F, Lambadaris J, Makkar R, Seddigh N, Nandy B (2002) BECN for congestion control in TCP/IP networks: study and comparative evaluation. In: Proceedings of the IEEE Globecom, vol 3, Taipei, Taiwan, pp 2588–2593
5. Almeida A, Belo C (2010) Explicit congestion control based on 1-bit probabilistic marking. Comput Commun 33:S30–S40
6. Alpcan T, Basar T (2005) A globally stable adaptive congestion control scheme for Internet-style networks with delay. IEEE/ACM Trans Netw 13:1261–1274
7. Altman E, Basar T (1995) Optimal rate control for high speed telecommunication networks. In: Proceedings of the 34th IEEE conference decision and control, vol 2, New Orleans, USA, pp 1389–1394

8. Altman E, Basar T (1997) Multi-user rate-based flow control: distributed game-theoretic approach. In: Proceedings of the 36th IEEE conference decision and control, vol 3, San Diego, USA, pp 2916–2921
9. Altman E, Basar T (1998) Multiuser rate-based flow control. IEEE Trans Commun 46: 940–949
10. Altman E, Baccelli F, Bolot JC (1994) Discrete-time analysis of adaptive rate control mechanisms. High Speed Netw Their Perform C-21:121–140
11. Altman E, Basar T, Srikant R (1997) Multi-user rate-based flow control with action delays: a team-theoretic approach. In: Proceedings of the 36th IEEE conference of decision and control, vol 3, San Diego, USA, pp 2387–2392
12. Altman E, Basar T, Srikant R (1998) Robust rate control for ABR sources. In: Proceedings of the IEEE INFOCOM, vol 1, San Fransisco, USA, pp 166–173
13. Altman E, Basar T, Srikant R (1999) Congestion control as a stochastic control problem with action delays. Automatica 35:1937–1950
14. Altman E, Basar T, Srikant R (2002) Nash equilibria for combined flow control and routing in networks: asymptotic behavior for a large number of users. IEEE Trans Autom Control 47:917–930
15. Aras CM, Kurose JF, Reeves DS, Schulzrinne H (1994) Real-time communication in packet-switched networks. Proc IEEE 82:122–139
16. Athuraliya S, Lapsley DE, Low SH (1999) Random early marking for Internet congestion control. In: Proceedings of IEEE Globecom, Rio de Janeiro, Brazil, pp 1747–1752
17. ATM Forum Traffic Management (1996) The ATM Forum Traffic Management specification version 4.0
18. ATM Forum Traffic Management (1999) The ATM Forum Traffic Management specification version 4.1
19. ATM Forum (2002) ATM User-Network Interwork Interface (UNI) specification version 4.1
20. Aweya J, Montuno DY, Ouellette M (2004) Effects of control loop delay on the stability of a rate control algorithm. Int J Commun Syst 17:833–850
21. Aweya J, Ouellette M, Montuno DY (2001) Discrete-time analysis of a rate control mechanism. Perform Eval 43:63–94
22. Aweya J, Ouellette M, Montuno DY (2001) A simple, scalable and provably stable explicit rate computation scheme for flow control in computer networks. Int J Commun Syst 14: 593–618
23. Aweya J, Ouellette M, Montuno DY (2004) Design and stability analysis of a rate control algorithm using the Routh-Hurwitz stability criterion. IEEE/ACM Trans Netw 12:719–732
24. Azmoodeh M, Davison R (1997) Performance management of public ATM networks – a scaleable and flexible approach. Proc IEEE 85:1639–1645
25. Barnhart A (1994) Use of the extended PRCA with various switch mechanisms. ATM Forum Contribution 94–0898
26. Barnhart A (1994) Explicit rate performance evaluations. ATM Forum Contribution 94–0983R1
27. Barnhart A (1995) Example switch algorithm for Sec 5.4 of TM Specification. ATM Forum Contribution 95–0195
28. Bartoszewicz A (2006) Nonlinear flow control strategies for connection oriented communication networks. Proc IEE Part D: Control Theory Appl 153:21–28
29. Bartoszewicz A, Molik T, Ignaciuk P (2009) Discrete time congestion controllers for multi-source connection-oriented communication networks. Int J Control 82:1237–1252
30. Benmohamed L, Meerkov SM (1993) Feedback control of congestion in packet switching networks: the case of a single congested node. IEEE/ACM Trans Netw 1:693–708
31. Benmohamed L, Meerkov SM (1997) Feedback control of congestion in packet switching networks: the case of multiple congested nodes. Int J Commun Syst 10:227–246
32. Black D, Blake S, Carlson M, Davies E, Wang Z, Weiss W (1998) An architecture for differentiated services. Netw Work Group, RFC 2475

33. Blanchini F, Lo Cigno R, Tempo R (2002) Robust rate control for integrated services packet networks. IEEE/ACM Trans Netw 10:644–652
34. Bolot JC (1992) A self-tuning regulator for adaptive overload control in communication networks. In: Proceedings of the 31st IEEE conference on decision and control, vol 1, pp 1022–1023
35. Bonomi F, Mitra D, Seery JB (1995) Adaptive algorithms for feedback-based flow control in high-speed, wide-area ATM networks. IEEE J Sel Areas Commun 13:1267–1283
36. Boyer PE, Tranchier DP (1992) A reservation principle with applications to the ATM traffic control. Comput Netw ISDN Syst 24:321–334
37. Brakmo LS, Peterson LL (1995) TCP Vegas: end-to-end congestion avoidance on a global Internet. IEEE J Sel Areas Commun 13:1465–1480
38. Bruni C, Delli Priscoli F, Koch G, Vergari S (2005) Traffic management in a band limited communication network: an optimal control approach. Int J Control 78:1249–1264
39. Carpenter BE, Nichols K (2002) Differentiated services in the Internet. Proc IEEE 90: 1479–1494
40. Charny A (1994) An algorithm for rate allocation in a packet-switching network with feedback. MSc Thesis, Massachusetts Institute of Technology
41. Charny A, Clark D, Jain R (1995) Congestion control with explicit rate indication. In: Proceedings of the IEEE International Communication conference, vol 3, pp 1954–1963
42. Chatté F, Ducourthial B, Nace D, Niculescu SI (2003) Fluid modelling of packet switching networks: perspectives for congestion control. Int J Syst Sci 34:585–597
43. Cheng RG, Chang CJ (1996) Design of a fuzzy traffic controller for ATM networks. IEEE/ACM Trans Netw 4:460–469
44. Chiu DM, Jain R (1989) Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. Comput Netw ISDN Syst 17:1–14
45. Chiussi FM, Xia Y, Kumar VP (1996) Dynamic max rate control algorithm for Available Bit Rate service in ATM networks. In: Proceedings of the IEEE Globecom, vol 3, London, UK, pp 2108–2117
46. Chong S, Nagarajan R, Wang YT (1998) First-order rate-based flow control with dynamic queue threshold for high-speed wide-area ATM networks. Comput Netw ISDN Syst 29: 2201–2212
47. Chong S, Nagarajan R, Wang YT (1998) Designing stable ABR flow control with rate feedback and open-loop control: first-order control case. Perform Eval 34:189–206
48. Danet A, Despres B, Le Best A, Pichon G, Ritzenthaler S (1976) The French public packet switching service: the TRANSPAC network. In: Proceedings of the 3rd international conference on computer communication, Toronto, Canada, pp 251–260
49. Douligeris C, Develekos G (1997) Neuro-fuzzy control in ATM networks. IEEE Commun Mag 35:154–162
50. El-Bawab TS, Shin JD (2002) Optical packet switching in core networks: between vision and reality. IEEE Commun Mag 40:60–65
51. El-Gendy MA, Bose A, Shin KG (2003) Evolution of the Internet QoS and support for soft real-time applications. Proc IEEE 91:1086–1104
52. Feng W, Kandlur D, Saha D, Shin K (1999) Blue: a new class of active queue management schemes. Tech Rep CSE-TR-387–99, University of Michigan
53. Firoiu V, Le Boudec JY, Towsley D, Zhang ZL (2002) Theories and models for Internet quality of service. Proc IEEE 90:1565–1591
54. Floyd S (1994) TCP and explicit congestion notification. ACM SIGCOMM Comput Commun Rev 24:10–23
55. Floyd S, Allman M (2007) Specifying new congestion control algorithms. Netw Work Group, RFC 5033
56. Floyd S, Jacobson V (1993) Random early detection gateways for congestion avoidance. IEEE/ACM Trans Netw 1:397–413
57. Frossard P, De Martin JC, Civanlar MR (2008) Media streaming with network diversity. Proc IEEE 96:39–53

58. Gerla M, Kleinrock L (1980) Flow control: a comparative survey. IEEE Trans Commun 28:553–574
59. Gerla M, Kleinrock L (1988) Congestion control in interconnected LANs. IEEE Netw 2: 72–76
60. Giessler A, Haenle JD, König A, Pade E (1978) Free buffer allocation – an investigation by simulation. Comp Netw 2:191–208
61. Gómez-Stern F, Fornés J, Rubio F (2002) Dead-time compensation for ABR traffic control over ATM networks. Control Eng Pract 10:481–491
62. Hluchyj M et al (1994) Closed-loop rate-based traffic management. ATM Forum Contribution 94–0211R3
63. Hughes D, Daley P (1994) Limitations of credit based flow control. ATM Forum Contribution 94–0776
64. Hunt D, Sathaye S, Brinkerhoff K (1994) The realities of flow control for ABR service. ATM Forum Contribution 94–0871
65. Hwang KS, Tan SW, Hsiao MC, Wu CS (2005) Cooperative multiagent congestion control for high-speed networks. IEEE Trans Syst Man Cybern B Cybern 35:255–268
66. Ignaciuk P, Bartoszewicz A (2008) Flow control in connection-oriented networks – a time-varying sampling period system case study. Kybernetika 44:336–359
67. Ignaciuk P, Bartoszewicz A (2008) Linear quadratic optimal discrete time sliding mode controller for connection oriented communication networks. IEEE Trans Ind Electron 55:4013–4021
68. Ignaciuk P, Bartoszewicz A (2009) Linear quadratic optimal sliding mode flow control for connection-oriented communication networks. Int J Robust Nonlinear Control 19:442–461
69. Ignaciuk P, Bartoszewicz A (2011) Discrete-time sliding-mode congestion control in multi-source communication networks with time-varying delay. IEEE Trans Control Syst Technol 19:852–867
70. Imer OC, Compans S, Basar T, Srikant R (2001) Available Bit Rate congestion control in ATM networks. IEEE Control Syst Mag 21:38–56
71. Izmailov R (1995) Adaptive feedback control algorithms for large data transfers in high-speed networks. IEEE Trans Autom Control 40:1469–1471
72. Izmailov R (1996) Analysis and optimization of feedback control algorithms for data transfer in high-speed networks. SIAM J Control Optim 34:1767–1780
73. Jacobson V (1988) Congestion avoidance and control. In: Proceedings of the ACM SIGCOMM, Stanford, USA, pp 314–329
74. Jaffe JM (1981) Bottleneck flow control. IEEE Trans Commun 29:954–962
75. Jagannathan S, Talluri J (2002) Predictive congestion control of ATM networks: multiple sources/single buffer scenario. Automatica 38:815–820
76. Jain R (1989) A delay based approach for congestion avoidance in interconnected heterogeneous computer networks. ACM SIGCOMM Comput Commun Rev 19:56–71
77. Jain R (1990) Congestion control in computer networks – issues and trends. IEEE Netw Mag 4:24–30
78. Jain R (1992) Myths about congestion management in highspeed networks. Internetw Res Exp 3:101–113
79. Jain R (1996) Congestion control and traffic management in ATM networks: recent advances and a survey. Comput Netw ISDN Syst 28:1723–1738
80. Jain R, Kalyanaraman S, Viswanathan R (1994) The OSU scheme for congestion avoidance using explicit rate indication. ATM Forum Contribution 94–0883
81. Jain R, Kalyanaraman S, Viswanathan R (1994) The EPRCA+ scheme. ATM Forum Contribution 94–0988
82. Jain R, Kalyanaraman S, Viswanathan R (1997) The OSU scheme for congestion avoidance in ATM networks: lessons learnt and extensions. Special issue on Traffic Control in ATM Networks: Perform Eval 31:67–88
83. Jain R, Kalyanaraman S, Viswanathan R, Goyal R (1995) A sample switch algorithm. ATM Forum Contribution 95–0178

84. Jain R, Kalyanaraman S, Goyal R, Fahmy S, Lu F (1995) ERICA+: extensions to the ERICA switch algorithm. ATM Forum Contribution 95–1145R1
85. Jain R, Ramakrishnan K, Chiu D (1987) Congestion avoidance in computer networks with a connectionless network layer. Tech Rep DEC-TR-506, Digital Equipment Corporation
86. Jin J, Wang WH, Palaniswami M (2009) A simple framework of utility max-min flow control using sliding mode approach. IEEE Commun Lett 13:360–362
87. Jing Y, He L, Dimirovski GM, Zhu H (2007) Robust stabilization of state and input delay for Active Queue Management algorithm. In: Proceedings of the American control conference, New York City, USA, pp 3083–3087
88. Jing Y, Yu N, Kong Z, Dimirovski GM (2008) Active Queue Management algorithm based on fuzzy sliding model controller. In: Proceedings of the 17th IFAC World Congress, Seoul, South Korea, pp 6148–6153
89. Johansson P, Nilsson AA (1997) Discrete time stability analysis of an explicit rate algorithm for the ABR service. In: Proceedings of IEEE ATM Workshop, 339–350
90. Kalyanaraman S (1997) Traffic management for the Available Bit Rate (ABR) service in Asynchronous Transfer Mode (ATM) networks. PhD dissertation, The Ohio State University
91. Kalyanaraman S, Jain R, Fahmy S, Goyal R, Vandalore B (2000) The ERICA switch algorithm for ABR traffic management in ATM networks. IEEE/ACM Trans Netw 8:87–98
92. Kameda H, Altman E, Pourtallier O (2008) A mixed optimum in symmetric distributed computer systems. IEEE Trans Autom Control 53:631–635
93. Katabi D, Handley M, Rohrs ChE (2002) Congestion control for high bandwidth-delay product networks. In: Proceedings of the ACM SIGCOMM, Pittsburgh, USA, pp 89–102
94. Kataria D (1994) Comments on rate-based proposal. ATM Forum Contribution 94–0384
95. Kawahara K, Oie Y, Murata M, Miyahara H (1995) Performance analysis of reactive congestion control for ATM networks. IEEE J Sel Areas Commun 13:651–661
96. Kelly FP (2001) Mathematical modelling of the Internet. In: Engquist B, Schmid W (eds) Mathematics unlimited: 2001 and beyond. Springer, Berlin
97. Kelly FP, Maulloo AK, Tan DKH (1998) Rate control for communication networks: shadow prices, proportional fairness and stability. J Oper Res Soc 49:237–252
98. Keshav S (1991) A control-theoretic approach to flow control. In: Proceedings of the ACM SIGCOMM, Zurich, Switzerland, pp 3–15
99. Kim BK, Kim BG, Chong I (1996) Dynamic averaging interval algorithm for ERICA ABR control scheme. ATM Forum Contribution 96–0062
100. Kolarov A, Ramamurthy B (1999) A control-theoretic approach to the design of an explicit rate controller for ABR service. IEEE/ACM Trans Netw 7:741–753
101. Kulkarni L, Li S (1998) Performance analysis of rate based feedback control for ATM networks. IEEE/ACM Trans Netw 6:797–810
102. Kung H, Chang K (1995) Receiver-oriented adaptive buffer allocation in credit-based flow control for ATM networks. In: Proceedings of the IEEE INFOCOM, vol 1, Boston, USA, pp 239–252
103. Kung H, Chapman A (1993) The FCVC (Flow Controlled Virtual Channel) proposal for ATM networks. In: Proceedings of the 1993 international conference on network protocols, San Francisco, USA, pp 116–127
104. Kung H et al. (1994) Flow controlled virtual connections proposal for ATM traffic management. ATM Forum Contribution 94–0632R2
105. Kunniyur SS, Srikant S (2004) An Adaptive Virtual Queue (AVQ) algorithm for active queue management. IEEE/ACM Trans Netw 12:286–299
106. Laberteaux KP, Rohrs ChE, Antsaklis PJ (2002) A practical controller for explicit rate congestion control. IEEE Trans Autom Control 47:960–978
107. Le Boudec JY (2008) Rate adaptation, congestion control and fairness: a tutorial. Ecole Polytechnique Fédérale de Lausanne
108. Lengliz I, Kamoun F (2000) A rate-based flow control method for ABR service in ATM networks. Comput Netw 34:129–138

109. Low SH, Paganini F, Doyle JC (2002) Internet congestion control. IEEE Control Syst Mag 22:28–43
110. Lyles B, Lin A (1994) Definition and preliminary simulation of a rate-based congestion control mechanism with explicit feedback of bottleneck rates. ATM Forum Contribution 94–0708
111. Mascolo S (1999) Congestion control in high-speed communication networks using the Smith principle. Automatica 35:1921–1935
112. Mascolo S (2000) Smith's principle for congestion control in high-speed data networks. IEEE Trans Autom Control 45:358–364
113. Mascolo S (2003) Dead-time and feed-forward disturbance compensation for congestion control in data networks. Int J Syst Sci 34:627–639
114. Mascolo S (2006) Modeling the Internet congestion control using a Smith controller with input shaping. Control Eng Pract 14:425–435
115. Maxemchuk NF, El Zarki M (1990) Routing and flow control in high-speed wide-area networks. Proc IEEE 78:104–221
116. Misra V, Gong WB, Towsley D (2000) Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In: Proceedings of the ACM SIGCOMM, Stockholm, Sweden, pp 151–160
117. Nash J (1950) Non-cooperative games. PhD dissertation, Princeton University
118. Neely JM, Modiano E, Li ChP (2008) Fairness and optimal stochastic control for heterogeneous networks. IEEE/ACM Trans Netw 16:396–409
119. Newman P (1993) Backward explicit congestion notification for ATM local area networks. In: Proceedings of the IEEE Globecom, vol 2, Houston, USA, pp 719–723
120. Newman P, Marshall G (1993) BECN congestion control. ATM Forum Contribution 94–789R1
121. Newman P, Marshall G (1993) Update on BECN congestion control. ATM Forum Contribution 94–855R1
122. Ohsaki H, Murata M, Suzuki H, Ikeda C, Miyahara H (1995) Rate-based congestion control for ATM networks. ACM SIGCOMM Comput Commun Rev 25:60–72
123. Pan Z, Altman E, Basar T (1996) Robust adaptive flow control in high speed telecommunication networks. In: Proceedings of the 35th IEEE conference on decision and control, vol 2, Kobe, Japan, pp 1341–1346
124. Pietrabissa A, Delli Priscoli F, Fiaschetti A, Di Paolo F (2006) A robust adaptive congestion control for communication networks with time-varying delays. In: Proceedings of the 2006 IEEE international conference control applications, Munich, Germany, pp 2093–2098
125. Pitsillides A, Şekercioğlu Y (2000) Congestion control. In: Pedrycz W, Vasilakos A (eds) Computational intelligence in telecommunications networks. CRC Press, Boca Raton
126. Pitsillides A, Şekercioğlu Y, Ramamurthy G (1997) Effective control of traffic flow in ATM networks using Fuzzy logic based Explicit Rate Marking (FERM). IEEE J Sel Areas Commun 15:209–225
127. Pitsillides A, Ioannou P, Lestas M, Rossides L (2005) Adaptive nonlinear congestion controller for a differentiated-services framework. IEEE/ACM Trans Netw 13:94–107
128. Pras A, Schönwälder J, Burgess M, Festor O, Pérez GM, Stadler R, Stiller B (2007) Key research challenges in network management. IEEE Commun Mag 45:104–110
129. Priscoli FD, Pietrabissa A (2004) Design of bandwidth-on-demand (BoD) protocol for satellite networks modelled as time-delay systems. Automatica 40:729–741
130. Quet PF, Ataşlar B, Özbay H (2004) On the design of AQM Supporting TCP flows using robust control theory. IEEE Trans Autom Control 49:1031–1036
131. Quet PF, Ataşlar B, İftar A, Özbay H, Kalyanaraman S, Kang T (2002) Rate-based flow controllers for communication networks in the presence of uncertain time-varying multiple time-delays. Automatica 38:917–928
132. Radunović B, Le Boudec JY (2007) A unified framework for max-min and min-max fairness with applications. IEEE/ACM Trans Netw 15:1073–1083

133. Ramakrishnan K (1993) Issues with Backward Explicit Congestion Notification based congestion control. ATM Forum Contribution 93–870
134. Ramakrishnan K, Newman P (1995) Integration of rate and credit schemes for ATM flow control. IEEE Netw 9:49–56
135. Ramakrishnan K, Zavgren J (1994) Preliminary simulation results of hop-by-hop/VC flow control and early packet discard. ATM Forum Contribution 94–0231
136. Ramakrishnan K, Chiu D, Jain R (1987) Congestion avoidance in computer networks with a connectionless network layer, part IV: a selective binary feedback scheme for general topologies methodology. Techn Rep DEC-TR-510, Digital Equipment Corporation
137. Ramakrishnan K, Floyd S, Black D (2001) The addition of Explicit Congestion Notification (ECN) to IP. Netw Work Group, RFC 3168
138. Raubold E, Haenle JD (1976) A method of deadlock-free resource allocation and flow control in packet networks. In: Proceedings of the 3rd international conference on computational Communication, pp 483–487
139. Ren T, Dimirovski GM, Jing Y (2006) ABR traffic control over ATM network using fuzzy immune-PID controller. In: Proceedings of the American Control Conference, pp 4876–4881
140. Ren T, Dimirovski GM, Jing Y, Zheng X (2007) Congestion control using integral SMC for ATM networks with multiple time delays and varying bandwidth. In: Proceedings of the 46th IEEE conference on decision and control, New Orleans, USA, pp 5192–5197
141. Ren T, Gao Z, Kong W, Jing Y, Yang M, Dimirovski GM (2008) Performance and robustness analysis of a fuzzy immune flow controller in ATM networks with time-varying multiple time delays. J Control Theory Appl 6:253–258
142. Ren F, Lin C, Yin X (2005) Design a congestion controller based on sliding mode variable structure control. Comput Commun 28:1050–1061
143. Rinde J (1977) Routing and control in a centrally-directed network. In: Proceedings of the national computer conference, Dallas, USA, pp 603–608
144. Roberts JW (2004) Internet traffic, QoS, and pricing. Proc IEEE 92:1389–1399
145. Roberts L (1994) Enhanced PRCA (Proportional Rate-Control Algorithm). ATM Forum Contribution 94–0735R1
146. Roberts L et al (1994) Closed-loop rate-based traffic management. ATM Forum Contribution 94–0438R1
147. Rohrs ChE, Berry R (1997) A linear control approach to explicit rate feedback in ATM networks. In: Proceedings of the IEEE INFOCOM, vol 1, Kobe, Japan, pp 277–282
148. Rohrs ChE, Berry R, O'Halek S (1996) Control engineer's look at ATM congestion avoidance. Comput Commun 19:226–234
149. Romanov A (1994) A performance enhancement for packetized ABR and VBR+ data. ATM Forum Contribution 94–0295
150. Sahin I, Simaan MA (2008) Competitive flow control in general multi-node multi-link communication networks. Int J Commun Syst 21:167–184
151. Saleh AM, Simmons JM (2006) Evolution toward the next-generation core optical network. IEEE J Lightwave Technol 24:3303–3321
152. Sichitiu ML, Bauer PH (2006) Asymptotic stability of congestion control systems with multiple sources. IEEE Trans Autom Control 51:292–298
153. Sichitiu ML, Bauer PH, Premaratne K (2003) The effect of uncertain time-variant delays in ATM networks with explicit rate feedback: a control theoretic approach. IEEE/ACM Trans Netw 11:628–637
154. Signal A, Jain R (2002) Terabit switching: a survey of techniques and current products. Comput Commun 25:547–556
155. Siu K, Tzeng H (1994) Adaptive proportional rate control for ABR service in ATM networks. Technical Report 94–07–01, University of California
156. Smith OJM (1959) A controller to overcome dead time. ISA J 6:28–33
157. Srikant R (2004) The mathematics of Internet congestion control. Birkhäuser, Boston
158. Su CF, Veciana G, Walrand J (2000) Explicit rate flow control for ABR services in ATM networks. IEEE/ACM Trans Netw 8:350–361

159. Sun DH, Zhang QH, Mu ZC (2004) Single parametric fuzzy adaptive PID control and robustness analysis based on the queue size of network node. In: Proceedings of the 2004 international conference on machine learning and cybernetics, vol 1, Shanghai, China, pp 397–400
160. Tan L, Pugh AC, Yin M (2003) Rate-based congestion control in ATM switching networks using a recursive digital filter. Control Eng Pract 11:1171–1181
161. Tanenbaum AS, Wetherall DJ (2010) Computer networks, 5th edn. Prentice Hall, Boston
162. Tarraf AA, Habib IW, Saadawi TN (1995) Intelligent traffic control for ATM broadband networks. IEEE Commun Mag 33:76–82
163. Tsang D, Wong W (1996) A new rate-based switch algorithm for ABR traffic to achieve max-min fairness with analytical approximation and delay adjustment. In: Proceedings of the IEEE INFOCOM, vol 3, San Francisco, USA, pp 1174–1181
164. Tymes L (1981) Routing and flow control in TYMNET. IEEE Trans Commun 29:392–398
165. Tzeng HY, Siu KY (1994) Enhanced Credit-based Congestion Notification (ECCN) flow control in ATM networks. ATM Forum Contribution 94–0450
166. Utkin VI (1992) Sliding modes in control and optimization. Springer, Berlin/Heidelberg
167. Utkin VI, Shi J (1996) Integral sliding mode in systems operating under uncertainty conditions. In: Proceedings of the 35th IEEE conference on decision and control, Kobe, Japan, 4591–4596
168. Ünal HU, Ataşlar B, İftar A, Özbay H (2006) H^{∞}-based flow control in data-communication networks with multiple time-delays. Technical Report 2006–001, Anadolu University
169. Wang H, Jing Y, Zhou Y, Chen Z, Liu X (2008) Sliding mode control for uncertain time-delay TCP/AQM network systems. In: Proceedings of the 17th IFAC World Congress, Seoul, South Korea, pp 12013–12018
170. Wolf LC, Griwodz C, Steinmetz R (1997) Multimedia communication. Proc IEEE 85: 1915–1933
171. Wu H, Ren F, Muc D, Gong X (2009) An efficient and fair explicit congestion control protocol for high bandwidth-delay product networks. Comput Commun 32:1138–1147
172. Xia Y, Subramanian L, Stoica I, Kalyanaraman S (2005) One more bit is enough. In: Proceedings of the ACM SIGCOMM, Philadelphia, USA, pp 37–48
173. Yan P, Gao Y, Özbay H (2005) A variable structure control approach to active queue management for TCP with ECN. IEEE Trans Control Syst Technol 13:203–215
174. Yan M, Kolemisevska-Gugulovska TD, Jing Y, Dimirovski GM (2007) Robust discrete-time sliding-mode control algorithm for TCP networks congestion control. In: Proceedings of the TELSIKS 2007, Nis, Serbia, pp 393–396
175. Yang M, Jing Y, Dimirovski GM, Zhang N (2007) Stability and performance analysis of a congestion control algorithm for networks. In: Proceedings of the 46th IEEE conference on decision and control, New Orleans, pp 4453–4458
176. Yin F, Dimirovski GM, Jing Y (2006) Robust stabilization of input delay for Internet congestion control. In: Proceedings of the American Control conference, Mineapolis, USA, pp 5576–5580
177. Zhang N, Jing Y, Yang M, Zhang S (2008) Robust AQM controller design for DiffServ network using sliding mode control. In: Proceedings of the 17th IFAC World Congress, Seoul, South Korea, pp 5635–5639
178. Zhang N, Yang M, Jing Y, Zhang S (2009) Congestion control for DiffServ network using second-order sliding mode control. IEEE Trans Ind Electron 56:3330–3336
179. Zhao Y, Li SQ, Sigarto S (1997) A linear dynamic model for design of stable explicit-rate ABR control schemes. In: Proceedings of IEEE INFOCOM, vol 1, Kobe, Japan, pp 283–292
180. Zheng X, Zhang N, Dimirovski GM, Jing Y (2008) Adaptive sliding mode congestion control for DiffServ network. In: Proceedings of the 17th IFAC World Congress, Seoul, South Korea, pp 12983–12987
181. Zhou Y, Wang H, Jing Y, Liu X (2008) Observer based robust controller design for Active Queue Management. In: Proceedings of the 17th IFAC World Congress, Seoul, South Korea, pp 12007–12012

Chapter 3
Fundamentals of Sliding-Mode Controller Design

The main purpose of control engineering is to steer the regulated plant in such a way that it operates in a required manner. The desirable performance of the plant should be obtained despite the unpredictable influence of the environment on all parts of the control system, including the plant itself, and no matter if the system designer knows precisely all the parameters of the plant. Even though parameters may change with time, load, and external circumstances, still, the system should preserve its nominal properties and ensure the required behavior of the plant. In other words, the principal objective of control engineering is to design control (or regulation) systems which are robust with respect to external disturbances and modeling uncertainty. This objective may be very well achieved using the sliding-mode technique [6, 11, 16, 18, 26, 28, 31, 43, 45, 55, 62, 66, 78, 79, 81, 85, 87], which is extensively used throughout this monograph. To be more precise, in the monograph, we focus our attention on the application of discrete sliding-mode control principles to the congestion elimination in data transmission networks. However, in order to make the text self-contained, we begin this chapter with presenting the main notions and concepts used in the field of variable structure systems and sliding-mode control.

In this chapter, we present basic concepts and definitions which will be used further in the book. First, we introduce the concept of variable structure system (VSS), and then we present the notions of sliding mode and sliding-mode control which are crucial for the issues presented in the main part of this book, i.e., in Chaps. 4, 5, and 6. We also present the basic properties of variable structure systems with sliding modes which make these systems a feasible option for many control applications, especially those which require good robustness with respect to model uncertainty and external disturbances. Next, the problem of chattering is exposed, and the fundamental techniques used to avoid (or at least reduce) this undesirable phenomenon are reviewed. The last section of this chapter introduces the concept of (quasi)sliding modes in discrete-time domain. In that section, the fundamental differences between continuous- and discrete-time SMC are discussed, and basic methods of controller design in discrete-time domain are presented.

P. Ignaciuk and A. Bartoszewicz, *Congestion Control in Data Transmission Networks: Sliding Mode and Other Designs*, Communications and Control Engineering,
DOI 10.1007/978-1-4471-4147-1__3, © Springer-Verlag London 2013

3.1 Variable Structure Systems

In recent years, much of the research in the area of control theory focused on the design of discontinuous feedback which switches the structure of the system according to the evolution of its state vector. This control idea may be illustrated by the following example.

Let us consider the second-order system

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= x_2 + u_i, \quad i = 1, 2, \end{aligned} \tag{3.1}$$

where $x_1(t)$ and $x_2(t)$ denote the system state variables (t is the time), with the following two feedback control laws:

$$u_1 = f_1(x_1, x_2) = -x_2 - x_1, \tag{3.2}$$

$$u_2 = f_2(x_1, x_2) = -x_2 - 4x_1. \tag{3.3}$$

Performance of system (3.1) controlled according to (3.2) is shown in Fig. 3.1, and Fig. 3.2 presents the behavior of the same system with feedback control (3.3). It can be clearly seen from those two figures that each of the feedback control laws (3.2) and (3.3) ensures the system stability only in the sense of Lyapunov.

However, if the following switching strategy is applied:

$$i = \begin{cases} 1, \text{ for } \min\{x_1, x_2\} < 0, \\ 2, \text{ for } \min\{x_1, x_2\} \geq 0, \end{cases} \tag{3.4}$$

then the system becomes asymptotically stable. This is illustrated in Fig. 3.3.

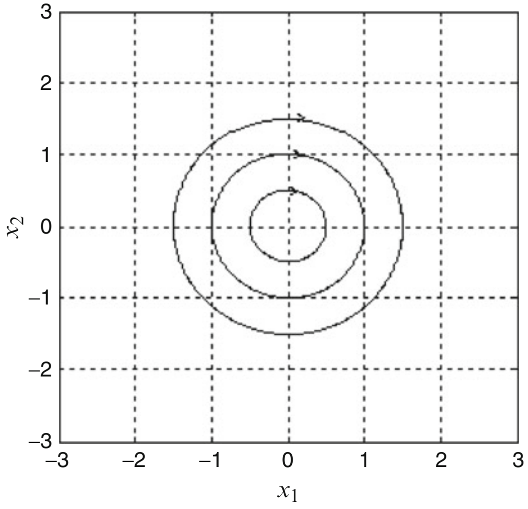


Fig. 3.1 Phase portrait of system (3.1) with controller (3.2)

Fig. 3.2 Phase portrait of system (3.1) with controller (3.3)

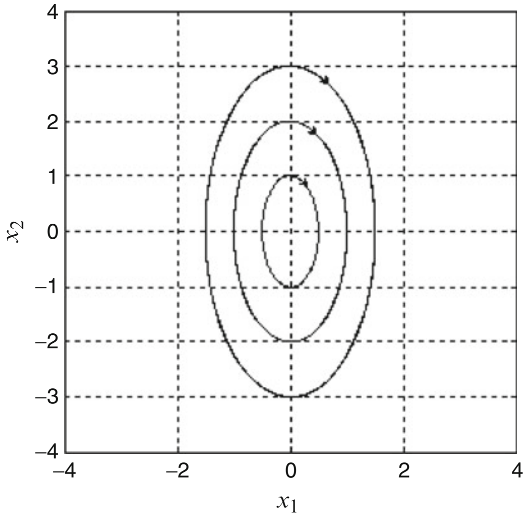
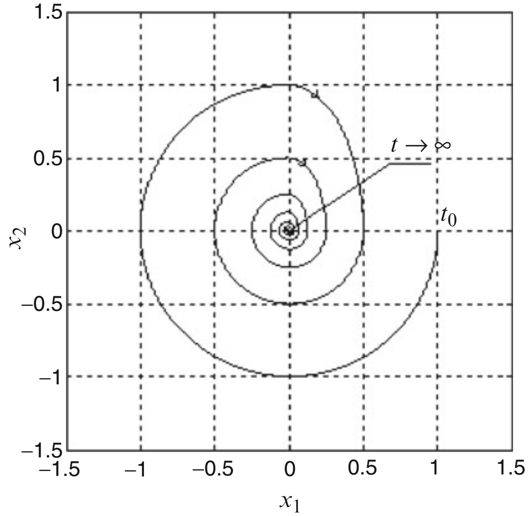


Fig. 3.3 Phase portrait of system (3.1) when switching strategy (3.4) is applied



Moreover, it is worth pointing out that system (3.1) with the same feedback control laws may exhibit completely different behavior (and even become unstable). For example, if the switching strategy (3.4) is modified as

$$i = \begin{cases} 1, \text{ for } \min\{x_1, x_2\} \geq 0, \\ 2, \text{ for } \min\{x_1, x_2\} < 0, \end{cases} \tag{3.5}$$

then the system output increases to infinity. The system dynamic behavior, in this situation, is illustrated in Fig. 3.4.

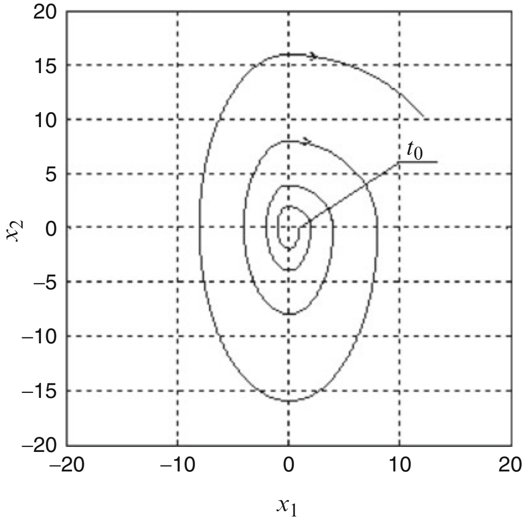


Fig. 3.4 Phase portrait of system (3.1) when switching strategy (3.5) is applied

This example presents the concept of variable structure control (VSC) and stresses that the system dynamics in VSC is determined not only by the applied feedback controllers but also, to a large extent, by the adopted switching strategy.

VSC is inherently a nonlinear technique, and as such, it offers a variety of advantages which cannot be achieved using conventional linear controllers. Our next example shows one of those favorable features – namely, it demonstrates that VSC may enable finite time error convergence. In this example, again we consider system (3.1); however, now we apply the following controller:

$$u = -x_2 - a\,\mathrm{sgn}\,(x_1) - b\,\mathrm{sgn}\,(x_2)\,, \tag{3.6}$$

where $a > b > 0$. Closer analysis of the behavior of system (3.1) with control law (3.6) demonstrates that, in this example, the system error converges to zero in finite time which can be expressed as

$$T = \frac{a}{b}\sqrt{2x_{10}}\left(\frac{1}{\sqrt{a-b}} + \frac{1}{\sqrt{a+b}}\right), \tag{3.7}$$

where x_{10} and $x_{20} = 0$ represent the initial conditions of system (3.1). Even though the error converges to zero in finite time, the number of oscillations in the system tends to infinity, with the period of oscillations decreasing to 0. This is illustrated in Figs. 3.5 and 3.6. In the simulation example presented in the figures, the following parameters are used: $a = 7$, $b = 3$, $x_{10} = 20$, and $x_{20} = 0$. Consequently, the system error is nullified at the time instant $t_n = 12.045$ and remains equal to zero for any time greater than t_n. Clearly, these favorable properties are achieved using finite control signal. This controller, due to the way the phase trajectory – shown in Fig. 3.5 – is drawn, is usually called “twisting controller.” It also serves as a good, simple example of the second-order SM controllers.

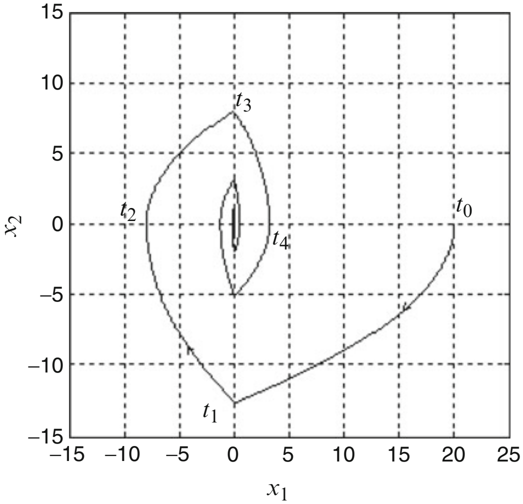


Fig. 3.5 Phase portrait of system (3.1) controlled according to (3.6)

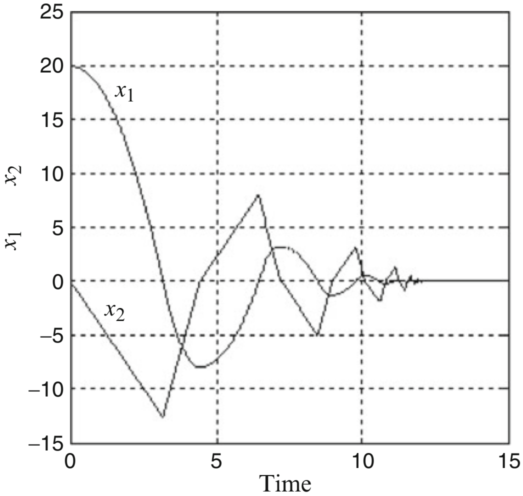


Fig. 3.6 State variables of system (3.1) controlled according to (3.6)

3.2 Sliding-Mode Control

The two examples presented up to now demonstrate the principal properties of VSC systems. However, the main advantage of the systems is obtained when the controlled plant exhibits the sliding motion [24, 35, 71, 77, 84]. The idea of SMC is to employ different feedback controllers acting on the opposite sides of a predetermined surface in the system state space. Each of those controllers pushes the system representative point towards the surface, so that this point approaches the surface, and once it hits the surface for the first time, it stays on it ever after.

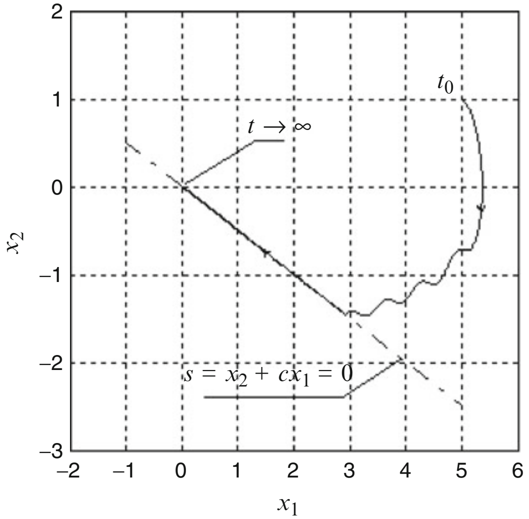


Fig. 3.7 Phase portrait of system (3.1) controlled according to (3.10)

The resulting motion of the system is restricted to the surface, which graphically can be interpreted as "sliding" of the system representative point along the surface. This idea is illustrated by the following example.

Let us consider another second-order plant

$$\begin{aligned}\dot{x}_1 &= x_2,\\ \dot{x}_2 &= b\cos(mx_1) + u, \quad |b| < 1,\end{aligned} \tag{3.8}$$

where b and m are possibly unknown constants. We select the following line in the state space:

$$s = x_2 + cx_1 = 0, \tag{3.9}$$

where c is a constant, and apply the controller

$$u = -cx_2 - \operatorname{sgn}(s). \tag{3.10}$$

In this equation sgn(·) function represents the sign of its argument, i.e., $\operatorname{sgn}(s < 0) = -1$ and $\operatorname{sgn}(s > 0) = +1$. With this controller, the system representative point moves towards line (3.9) always when it does not belong to the line. Then, once it hits the line, the controller switches the plant input (in the ideal case) with infinite frequency. Therefore, line (3.9) is called the switching line. Furthermore, since after reaching the line, the system representative point slides along it, then the line is also called the sliding line. This example is illustrated in Fig. 3.7.

The system parameters used in the presented simulation are $c = 0.5$, $b = 0.75$, and $m = 10$, and the simulation is performed for the following initial conditions:

$x_{10} = 5$ and $x_{20} = 1$. Notice that when the plant remains in the sliding mode, its dynamics is completely determined by the switching line (or in general the switching hypersurface) parameters. This implies that neither model uncertainty nor matched external disturbance affects the plant dynamics [25] which is a highly desirable system property. This property can also be justified geometrically, if one notices that in our example the slope of line (3.9) fully governs the plant motion in the sliding mode. Therefore, in SMC systems, we usually make the distinction between two phases: the first one, called the reaching phase, lasts until the controlled plant representative point hits the switching surface and the second one, the sliding phase, begins when the representative point reaches the surface. In the latter phase, the plant insensitivity to a class of modeling inaccuracies and external disturbances is ensured. Let us stress that the system robustness with respect to unmodeled dynamics, parameter uncertainty, and external disturbances is guaranteed only in the sliding mode. Therefore, shortening or (if possible) even complete elimination of the reaching phase is an important and timely research subject [15]. Thus, in the next section, we briefly present some recent results concerning this issue.

Another immediate consequence of the fact that in the sliding mode, the system representative point is restricted to the switching hypersurface (which is a subset of the state space) is reduction of the system order. If the system of the order n has m-independent inputs, then the sliding mode takes place on the intersection of m hypersurfaces and the reduced order of the system is equal to the difference $n - m$. To be more precise, in multi-input systems, the sliding mode may take place either independently on each switching hypersurface or only on the intersection of the surfaces. In the first case, the system representative point approaches each surface at any time instant, and once it hits any of the surfaces, it stays on this surface ever after. This scenario is shown in Fig. 3.8. In the latter case, however, the system representative point does not necessarily approach each of the surfaces, but it always moves towards their intersection. In this case, which is illustrated in Fig. 3.9, the system representative point may hit a surface and move away from it (might possibly cross a switching surface), but once it reaches the intersection of all the surfaces, then the representative point never leaves it.

As it has already been mentioned, the switching surface completely determines the plant dynamics in the sliding mode. Therefore, selecting this surface [4, 15, 32, 36–39, 42] is one of the two major tasks in the process of SMC system design. In order to stress this issue, let us point out that the same controller which has been considered in the last example may result in a very different system performance, if the sliding line slope c is selected in another way. This can be easily noticed if one takes into account any negative c. Then, controller (3.10) still ensures stability of the sliding motion on line (3.9), i.e., the system representative point still converges to the line; however, the system is unstable since both state variables x_1 and x_2 tend to (either plus or minus) infinity while the system representative point slides away from the origin of the phase plane along line (3.9).

The other major task in the SMC system design is the selection of an appropriate control law. This can be achieved either by assuming a certain kind of control law (usually motivated by some previous engineering experience) and proving that this

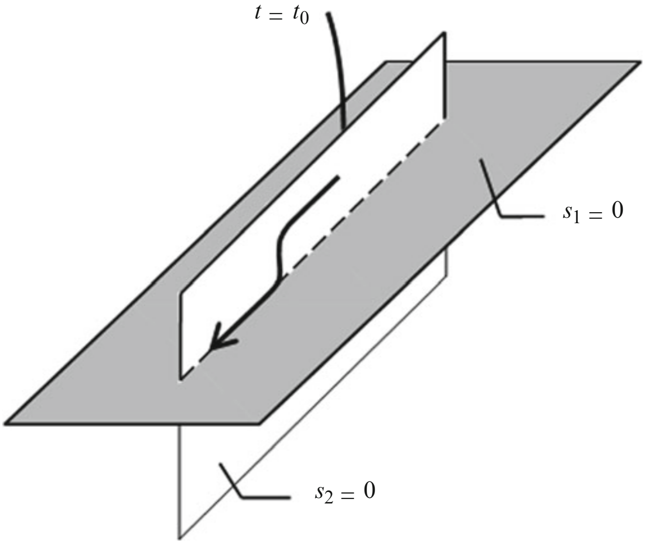


Fig. 3.8 Independent sliding motion on each switching surface

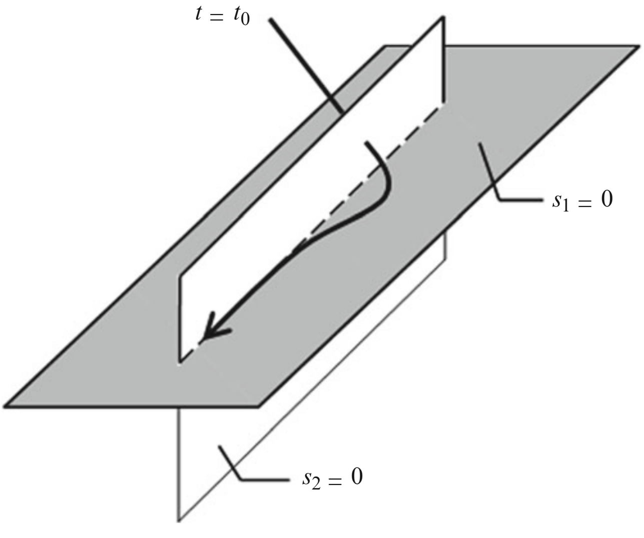


Fig. 3.9 Sliding motion on the intersection of the switching surfaces

control satisfies one of the so-called reaching conditions or by applying the reaching law approach. The reaching conditions [26] ensure stability of the sliding motion, and therefore, they are naturally derived using Lyapunov stability theory. On the other hand, if the reaching law approach is adopted for the purpose of an SM controller construction [35], then a totally different design philosophy is employed. In this case, the desired evolution of the switching variable $s(\cdot)$ is specified first, and then a control law ensuring that $s(\cdot)$ changes according to the specification is determined.

3.3 Chattering

SM controllers guarantee system insensitivity with respect to matched disturbance and model uncertainty and cause reduction of the plant order. Moreover, they are computationally efficient and may be applied to a wide range of various, possibly nonlinear and time-varying plants. However, often, they also exhibit a serious drawback which essentially hinders their practical applications. This drawback – high-frequency oscillations which inevitably appear in any real system whose input is supposed to switch infinitely fast – is usually called chattering. If system (3.8) exhibits any, even arbitrarily small, delay in the input channel, then control strategy (3.10) will cause oscillations whose frequency and amplitude depend on the delay. With the decreasing of the delay time, the frequency rises and the amplitude gets smaller. This is a highly undesirable phenomenon, because it causes serious wear and tear on the actuator components and may excite unmodeled high-frequency modes in the system. Therefore, a few methods to eliminate chattering have been proposed. The most popular of them uses function

$$\operatorname{sat}(s) = \begin{cases} -1, \text{ for} & s < -\rho, \\ \frac{1}{\rho}s, \text{ for} & |s| \le \rho, \\ 1, \text{ for} & s > \rho, \end{cases} \tag{3.11}$$

(where ρ is a positive, usually small constant) instead of $\operatorname{sgn}(s)$ in the definition of the discontinuous control term. Function $\operatorname{sat}(s)$ is illustrated in Fig. 3.10. With this modification, the term becomes continuous and the switching variable does not converge to zero but to the closed interval $[-\rho, \rho]$. Consequently, the system

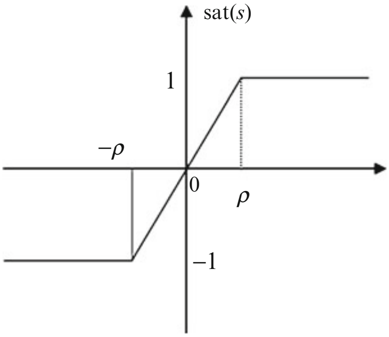


Fig. 3.10 Function sat(s)

representative point after the reaching phase termination belongs to a layer around the switching hyperplane, and therefore, this strategy is called boundary layer controller [71].

Other approaches to the chattering elimination include:

- Introduction of other nonlinear approximations of the discontinuous control term, for example, the so-called fractional approximation defined as

$$\mathrm{approx}(s) = \frac{s}{\rho + |s|} \tag{3.12}$$

  [1, 19, 74, 86].
- Replacing the boundary layer with a sliding sector [67, 83].
- Using dynamic SM controllers [5, 52, 61, 64, 68–70, 75, 88].
- Using fuzzy SM controllers [22, 56, 57].
- Using second- or higher-order SM controllers [7–9, 50, 51].

The phenomenon of chattering has been extensively analyzed in many papers using describing function method and various stability criteria [14, 23, 65].

3.4 Sliding Modes in Discrete-Time Systems

Early research in the field of SMC [29, 77] focused on continuous-time systems, which can change the plant input at any instant of time t. However, nowadays, with the increasing use of digital equipment in various feedback regulation problems, controllers are usually built as microprocessor systems and their output signal cannot change at any time, but only at the predetermined, discrete instants of time $t = kT$, where k is a natural number and T denotes the sampling (or discretization) period. Therefore, digitally implemented SMC systems often exhibit essentially different properties from the same systems realized using only analogue devices. Those digitally implemented systems are usually called discrete-time sliding-mode control (DSMC) systems, and they no longer ensure complete insensitivity of the controlled plant with reference to external disturbances and model uncertainty. However, they still offer some degree of robustness measured in a strictly defined sense, most often measured in terms of the switching variable magnitude.

There are two major approaches to the DSMC system design. One of them refers to the notion of the sliding hypersurface (or typically hyperplane), and the other one uses the concept of the sliding sector. The first approach is a natural generalization of the standard idea of continuous-time SMC with conventional sliding hyperplanes and has been developed gradually by many authors [10, 13, 44, 47, 53, 54, 63, 72, 73, 80]. In general, the idea comprises of selecting an appropriate sliding hyperplane $s(kT) = 0$ and choosing such a control law $u(kT)$ which ensures that (in the absence of disturbances and model uncertainty) the system representative point arrives at the hyperplane at the next instant of time, i.e., that $s[(k+1)T)] = 0$. We will illustrate this idea with the following, very standard example. Let us consider a single-input, linear discrete-time system

$$\mathbf{x}\left[(k+1)\,T\right] = \mathbf{A}\mathbf{x}\,(kT) + \mathbf{b}u\,(kT)\,, \tag{3.13}$$

where $\mathbf{x}$ is the state vector, $\mathbf{A}$ is an $n \times n$ state matrix, and $\mathbf{b}$ is an $n \times 1$ input vector. Then we choose the following sliding hyperplane

$$s\,(kT) = \mathbf{c}^{\mathrm{T}}\mathbf{x}\,(kT) = 0 \tag{3.14}$$

in the n-dimensional state space $\Re^n$, and in order to obtain $s[(k+1)T)] = 0$, we select

$$u\,(kT) = -\left(\mathbf{c}^{\mathrm{T}}\mathbf{b}\right)^{-1}\mathbf{c}^{\mathrm{T}}\mathbf{A}\mathbf{x}\,(kT)\,. \tag{3.15}$$

Clearly, vector $\mathbf{c}$ has to be chosen in such a way that $\mathbf{c}^{\mathrm{T}}\mathbf{b} \neq 0$. Substituting equations (3.13) and (3.15) into (3.14), one can easily notice that this controller actually brings the system representative point on plane (3.14) at the time instant $(k+1)T$. Thus, the ideal sliding motion without chattering takes place in the system. Moreover, when control strategy (3.15) is applied, then the closed-loop system state matrix has the following form $\mathbf{A}_{\mathbf{cl}} = [\mathbf{I}_n - \mathbf{b}(\mathbf{c}^{\mathrm{T}}\mathbf{b})^{-1}\mathbf{c}^{\mathrm{T}}]\mathbf{A}$, where $\mathbf{I}_n = \mathrm{diag}\{1,\ 1, \ldots,\ 1\}$ is an n by n identity matrix. This shows that the system dynamic performance may be tuned as desired, and various design objectives might be achieved. Those may include but are not limited to finite time error convergence (time optimal performance) [17], obtained when the closed-loop system characteristic polynomial has the form

$$\det\left(z\,\mathbf{I}_n - \mathbf{A}_{\mathbf{cl}}\right) = z^n; \tag{3.16}$$

linear-quadratic (LQ) optimal [36, 37, 42]; integral absolute error (IAE) optimal; integral time multiplied by the absolute error (ITAE) optimal performance; etc.

However, as opposed to the continuous SMC systems, the DSMC considered in this section ensures that the switching variable is equal to zero only at the sampling instants, while at any time t which is not an integer multiple of the discretization period the switching variable may attain certain values different from zero. In other words, intersampling behavior of this variable is not determined. Furthermore, if system (3.13) is subject to external disturbance $\mathbf{d}(kT)$, i.e., it is described as

$$\mathbf{x}\left[(k+1)\,T\right] = \mathbf{A}\mathbf{x}\,(kT) + \mathbf{b}u\,(kT) + \mathbf{d}\,(kT)\,, \tag{3.17}$$

where the absolute value of the ith component $d_i(kT)$ of vector $\mathbf{d}(kT)$ is always upper-bounded by a known, nonnegative value $d_{i\max}$ ($|d_i| \leq d_{i\max}$), then the system representative point no longer stays on plane (3.14) even at the discrete sampling instants but remains in a vicinity of the plane

$$|\, s\,[(k+1)\,T]\,| \leq \sum_{i=1}^{n} |\, c_i \,|\, d_{i\,\max} \tag{3.18}$$

instead. In other words, the system representative point remains in a band around the sliding hyperplane in the state space, and therefore, this type of motion is sometimes called a quasi-sliding mode or a nonideal sliding motion [12, 32]. Clearly, the thinner the band (smaller the width of the band), the better system robustness with respect to the considered disturbance is achieved [76].

Let us also point out that the discussion which has been presented in this section clearly shows that the notions of SMC and variable structure systems are not equivalent. In many classical papers, examples of continuous-time variable structure systems without sliding mode are presented [26, 77], and this example gives an idea of an SMC system which actually is not a variable structure system.

Another way of looking at this issue has been proposed in paper [32] where the discontinuous term was incorporated in the control law resulting in a zigzag motion around the sliding hyperplane. This zigzag motion is in some way similar to the chattering phenomenon which occurs in continuous-time SMC systems. However, the main difference between the continuous SMC and the DSMC proposed in [32] and further discussed in [13] is the frequency of oscillations which occur in the system. In the continuous-time SMC, the frequency tends to infinity, while in DSMC, the frequency is finite and it depends on the sampling period.

The other approach to the DSMC system design – originally proposed by Furuta in paper [30] – exploits the notion of the sliding sector [20, 58, 59]. The sector (cone-like region in the state space) is selected in such a way that if the system representative point belongs to it, then the point always approaches the state space origin.

Early DSMC schemes, discussed up to now, usually require full state vector (all state variables) to be available for measurement. However, this is rarely the case in practical implementations where some state variables may not be easily accessible and others may even have no physical meaning at all. Therefore, several researchers proposed output feedback [21, 26, 27, 33, 34, 48, 49, 60] and observer-based SM controllers [29, 34, 46, 82]. Probably the most significant and well-worked-out approach to the output feedback DSMC, known as multirate output feedback DSMC, has been proposed in [3, 40–42]. In this very attractive approach, the available output is sampled (measured) at a faster rate than the input signal rate which enables to obtain implicitly the unmeasurable state variables. We believe that currently the design of advanced sliding surfaces for discrete-time sliding-mode controllers might be identified as another important theoretical research topic in the field. These surfaces include linear hyperplanes optimal in the sense of quadratic performance index [36, 37, 42] and appropriately selected nonlinear hypersurfaces [2, 4].

In this section, we presented a concise introduction to the sliding-mode control of continuous- and discrete-time systems. We did not attempt to make our presentation particularly in-depth, exhaustive, or complete, but we tried to give a brief background for the issues presented further in this monograph. Indeed, in this section, we explained some basic notions which we will use in Chaps. 4, 5, and 6 for the design of the sliding-mode congestion controllers for data transmission networks.

References

1. Ambrosino G, Celentano G, Garofalo F (1984) Variable structure model reference adaptive control systems. Int J Control 39:1339–1349
2. Bandyopadhyay B, Fulwani D (2009) High-performance tracking controller for discrete plant using nonlinear sliding surface. IEEE Trans Ind Electron 56:3628–3637
3. Bandyopadhyay B, Janardhanan S (2006) Discrete-time sliding mode control: a multirate output feedback approach, LNCIS 323. Springer, Berlin/Heidelberg
4. Bandyopadhyay B, Fulwani D, Kim KS (2010) Sliding mode control using novel sliding surfaces, LNCIS 392. Springer, Berlin/Heidelberg
5. Bartolini G, Pydynowski P (1996) An improved, chattering free, VSC scheme for uncertain dynamical systems. IEEE Trans Autom Control 41:1220–1226
6. Bartolini G, Ferrara A, Spurgeon SK (eds) (1997) New trends in sliding mode control. Special Issue: Int J Robust Nonlinear Control 7:297–427
7. Bartolini G, Ferrara A, Usai E (1997) Applications of a sub-optimal discontinuous control algorithm for uncertain second order systems. Int J Robust Nonlinear Control 7:299–319
8. Bartolini G, Ferrara A, Usai E (1997) Output tracking control of uncertain nonlinear second order systems. Automatica 33:2203–2212
9. Bartolini G, Ferrara A, Usai E (1998) Chattering avoidance by second-order sliding mode control. IEEE Trans Autom Control 43:241–246
10. Bartolini G, Ferrara A, Utkin VI (1995) Adaptive sliding mode control in discrete-time systems. Automatica 31:769–773
11. Bartolini G, Fridman L, Pisano A (eds) (2008) Modern sliding mode control theory, LNCIS 375. Springer, Berlin/Heidelberg
12. Bartoszewicz A (1996) Remarks on ‘Discrete-time variable structure control systems’. IEEE Trans Ind Electron 43:235–238
13. Bartoszewicz A (1998) Discrete time quasi-sliding mode control strategies. IEEE Trans Ind Electron 45:633–637
14. Bartoszewicz A (2000) Chattering attenuation in sliding mode control systems. Control Cybern 29:585–594
15. Bartoszewicz A, Nowacka-Leverton A (2009) Time-varying sliding modes for second and third order systems, LNCIS 382. Springer, Berlin/Heidelberg
16. Bartoszewicz A, Patton RJ (eds) (2007) Sliding mode control. Special issue: Int J Adapt Control Signal Process 21:635–825
17. Bartoszewicz A, Żuk J (2009) Discrete-time sliding mode flow controller for multi-source connection-oriented communication networks. J Vib Control 15:1745–1760
18. Bartoszewicz A, Kaynak O, Utkin VI (eds) (2008) Sliding mode control in industrial applications. Special Issue: IEEE Trans Ind Electron 55:3806–4074
19. Burton C, Zinober AIS (1986) Continuous approximation of variable structure control. Int J Syst Sci 17:875–885
20. Chan CY (1991) Servo-systems with discrete-variable structure control. Syst Control Lett 17:321–325

21. Chan CY (1991) Discrete adaptive sliding mode control of a class of stochastic systems. Automatica 35:1491–1498
22. Choi SB, Kim JS (1997) A fuzzy-sliding mode controller for robust tracking of robotic manipulators. Mechatronics 7:199–216
23. Chung SChY, Lin ChL (1999) A transformed Luré problem for sliding mode control and chattering reduction. IEEE Trans Autom Control 43:563–568
24. DeCarlo RA, Żak SH, Matthews GP (1988) Variable structure control of nonlinear multivariable systems: a tutorial. Proc IEEE 76:212–232
25. Draženović B (1969) The invariance conditions in variable structure systems. Automatica 5:287–295
26. Edwards C, Spurgeon SK (1998) Sliding mode control: theory and applications. Taylor & Francis, London
27. Edwards C, Spurgeon SK, Hebden RG (2003) On the design of sliding mode output feedback controllers. Int J Control 76:893–905
28. Edwards C, Fossas Colet E, Fridman L (eds) (2006) Advances in variable structure and sliding mode control, LNCIS 334. Springer, Berlin/Heidelberg
29. Emelyanov SV (1967) Variable structure control systems. Nauka, Moscow (in Russian)
30. Furuta K (1990) Sliding mode control of a discrete system. Syst Control Lett 14:145–152
31. Gao W (ed) (1993) Variable structure control. Special Section: IEEE Trans Ind Elect 40:1–88
32. Gao W, Wang Y, Homaifa A (1995) Discrete-time variable structure control systems. IEEE Trans Ind Electron 42:117–122
33. Hsu L, Lizarralde F (1998) Comments and further results regarding 'On variable structure output feedback controllers'. IEEE Trans AutomControl 43:1338–1340
34. Hui S, Żak SH (1999) On discrete-time variable structure sliding mode control. Syst Control Lett 38:283–288
35. Hung JY, Gao W, Hung JC (1993) Variable structure control: a survey. IEEE Trans Ind Electron 40:2–22
36. Ignaciuk P, Bartoszewicz A (2008) Linear quadratic optimal discrete time sliding mode controller for connection oriented communication networks. IEEE Trans Ind Electron 55: 4013–4021
37. Ignaciuk P, Bartoszewicz A (2009) Linear quadratic optimal sliding mode flow control for connection-oriented communication networks. Int J Robust Nonlinear Control 19:442–461
38. Ignaciuk P, Bartoszewicz A (2012) LQ optimal and reaching law based sliding modes for inventory management systems. Int J Syst Sci 43:105–116
39. Ignaciuk P, Bartoszewicz A (2012) LQ optimal sliding-mode supply policy for periodic-review perishable inventory systems. J Franklin Inst. doi:10.1016/j.jfranklin.2011.04.003
40. Janardhanan S, Bandyopadhyay B (2006) Output feedback sliding-mode control for uncertain systems using fast output sampling technique. IEEE Trans Ind Electron 53:1677–1682
41. Janardhanan S, Bandyopadhyay B (2007) Multirate output feedback based robust quasi-sliding mode control of discrete-time systems. IEEE Trans Autom Control 52:499–503
42. Janardhanan S, Kariwala V (2008) Multirate-output-feedback-based LQ-optimal discrete-time sliding mode control. IEEE Trans Autom Control 53:367–373
43. Kaynak O (ed) (2001) Computationally intelligent methodologies and sliding-mode control. Special Issue: IEEE Trans Ind Elect 48:2–240
44. Kaynak O, Denker A (1993) Discrete time sliding mode control in the presence of system uncertainty. Int J Control 57:1177–1189
45. Kaynak O, Bartoszewicz A, Utkin VI (eds) (2009) Sliding mode control in industrial applications. Special Issue: IEEE Trans Ind Electron 56:3271–3784
46. Khan S, Sabanović A, Nergiz AO (2009) Scaled bilateral teleoperation using discrete-time sliding-mode controller. IEEE Trans Ind Electron 56:3609–3618
47. Kotta U (1989) Comments on 'on the stability of discrete-time sliding mode control systems'. IEEE Trans Autom Control 34:1021–1022
48. Kwan CM (1996) On variable structure output feedback controllers. IEEE Trans Autom Control 41:1691–1693

49. Kwan CM (2001) Further results on variable output feedback controllers. IEEE Trans Autom Control 46:1505–1508
50. Levant A (1993) Sliding order and sliding accuracy in sliding mode control. Int J Control 58:1247–1263
51. Levant A (2003) Higher-order sliding modes, differentiation and output-feedback control. Int J Control 76:924–941
52. Lu XY, Spurgeon SK (1998) A new sliding mode approach to asymptotic feedback linearization with application to the control of non-flat systems. J Appl Math Comput Sci 8:21–37
53. Milosavljević Č (1985) General conditions for the existence of a quasisliding mode on the switching hyperplane in discrete variable structure systems. Autom Remote Control 46: 307–314
54. Milosavljević Č, Peruničić-Draženović B, Veselić B, Mitić D (2006) Sampled data quasi-sliding mode control strategies. In: Proceedings of IEEE international conference on industrial technology, Mumbai, India, pp 2640–2645
55. Misawa E, Utkin VI (eds) (2000) Variable structure systems. Special Issue: Trans ASME – J Dyn Syst Meas Control 122:585–819
56. Palm R (1994) Robust control by fuzzy sliding mode. Automatica 30:1429–1437
57. Palm R, Driankov D, Hellendoorn H (1997) Model based fuzzy control. Springer, Berlin/Heidelberg
58. Pan Y, Furuta K (1997) Discrete-time VSS controller design. Int J Robust Nonlinear Control 7:373–386
59. Pan Y, Furuta K (2007) Variable structure control with sliding sector based on hybrid switching law. Int J Adapt Control Signal Process 21:764–778
60. Richter H, Misawa EA (2002) Boundary layer eigenvalues in observer based discrete-time sliding mode control. In: Proceedings of American Control Conference, Anchorage, USA, pp 2935–2936
61. Rios-Bolivar M, Zinober AIS, Sira-Ramirez H (1997) Dynamical adaptive sliding mode output tracking control of a class of nonlinear systems. Int J Robust Nonlinear Control 7:387–405
62. Sabanović A, Fridman L, Spurgeon SK (2004) Variable structure systems: from principles to implementation. IEE Book Series, London
63. Sarpturk SZ, Istefanopulos Y, Kaynak O (1987) On the stability of discrete-time sliding mode control systems. IEEE Trans Autom Control 32:930–932
64. Selisteanu D, Petre E, Rasvan VB (2007) Sliding mode and adaptive sliding-mode control of a class of nonlinear bioprocesses. Int J Adapt Control Signal Process 21:795–822
65. Shtessel Y, Lee YJ (1996) New approach to chattering analysis in systems with sliding modes. In: Proceedings of the 35th IEEE conference on decision and control, Kobe, Japan, pp 4014–4019
66. Shtessel Y, Fridman L, Zinober AIS (2008) Advances in higher order sliding mode control. Special Issue: Int J Robust Nonlinear Control 18:381–585
67. Shyu K, Tsai Y, Yung C (1992) A modified variable structure controller. Automatica 28: 1209–1213
68. Sira-Ramirez H (1993) A dynamical variable structure control strategy in asymptotic output tracking problems. IEEE Trans Autom Control 38:615–620
69. Sira-Ramirez H (1993) On the dynamical sliding mode control of nonlinear systems. Int J Control 57:1039–1061
70. Sira-Ramirez H, Llanes-Santiago O (1994) Dynamical discontinuous feedback strategies in the regulation of nonlinear chemical processes. IEEE Trans Control Syst Technol 2:11–21
71. Slotine JJ, Li W (1991) Applied nonlinear control. Prentice-Hall, Englewood Cliffs
72. Spurgeon SK (1991) Sliding mode control design for uncertain discrete-time systems. In: Proceedings of the 30th IEEE conference on decision and control, Sussex, UK, pp 2136–2141
73. Spurgeon SK (1992) Hyperplane design techniques for discrete-time variable structure control systems. Int J Control 55:445–456
74. Spurgeon SK, Davies RA (1993) A nonlinear control strategy for robust sliding mode performance in the presence of unmatched uncertainty. Int J Control 57:1107–1123

75. Spurgeon SK, Lu XY (1997) Output tracking using dynamic sliding mode techniques. Int J Robust Nonlinear Control 7:407–427
76. Su WCh, Drakunov SV, Özgüner Ü (2000) An O(T^2) boundary layer in sliding mode for sampled-data systems. IEEE Trans Autom Control 45:482–485
77. Utkin VI (1977) Variable structure systems with sliding modes. IEEE Trans Autom Control 22:212–222
78. Utkin VI (1992) Sliding modes in control and optimization. Springer, Berlin/Heidelberg
79. Utkin VI (ed) (1993) Sliding mode control. Special Issue: Int J Control 57:1003–1259
80. Utkin VI, Drakunov SV (1989) On discrete-time sliding mode control. In: Proceedings of the 4th IFAC symposium on nonlinear control system, Capri, Italy, pp 484–489
81. Utkin VI, Guldner J, Shi J (2009) Sliding mode control in electro-mechanical systems, 2nd edn. CRC Press, Boca Raton
82. Veluvolu KC, Soh YCh (2009) Discrete-time sliding-mode state and unknown input estimations for nonlinear systems. IEEE Trans Ind Electron 56:3443–3452
83. Xu J, Lee T, Wang M, Yu X (1996) Design of variable structure controllers with continuous switching control. Int J Control 65:409–431
84. Young KD, Utkin VI, Özgüner Ü (1999) A control engineer's guide to sliding mode control. IEEE Trans Control Syst Technol 7:328–342
85. Yu X (ed) (1998) Adaptive learning and control using sliding modes. Special Issue: J Appl Math Comput Sci 8:5–197
86. Yu H, Lloyd S (1997) Variable structure adaptive control of robot manipulators. Proc IEE Part D Control Theory Appl 144:167–176
87. Zinober ASI (1994) Variable structure and Lyapunov control. Springer, New York
88. Zlateva P (1996) Variable-structure control of nonlinear systems. Control Eng Pract 4: 1023–1028

Chapter 4
Flow Control in Continuous-Time Systems

In this chapter, we introduce the fundamental concepts behind the fluid-flow modeling of data traffic in communication networks. We emphasize the effects caused by action delay, which is the time that elapses from the moment the control information (or the controller command) is sent by a network node, the information reaches the data source it is destined for, appropriate action is taken by the source, and until subsequently that action affects the state of the node that issued the command. Indeed, as recognized in many significant papers, for example [3, 6, 7, 10, 14, 17, 22, 26], the existence of action delay constitutes the main obstacle in providing efficient control in data transmission networks, and it should be explicitly considered in the controller design and system analysis. Since we intend to make use of the benefits of SMC, which is well known to be robust and efficient regulation technique successfully applied in many engineering areas (see, e.g., recent special issues [4, 15, 25]), it is of paramount importance to account for the adverse effects of delay. This is due to the fact that delay reduces the system robustness – typically, mismatched perturbations are introduced and the invariance property [9] no longer holds – which threatens stability of the sliding motion. In the design procedures presented in this chapter, we overcome the delay problem by an appropriate selection of the switching function which incorporates a state predictor. In this way, the delay in the feedback loop no longer poses a stability threat, and the system dynamics can be tuned for the maximum responsiveness to the changes of networking conditions reflected in the fluctuations of the available bandwidth.

For the purpose of presentation, we adopt the approach frequently encountered in the literature on congestion control, and treat the case of a single flow (e.g., [1, 11, 17]), and multiple connections controlled by a network node (e.g., [2, 12, 18]) separately. We believe that such approach facilitates understanding of the system modeling and provides a good starting point for the exposition of sophisticated design procedures at a later stage. From the methodological point of view, it helps to get familiarized with the fundamental concepts of design first and then proceed to more elaborate schemes without being confused by the potential intricacies arising in more complex topologies.

P. Ignaciuk and A. Bartoszewicz, *Congestion Control in Data Transmission Networks: Sliding Mode and Other Designs*, Communications and Control Engineering,
DOI 10.1007/978-1-4471-4147-1_4, © Springer-Verlag London 2013

4.1 Flow Control in a Single Virtual Circuit

We begin the analysis of the networking phenomena accompanying the data transfer in communication systems by considering a single flow in a connection-oriented network. In this type of networks, data transmission is preceded by the connection setup phase, during which a VC between the source and its destination is created, and appropriate switching rules in the nodes along the data transmission path are established. Afterwards, the data source sends packets at the rate determined by the controller placed at a network node. Typically, the packets pass through a series of nodes operating in the store-and-forward mode without the traffic prioritization along the established data route to be finally delivered to the destination. It means that when a data packet is received at the node input link, it is directed to the buffer of the appropriate output interface, where it waits in the First-In-First-Out (FIFO) queue to be relayed to the next node on the established path or to be sent to the destination (if the considered node is the last one on the transfer route). However, somewhere on the transmission path, a node is encountered, whose output link cannot handle the incoming flow. Consequently, congestion occurs, and packets accumulate in the buffer allocated for that link. For the purpose of exposition of the fundamental flow control aspects, we assume in this chapter that the source is persistent, i.e., it has always enough data to transmit at the maximum rate allowed by the network. Thus, the congestion control problem can be solved only through appropriate input rate adjustment. The source transmission rate adjustments are performed exactly according to the command from the controller operating at the bottleneck node. This assumption will be relaxed in a latter part of this work – in Chaps. 6 and 7 – when the issues related to nonideal data sources and various rate constraints are addressed.

4.1.1 Network Model

The schematic diagram of the connection is presented in Fig. 4.1. The source sends packets into the network at a rate determined by the controller placed at the bottleneck node. The rate assigned by the controller is represented by function $u(t)$, where t is a continuous variable denoting time. After forward propagation delay T_{F} packets reach the bottleneck node and are served according to the bandwidth availability at the output link. The remaining data accumulates in the buffer allocated for the output (bottleneck) link. The packet queue length in the buffer, which at time t will be denoted as $y(t)$, and its demand value y_{D} are used to calculate the current amount of data to be sent by the source. The information about the new rate is available at the source with backward propagation delay T_{B} after being generated by the bottleneck node. We assume that the feedback information is delivered with priority over the user's data on the preestablished path (it is not subject to the queuing delay at the nodes), which implies that the delay in the feedback loop $\mathrm{RTT} = T_{\mathrm{F}} + T_{\mathrm{B}}$ remains constant for the duration of the connection. Moreover, RTT

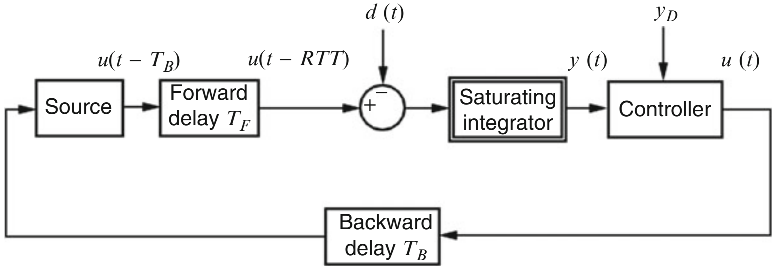


Fig. 4.1 Network model – single virtual circuit

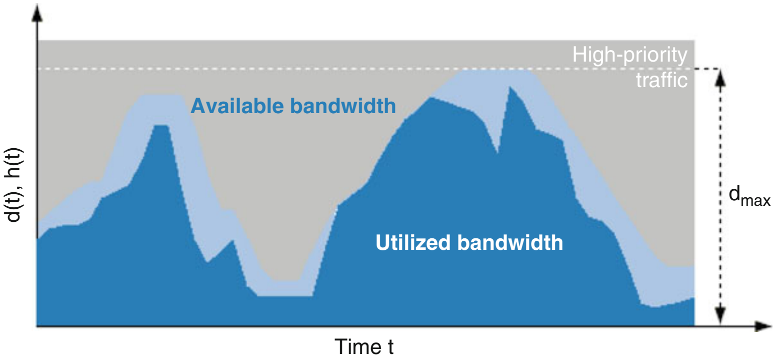


Fig. 4.2 Available and utilized bandwidth at the bottleneck link

value does not depend on the location of the bottleneck node on the data path. The quantities T_F and T_B may differ, but their sum constituting RTT remains unchanged.

The rate of packet outflow from the bottleneck link buffer depends on the bandwidth actually available at the link. The bandwidth is modeled as an *a priori* unknown, bounded function of time $d(t)$, such that

$$0 \leq d(t) \leq d_{\max}. \tag{4.1}$$

This is motivated by the fact that the controlled traffic is served within the link capacity which is left unused by the high-priority flows, such as video-on-demand or VoIP streams – see Fig. 4.2. Since the intensity of high-priority traffic may vary with time in a way unpredictable to the control process, so does the available bandwidth $d(\cdot)$ which can be treated as an external disturbance in the system. Inequalities (4.1) constitute the only constraint imposed on the available bandwidth, which means that any nonnegative, bounded function $d(t)$ is allowed. In particular, this general definition accounts for all the stochastic distributions typically analyzed in the considered problem.

If there are packets ready for transmission in the buffer, then the bandwidth actually consumed by the source $h(t)$ (the stream of packets actually leaving the node) will be equal to the available bandwidth. Otherwise, the output link is underutilized and the exploited bandwidth matches the data arrival rate at the node. Thus, we may write

$$0 \leq h(t) \leq d(t) \leq d_{\max}. \tag{4.2}$$

The relation between the available and utilized bandwidth is illustrated in Fig. 4.2.

The rate of change of the queue length at any instant of time depends on the data arrival speed and on consumed bandwidth $h(\cdot)$; thus,

$$\dot{y}(t) = u\,(t - \mathrm{RTT}) - h(t). \tag{4.3}$$

We assume that before setting up the connection, the bottleneck link buffer is empty, i.e., $y(t < 0) = 0$. Therefore, taking into account the zero initial conditions, for any $t \geq 0$, the length of the queue at the node may be expressed as

$$y(t) = \int_0^t u\,(\tau - \mathrm{RTT})\mathrm{d}\tau - \int_0^t h\,(\tau)\mathrm{d}\tau. \tag{4.4}$$

Equation (4.4) is applicable to the system in which the buffer at the bottleneck node used to store the intermediate packets is never overflowed and the incoming packets need not be dropped, i.e., Eq. (4.4) holds under the so-called infinite-buffer assumption. We will show in the next section that with an appropriately chosen control law, the queue length does not increase beyond a precisely determined finite value, which means that relation (4.4) actually holds during the whole transmission process. Note that the benefit of loss-free transmission is attractive not only from the point of view of the network users or telecommunication service providers. It is also desirable from the perspective of control system designers. Obviously, if one can guarantee that the plant remains in the linear region of operation for the whole range of disturbance $d(\cdot)$ defined in (4.1), then the linear design methods can be employed to maximize the efficiency of the fundamental control action. As a result, the analysis of the entire system dynamics can be simplified, and global stability conditions can be provided.

Assuming that the controller determines the initial rate at the time instant $t = 0$, the first packets arrive at the node at $t = \mathrm{RTT}$ and $y(t \leq \mathrm{RTT}) = 0$. Consequently, we may rewrite (4.4) as

$$y(t) = \int_0^{t-\mathrm{RTT}} u\,(\tau)\mathrm{d}\tau - \int_0^t h\,(\tau)\mathrm{d}\tau. \tag{4.5}$$

We will use this identity for the property analysis of the considered communication system, regulated by SM control algorithm formulated in the next section.

4.1.2 Sliding-Mode Flow Controller

The design of SM controllers is usually conducted in two phases. First, a switching function $\sigma(t)$, which determines the overall system dynamical properties, is selected. In the second step, in turn, a control law (typically a nonlinear one) is chosen to bring the system representative point onto the surface $\sigma(t) = 0$ and maintain the point in the vicinity of this surface afterwards despite the presence of parametric uncertainties and external disturbances. As the properties of the closed-loop system depend to a large extent on the appropriate choice of the switching function, careful attention should be given when deciding about its form at the very beginning of the design procedure. Taking into account the fact that the major obstacle in providing efficient control for the class of systems considered in this work is the delay in the feedback loop (dead time – DT), the selection of switching function should explicitly account for the effects produced by nonnegligible latency. In order to compensate DT in the analyzed communication system, we propose to use a switching function employing a state predictor. We define the following function $\sigma(t)$,

$$\sigma(t) = y_{\mathrm{D}} - y(t) - \int\limits_{t-\mathrm{RTT}}^{t} u\,(\tau)\mathrm{d}\tau, \tag{4.6}$$

and the SM control law for the considered network in the following form:

$$u(t) = \frac{1}{2}u_{\max} + \frac{1}{2}u_{\max}\ \mathrm{sgn}\ \sigma(t), \tag{4.7}$$

where $u_{\max}$ is a positive constant greater than $d_{\max}$. The sgn(x) function in (4.7) takes two values -1 and 1 depending on the value of argument x, i.e.,

$$\mathrm{sgn}(x) = \begin{cases} -1, & \text{if} \quad x \leq 0, \\ 1, & \text{if} \quad x > 0. \end{cases} \tag{4.8}$$

Therefore, the controller switches between the maximum and the minimum rate according to the value of $\sigma(t)$, generating the following command:

$$u(t) = \begin{cases} u_{\max}, & \text{if} \quad y_{\mathrm{D}} - y(t) - \int_{t-\mathrm{RTT}}^{t} u\,(\tau)\,\mathrm{d}\tau > 0, \\ 0, & \text{if} \quad y_{\mathrm{D}} - y(t) - \int_{t-\mathrm{RTT}}^{t} u\,(\tau)\,\mathrm{d}\tau \leq 0. \end{cases} \tag{4.9}$$

The proposed control strategy may also be interpreted as a combination of a dead-time compensator (DTC) with a nonlinear on-off controller. In the next section, important properties of the proposed control strategy will be defined and strictly proved.

4.1.3 Properties of the Proposed Controller

From the point of view of maximizing the network efficiency, it is convenient to apply such control algorithms which, on the one hand, will reduce (ideally eliminate) data losses and, on the other hand, will make use of as much of the available bandwidth as possible. In the following two theorems, we specify precise conditions that allow us to provide the maximum throughput in the network governed by the proposed control strategy.

Theorem 4.1. *If the proposed strategy is applied in the considered data transmission network, then the queue length in the bottleneck link buffer is always upper-bounded, i.e.,*

$$\underset{t \geq 0}{\forall} \; y(t) \leq y_{\mathrm{D}}. \tag{4.10}$$

Proof. As it has already been mentioned, for any time smaller than or equal to RTT, the queue length $y(t) = 0$. Therefore, in order to prove the theorem, it is necessary to show that $y(\cdot)$ will not exceed its demand value y_{D} at any time t greater than RTT.

Let

$$\varphi(t) = y(t) + \int_{t-\mathrm{RTT}}^{t} u(\tau)\,\mathrm{d}\tau. \tag{4.11}$$

This function represents the sum of three quantities:

1. The amount of data currently waiting in the bottleneck node queue $y(t)$.
2. The amount of "in-flight" data, i.e., the data which has already been sent by the source but has not yet arrived at the bottleneck node.
3. The amount of data which will be delivered by the source because the controller has already sent out an appropriate command signal to the source.

The integral $\int_{t-\mathrm{RTT}}^{t} u(\tau)\,\mathrm{d}\tau$ in (4.11) represents the cumulative amount of data specified in points 2 and 3. On the basis of (4.5), we may express $\varphi(t)$ as

$$\varphi(t) = \int_{0}^{t-\mathrm{RTT}} u(\tau)\,\mathrm{d}\tau - \int_{0}^{t} h(\tau)\,\mathrm{d}\tau + \int_{t-\mathrm{RTT}}^{t} u(\tau)\,\mathrm{d}\tau = \int_{0}^{t} u(\tau)\,\mathrm{d}\tau - \int_{0}^{t} h(\tau)\,\mathrm{d}\tau. \tag{4.12}$$

Taking into account relation (4.2), it can be easily concluded that this function increases only if $u(t) = u_{\max}$. Together with (4.9), this implies that

$$\underset{t \geq 0}{\forall} \; \varphi(t) \leq y_{\mathrm{D}}. \tag{4.13}$$

Since $u(t)$ is always nonnegative, one concludes that

$$y(t) = \varphi(t) - \int\limits_{t-\mathrm{RTT}}^{t} u(t)\mathrm{d}\tau \leq \varphi(t) \leq y_{\mathrm{D}}. \tag{4.14}$$

This conclusion ends the proof. □

Another desirable property of a correctly designed data flow control system is full link utilization. Notice that this condition is satisfied if the queue length is greater than zero, since then $h(t) = d(t)$. The next theorem shows how the buffer capacity should be chosen in order to ensure a strictly positive queue length and, as a consequence, full bottleneck link bandwidth utilization for all t greater than some t_0.

Theorem 4.2. *If the bottleneck link buffer capacity is greater than or equal to the demand value of the queue length y_{D}, and the following inequality holds:*

$$y_D > u_{\max}\mathrm{RTT}, \tag{4.15}$$

then for any $t > RTT$, the queue length is greater than zero.

Proof. At the initial time $t = 0$ function $\varphi(t)$, defined by (4.11), is equal to zero. Then (since by definition $u_{\max}$ is greater than $d_{\max}$) function $\varphi(\cdot)$ increases until its value becomes y_{D}. At the time instant $t = \mathrm{RTT}$, either $\varphi(\mathrm{RTT}) = y_{\mathrm{D}}$ or $\varphi(\mathrm{RTT})$ is smaller than y_D and increasing. In the first case, $\varphi(t) = y_{\mathrm{D}}$ for any time $t \geq \mathrm{RTT}$. Thus, taking into account (4.11) and (4.15), one can directly conclude that for any $t \geq \mathrm{RTT}$,

$$y(t) = \varphi(t) - \int\limits_{t-\mathrm{RTT}}^{t} u(\tau)\,\mathrm{d}\tau \geq y_{\mathrm{D}} - u_{\max}\mathrm{RTT} > 0. \tag{4.16}$$

On the other hand, the latter case implies that for any time $t \leq \mathrm{RTT}$, the flow rate $u(t) = u_{\max}$. Consequently, the number of packets accumulated in the network χ (i.e., those packets that have already been sent and those that are still required to be sent by the source because an appropriate control signal has already been generated) at the time instant $t = \mathrm{RTT}$ can be expressed as

$$\chi(\mathrm{RTT}) = u_{\max}\mathrm{RTT} \tag{4.17}$$

and

$$\varphi(\mathrm{RTT}) = \chi(\mathrm{RTT}) + y(\mathrm{RTT}) = u_{\max}\mathrm{RTT} + y(\mathrm{RTT}) \geq u_{\max}\mathrm{RTT}. \tag{4.18}$$

As it has already been mentioned, in this case, $\varphi(t)$ is increasing at $t = \mathrm{RTT}$ and nondecreasing for any $t > \mathrm{RTT}$. Consequently, for any time t greater than RTT, $\varphi(t)$ is strictly greater than $u_{\max}\mathrm{RTT}$. Consequently, it follows from (4.11) that for any $t > \mathrm{RTT}$,

$$y(t) = \varphi(t) - \int_{t-\mathrm{RTT}}^{t} u(\tau)\,\mathrm{d}\tau \geq \varphi(t) - u_{\max}\mathrm{RTT} > 0. \tag{4.19}$$

Finally, taking into account inequalities (4.16) and (4.19), one concludes that for any time t greater than RTT, the queue length is strictly positive. This ends the proof of the theorem. □

Theorem 4.2 shows that using the strategy proposed in this study, one can always ensure full link utilization provided that the bottleneck node buffer capacity is greater than $u_{\max}\mathrm{RTT}$. In fact, this property can be achieved if the buffer capacity satisfies a weaker condition, i.e., if

$$y_{\mathrm{D}} > d_{\max}\mathrm{RTT}. \tag{4.20}$$

However, in this case, the queue length is not guaranteed to be strictly positive for any time t greater than RTT but only after the elapse of some longer time since setting up the connection. Notice also that the buffer space which needs to be provided to ensure maximum throughput is specified following the worst-case uncertainty analysis. However, since the value given in (4.15) scales linearly with the maximum rate (and the available bandwidth), in the situation when the mean available bandwidth differs significantly from the maximum one, it may be convenient to substitute $u_{\max}$ in (4.15) with some positive $d_{\mathrm{L}} < d_{\max}$. In such a case, maximum throughput is no longer ensured, yet the average queue length will be reduced. In consequence, smaller buffer size will suffice to provide loss-free, efficient data transfer.

4.1.4 Simulation Results

We verify the properties of the described control strategy in a series of simulation tests for the connection with $\mathrm{RTT} = 100$ ms. The maximum bandwidth $d_{\max} = 1{,}000$ packets/s and the maximum source sending rate $u_{\max} = 1{,}100$ packets/s $= 1.1d_{\max}$. In the tests, while assessing throughput, we take into account the actual packet losses due to buffer overflow and the "virtual" losses from missed transfer opportunities occurring when part of the available bandwidth is left unused due to an empty buffer. Hence, the maximum throughput can be achieved only when no packet needs to be discarded due to buffer overflow, and there is always sufficient number of packets in the buffer to keep the outgoing link fully occupied with data transfer.

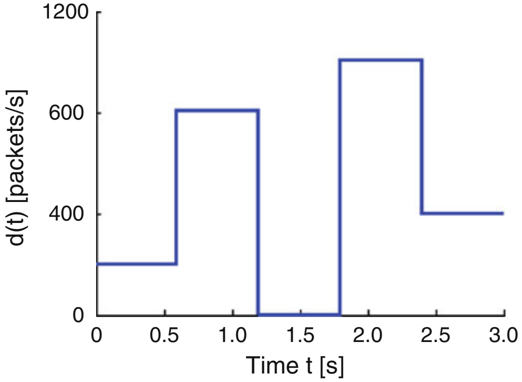


Fig. 4.3 Available bandwidth – staircase pattern

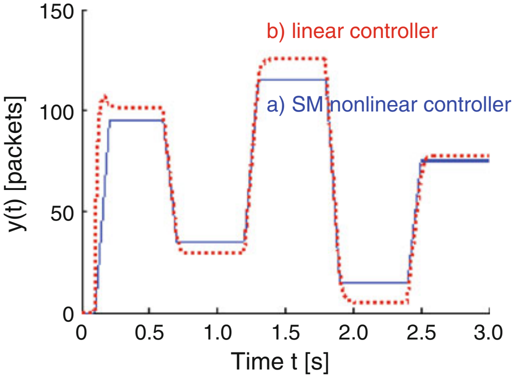


Fig. 4.4 Packet queue length: *a* nonlinear controller (4.7) ($y_D = 115$ packets) and *b* linear controller [17] (gain = 50 s^{-1}, $y_D = 125$ packets)

Test 1. In the first simulation scenario, we test the controller response to the most adverse networking conditions, i.e., sudden bandwidth changes of large amplitude. The bandwidth pattern used in the test is shown in Fig. 4.3.

In order to avoid buffer overflow and to ensure full bottleneck link utilization, the demand value of the queue length must be greater than $y_{D\,\min} = 110$ packets. Therefore, $y_D = 115$ packets was selected. The queue length in the buffer of the bottleneck link is presented in Fig. 4.4. It can be seen from that figure (curve a) that the demand value of the queue length is never overrun and that the queue length $y(t)$ is strictly positive for any time greater than 100 ms. These two properties imply no buffer overflow and full bottleneck link utilization. Furthermore, it is worth pointing out that for a similar (but linear) scheme in the existing literature [17] using the SP, the smallest demand queue length that ensures full bandwidth utilization and no packet loss is always greater than the same value $y_{D\,\min} = 110$ packets for the strategy proposed in this work. Setting $y_D = 125 > 120$ packets as dictated in [17] (with the gain constant adjusted to 50 s^{-1}), we obtain curve b in Fig. 4.4. It is clear from the graph that our strategy provides faster transient performance and consequently requires smaller bottleneck buffer capacity than the linear strategy.

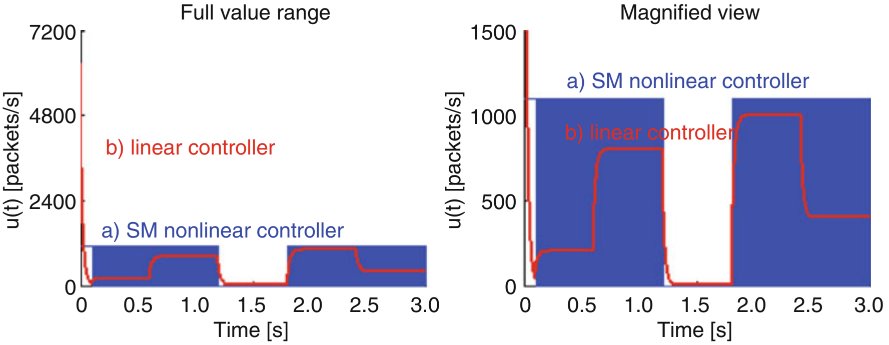
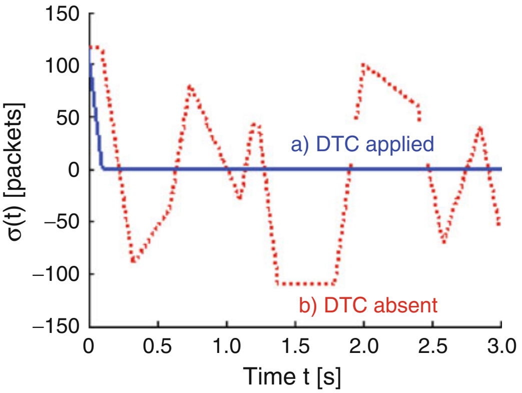


Fig. 4.5 Transmission rate: *a* nonlinear controller (4.7) ($y_\mathrm{D} = 115$ packets) and *b* linear controller [17] (gain $= 50\ \mathrm{s}^{-1}$, $y_\mathrm{D} = 125$ packets)

Fig. 4.6 Switching variable: *a* with DTC applied and *b* with DTC absent



The transmission rate of the source is illustrated in Fig. 4.5. Comparing the designed controller (4.7) with the linear scheme, we can notice that our strategy requires frequent switching of the control signal, which may be difficult to realize by certain transmitters. Therefore, our SM controller, which provides faster convergence of the queue length to steady-state values and requires smaller buffer space to eliminate losses and achieve 100% of bandwidth utilization, is primarily intended for the networks in which the buffer capacity limitations are of major concern and fast transmitters are available.

The evolution of the sliding variable is shown in Fig. 4.6, curve a. We can see from the plot that the sliding surface $\sigma(t) = 0$ is attained in finite time, and subsequently, the system dynamics is confined to that plane even though the matching conditions are not satisfied for the external disturbance $d(\cdot)$. The stability of the sliding motion is achieved due to the special choice of the switching function employing the delay compensating term $\int_{t-\mathrm{RTT}}^{t} u(\tau)\,\mathrm{d}\tau$. To better illustrate the importance of this term, we run the simulation for controller (4.7) with a different

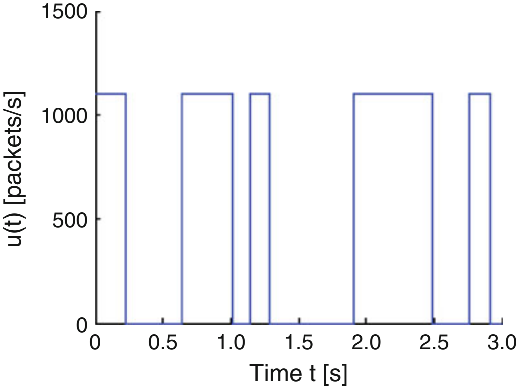


Fig. 4.7 Transmission rate established by controller (4.7) with switching function $\sigma(t) = y_\mathrm{D} - y(t)$

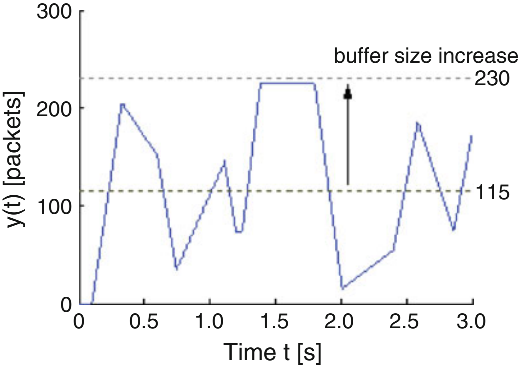


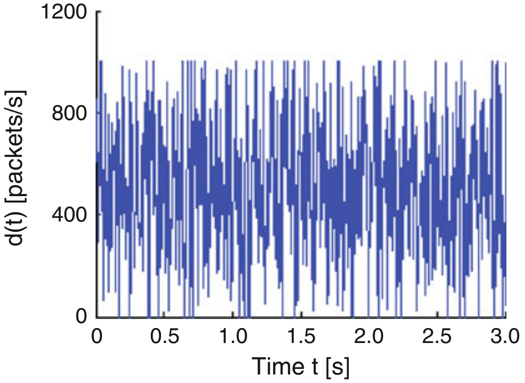
Fig. 4.8 Packet queue length when controller (4.7) operates with function $\sigma(t) = y_\mathrm{D} - y(t)$

switching function with the integral removed from Eq. (4.6), i.e., $\sigma(t) = y_\mathrm{D} - y(t)$. The plot of $\sigma(\cdot)$ is given in Fig. 4.6, curve b, the established control signal is illustrated in Fig. 4.7, and the resulting output variable in Fig. 4.8.

It is evident from Fig. 4.6, curve b, that the sliding motion cannot be ensured in the presence of the external disturbance when the delay compensating term is removed from the switching function. However, it comes from the analysis of the plots in Figs. 4.7 and 4.8 that the closed-loop system stability in the Lyapunov sense is maintained. This is attributed to the nonlinearity incorporated in controller (4.7), which limits the generated transmission rate to the two-value set $\{0, u_\mathrm{max}\}$. The queue length evolution depicted in Fig. 4.8 shows that when the effects of delay are neglected in the design of the switching function, the buffer capacity requirements grow extensively. In order to eliminate packet losses, the buffer size should be enlarged from y_D to $2y_\mathrm{D}$, which obviously decreases the efficiency of memory allocation policy at the network node.

Test 2. In the second simulation scenario, we test the controller performance in the communication system with highly variable stochastic bandwidth. The bandwidth

Fig. 4.9 Available bandwidth – normal distribution D_{norm} (500, 300)



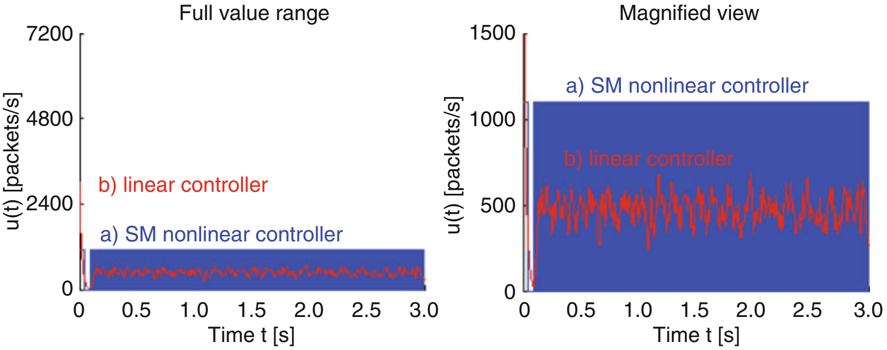


Fig. 4.10 Transmission rate: *a* nonlinear controller (4.7) ($y_D = 57$ packets) and *b* linear controller [17] (gain = 50 s^{-1}, $y_D = 62$ packets)

pattern evolving according to the normal distribution $D_{\text{norm}}(\mu, \delta)$ with mean $d_\mu = 500$ packets/s and standard deviation $d_\delta = 300$ packets/s used in the test is illustrated in Fig. 4.9.

Since the mean available bandwidth significantly differs from the maximum value, instead of $d_{\max}$, we use $d_L = 500$ packets/s in the formula for the demand queue length (4.15) and set y_D as $57 > 55$ packets. For fair comparison, we also appropriately reduce the demand queue length for strategy [17] setting $y_D = 62$ packets $> d_L(\text{RTT} + \text{gain}^{-1}) = 500{\cdot}(0.1 + 1/50) = 60$ packets. The transmission rate is presented in Fig. 4.10 and the queue length at the bottleneck node in Fig. 4.11. It follows from the graphs that while both strategies guarantee loss-free transmission (buffer space is not exceeded), the nonlinear controller exhibits faster response and imposes smaller memory requirements than the linear scheme. Again, the benefits of smaller buffer space are achieved provided that a fast transmitter is available at the source – the one able to follow the high-frequency switching of the control signal.

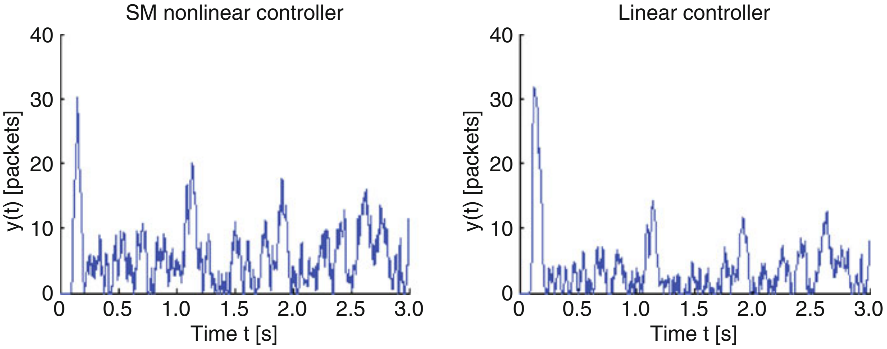


Fig. 4.11 Packet queue length: nonlinear controller (4.7) ($y_\mathrm{D} = 57$ packets) and linear controller [17] (gain = 50 s^{-1}, $y_\mathrm{D} = 62$ packets)

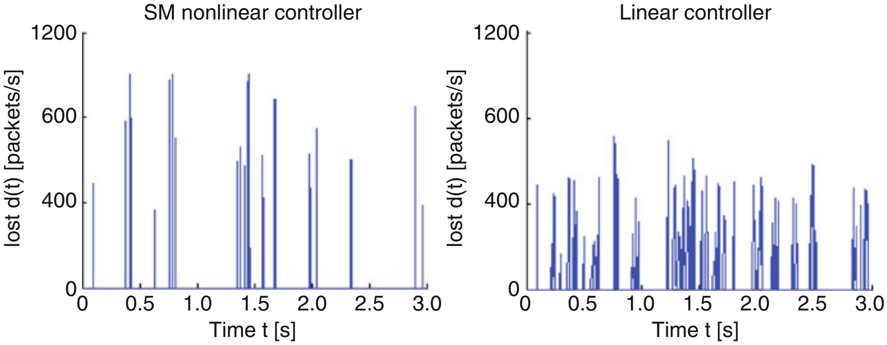


Fig. 4.12 Missed opportunities for transferring data

Although it is no longer guaranteed to keep the link bandwidth fully utilized (the bandwidth utilization decreases to 98% in the case of nonlinear controller (4.7) and 94% in the case of linear one [17]), we obtain 50% buffer capacity savings. The missed opportunities for transferring the data due to empty buffer are shown in Fig. 4.12.

The switching variable is depicted in Fig. 4.13. The surface $\sigma(t) = 0$ is attained in finite time, and afterwards the system remains in sliding motion.

4.2 Flow Control in a Multisource Network

In this section, we extend the SMC design from the single flow case discussed in Sect. 4.1 to the network supporting multiple connections passing through the bottleneck link. First, we describe the multisource network model. Then, we define

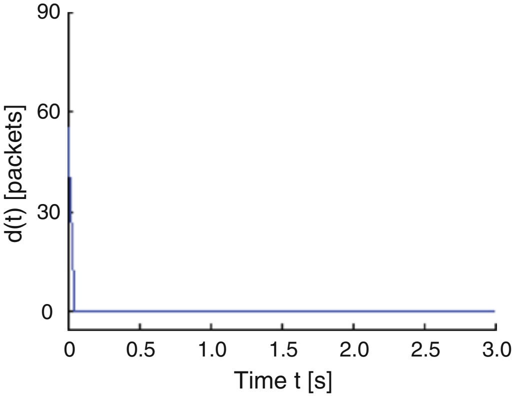


Fig. 4.13 Switching variable for controller (4.7)

a new switching function which takes into account multiple flows characterized by different delays and apply the already defined nonlinear control law. Next, similarly as in the case of the single connection, the control law properties are stated as two theorems and are proved analytically. The properties are illustrated in a simulation scenario given in the last part of this section.

4.2.1 Network Model

We analyze the situation of multiple flows passing through the bottleneck link of a network node. In the general case, m virtual connections will pass through that link, each originating at a different source. The decision whether a new flow is accepted by the network is made by the admission control procedure in the connection setup phase. Consequently, the node can keep track of the number of connections served by the output link. It is assumed here that only a single node is the bottleneck for the considered set of VCs – a detailed discussion on the algorithm extensions to multibottleneck topologies will be covered in the further part of this work – in Chaps. 6 and 7.

For connection p ($p = 1, 2, \ldots, m$), we can state

$$\mathrm{RTT}_p = T_{\mathrm{F}}^p + T_{\mathrm{B}}^p, \tag{4.21}$$

where T_{F}^p is the forward delay (the delay on the route from source p to the bottleneck node) and T_{B}^p is the backward delay (the delay on the route from the bottleneck node to the destination and back to source p). The VCs are numbered in such a way that

$$\mathrm{RTT}_1 \leq \mathrm{RTT}_2 \leq \ldots \leq \mathrm{RTT}_{m-1} \leq \mathrm{RTT}_m. \tag{4.22}$$

Further in the text, the overall transmission rate generated by the controller at time instant t is denoted by $u(t)$ and the individual source rate by $u_p(t)$. We assume in this section that the total rate $u(t)$ generated by the controller implemented at the node is equally distributed between all the sources contributing to the bottleneck queue. Other allocation strategies which can be realized within the chosen framework, for example, max-min [13] or proportionally [16] fair, will be discussed in Chap. 6. When equal rate partitioning is applied, the rate of source p

$$u_p(t) = \frac{1}{m} u\left(t - T_{\mathrm{B}}^{p}\right). \tag{4.23}$$

It is assumed that the controller does not assign a positive transmission rate for $t < 0$, i.e., $u(t < 0) = 0$.

The queue length dynamics depends on the arrival rate of packets from all the sources and on the utilized bandwidth $h(\cdot)$ that is subject to constraint (4.2). The queue length dynamics can be expressed by the following equation:

$$\dot{y}(t) = \sum_{p=1}^{m} u_p\left(t - T_{\mathrm{F}}^{p}\right) - h(t). \tag{4.24}$$

Consequently, $y(t)$ may be calculated as

$$y(t) = \sum_{p=1}^{m} \int_{0}^{t} u_p\left(\tau - T_{\mathrm{F}}^{p}\right) \mathrm{d}\tau - \int_{0}^{t} h\left(\tau\right) \mathrm{d}\tau. \tag{4.25}$$

Similarly as in the previous section, it is assumed that initially the buffer is empty, i.e., $y(0) = 0$. Taking into account (4.23), we may rewrite the formula that expresses the queue length at time instant t as

$$y(t) = \sum_{p=1}^{m} \int_{0}^{t} \frac{1}{m} u\left(\tau - \mathrm{RTT}_p\right) \mathrm{d}\tau - \int_{0}^{t} h\left(\tau\right) \mathrm{d}\tau, \tag{4.26}$$

which after considering the initial condition $u(t < 0) = 0$ reduces to

$$y(t) = \sum_{p=1}^{m} \int_{-\mathrm{RTT}_p}^{t-\mathrm{RTT}_p} \frac{1}{m} u\left(\tau\right) \mathrm{d}\tau - \int_{0}^{t} h\left(\tau\right) \mathrm{d}\tau = \sum_{p=1}^{m} \int_{0}^{t-\mathrm{RTT}_p} \frac{1}{m} u\left(\tau\right) \mathrm{d}\tau - \int_{0}^{t} h\left(\tau\right) \mathrm{d}\tau. \tag{4.27}$$

The schematic diagram of the considered network is shown in Fig. 4.14.

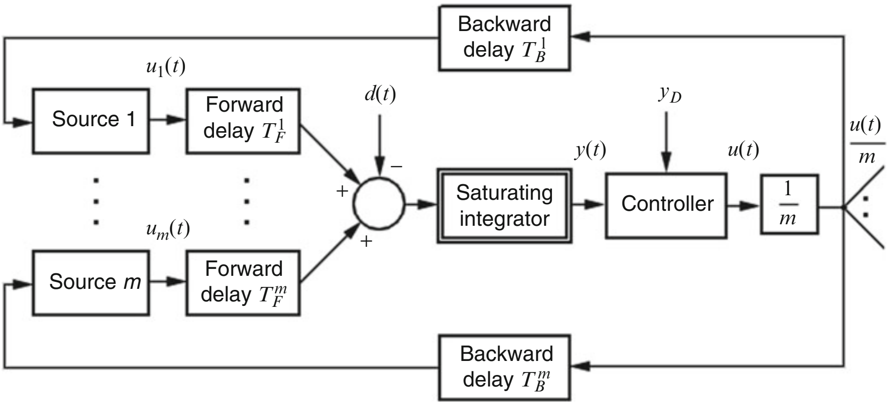


Fig. 4.14 Network model – multiple virtual circuits

4.2.2 Sliding-Mode Flow Controller

In order to account for the effects of multiple flows feeding the packet queue, we need to redefine the switching variable $\sigma(t)$ proposed in Sect. 4.1. The new function takes the following form:

$$\sigma(t) = y_{\mathrm{D}} - y(t) - \sum_{p=1}^{m} \frac{1}{m} \int\limits_{t-\mathrm{RTT}_p}^{t} u(\tau)\mathrm{d}\tau. \tag{4.28}$$

The control law for the multisource network is the same as in the single-source scenario, i.e.,

$$u(t) = \frac{1}{2}u_{\max} + \frac{1}{2}u_{\max}\,\mathrm{sgn}\,\sigma(t);$$

however, now, the rate calculation is based on $\sigma(t)$ defined by (4.28). The controller switches between the maximum and the minimum rate according to the value of $\sigma(t)$ generating the command:

$$u(t) = \begin{cases} u_{\max}, \text{ if } & y_{\mathrm{D}} - y(t) - \sum\limits_{p=1}^{m} \frac{1}{m} \int\limits_{t-\mathrm{RTT}_p}^{t} u(\tau)\mathrm{d}\tau > 0, \\ 0, \quad \text{ if } & y_{\mathrm{D}} - y(t) - \sum\limits_{p=1}^{m} \frac{1}{m} \int\limits_{t-\mathrm{RTT}_p}^{t} u(\tau)\mathrm{d}\tau \le 0. \end{cases} \tag{4.29}$$

In the next section, important properties of the proposed control strategy related to handling the flow of data will be defined and strictly proved.

4.2.3 Properties of the Proposed Controller

The properties of the proposed control strategy are stated in two theorems. The first proposition shows that the queue length in the bottleneck link buffer never exceeds the demand value no matter the bandwidth variations. This means that if the buffer capacity is selected at least equal to y_{D}, then packet losses are eliminated. The second theorem specifies how to select y_{D} to meet the condition $y(t) > 0$, which ensures that all of the available bandwidth is used for data transfer.

Theorem 4.3. *If strategy (4.29) is applied to control the flow of data in the considered network, then the queue length in the bottleneck link buffer is always upper-bounded by its demand value y_D.*

Proof. It follows from the specified initial conditions that for any time instant smaller than or equal to RTT_1, the queue length equals zero. Hence, it is sufficient to show that the queue length will not exceed its demand value y_{D} at any time t greater than RTT_1.

We define an auxiliary function

$$\varphi(t) = y(t) + \sum_{p=1}^{m} \frac{1}{m} \int\limits_{t-\mathrm{RTT}_p}^{t} u(\tau)\,\mathrm{d}\tau, \tag{4.30}$$

which represents the sum of packets currently waiting in the bottleneck buffer and the number of in-flight ones. On the basis of (4.27), we may express $\varphi(t)$ as

$$\begin{aligned}\varphi(t) &= \sum_{p=1}^{m} \frac{1}{m} \int\limits_{0}^{t-\mathrm{RTT}_p} u(\tau)\,\mathrm{d}\tau - \int\limits_{0}^{t} h(\tau)\,\mathrm{d}\tau + \sum_{p=1}^{m} \frac{1}{m} \int\limits_{t-\mathrm{RTT}_p}^{t} u(\tau)\mathrm{d}\tau \\ &= \sum_{p=1}^{m} \frac{1}{m} \int\limits_{0}^{t} u(\tau)\,\mathrm{d}\tau - \int\limits_{0}^{t} h(\tau)\,\mathrm{d}\tau = \int\limits_{0}^{t} u(\tau)\mathrm{d}\tau - \int\limits_{0}^{t} h(\tau)\mathrm{d}\tau.\end{aligned} \tag{4.31}$$

It follows from (4.2) that this function increases only if $u(t) = u_{\max}$. Together with (4.29), this implies that $\forall t \geq 0, \varphi(t) \leq y_{\mathrm{D}}$. Since $u(t)$ is nonnegative, we conclude that

$$y(t) = \varphi(t) - \sum_{p=1}^{m} \frac{1}{m} \int\limits_{t-\mathrm{RTT}_p}^{t} u(\tau)\,\mathrm{d}\tau \leq \varphi(t) \leq y_{\mathrm{D}}. \tag{4.32}$$

This ends the proof. □

The next theorem specifies how the demand queue length should be selected so that all of the available bandwidth at the node output interface is used for data transfer.

Theorem 4.4. *If the bottleneck link buffer capacity is greater than or equal to the demand value of the queue length y_{D}, and the following inequality holds:*

$$y_{\mathrm{D}} > u_{\max}\left(\frac{1}{m}\sum_{p=1}^{m}\mathrm{RTT}_p\right), \tag{4.33}$$

then for any $t > RTT_m$, the queue length is strictly greater than zero.

Proof. At the initial time $t = 0$ function $\varphi(\cdot)$, defined by (4.30), is equal to zero. Then, since $u_{\max} > d_{\max}$, $\varphi(\cdot)$ increases until its value becomes y_D. At the time instant $t = \mathrm{RTT}_m$, either $\varphi(\mathrm{RTT}_m) = y_{\mathrm{D}}$ or $\varphi(\mathrm{RTT}_m)$ is smaller than y_{D} and increasing. In the first case, $\varphi(t) = y_{\mathrm{D}}$ for any time $t \geq \mathrm{RTT}_m$. Thus, taking into account (4.30) and (4.33), we directly conclude that for any $t \geq \mathrm{RTT}_m$,

$$y(t) = \varphi(t) - \sum_{p=1}^{m}\frac{1}{m}\int_{t-\mathrm{RTT}_p}^{t} u(t)\mathrm{d}\tau \geq y_{\mathrm{D}} - u_{\max}\left(\frac{1}{m}\sum_{p=1}^{m}\mathrm{RTT}_p\right) > 0. \tag{4.34}$$

The latter case implies that for any time $t \leq \mathrm{RTT}_m$, the flow rate $u(t) = u_{\max}$. Consequently, the number of packets accumulated in the network χ (i.e., those packets that have already been sent and those that will be sent by the sources due to the already issued command) at the time instant $t = \mathrm{RTT}_m$ can be expressed as

$$\chi\left(\mathrm{RTT}_m\right) = \left[\frac{1}{m}\left(\mathrm{RTT}_m - \mathrm{RTT}_{m-1}\right) + \frac{2}{m}\left(\mathrm{RTT}_{m-1} - \mathrm{RTT}_{m-2}\right)\right.$$
$$\left. + \cdots + \frac{m-1}{m}\left(\mathrm{RTT}_2 - \mathrm{RTT}_1\right) + \mathrm{RTT}_1\right] u_{\max} = \left(\frac{1}{m}\sum_{p=1}^{m}\mathrm{RTT}_p\right) u_{\max} \tag{4.35}$$

and

$$\begin{aligned}\varphi\left(\mathrm{RTT}_m\right) &= \chi\left(\mathrm{RTT}_m\right) + y\left(\mathrm{RTT}_m\right)\\ &= \frac{u_{\max}}{m}\sum_{p=1}^{m}\mathrm{RTT}_p + y\left(\mathrm{RTT}_m\right) \geq \frac{u_{\max}}{m}\sum_{p=1}^{m}\mathrm{RTT}_p.\end{aligned} \tag{4.36}$$

In this case, $\varphi(t)$ is increasing at $t = \mathrm{RTT}_m$ and nondecreasing for any $t > \mathrm{RTT}_m$. Consequently, for any time t greater than RTT_m, $\varphi(t)$ is strictly greater than $u_{\max}\sum_{p=1}^{m}\mathrm{RTT}_p/m$. Then, it follows from (4.30) that for any $t > \mathrm{RTT}_m$,

$$y(t) = \varphi(t) - \sum_{p=1}^{n}\frac{1}{m}\int_{t-\mathrm{RTT}_p}^{t} u(\tau)\,\mathrm{d}\tau \geq \varphi(t) - \frac{u_{\max}}{m}\sum_{p=1}^{m}\mathrm{RTT}_p > 0. \tag{4.37}$$

Finally, using inequalities (4.34) and (4.37), we may conclude that for any time t greater than RTT_m, the queue length is strictly positive. This ends the proof. □

Theorem 4.4 shows that strategy (4.29) ensures full link utilization in the network supporting multiple flows provided that the buffer capacity at the bottleneck node is greater than $u_{\max} \sum_{p=1}^{m} \mathrm{RTT}_p \,/m$. Similarly as in the case of the single connection, this property can be achieved if the buffer capacity satisfies a weaker condition, i.e.,

$$y_{\mathrm{D}} > d_{\max} \left(\frac{1}{m} \sum_{p=1}^{m} \mathrm{RTT}_p \right). \tag{4.38}$$

However, in such a case, a longer time than RTT_m must elapse since setting up the connection before the full bandwidth utilization takes place. Moreover, since the value given in (4.33) scales linearly with the maximum rate, in the situation when the mean available bandwidth differs much from the maximum one, it may be advisable to replace $u_{\max}$ by a smaller value $d_{\mathrm{L}} < d_{\max}$. In such a case, the maximum throughput cannot be guaranteed for arbitrary bounded bandwidth, but the average queue length will be reduced. As a result, the memory allocation policy will be more efficient.

In the next section, we study the properties of control law (4.29) in a series of simulation tests.

4.2.4 Simulation Results

In order to verify the performance of control strategy (4.29), we consider the model of a wide area network consisting of four VCs ($m = 4$) passing through the bottleneck node. The connections are characterized by the delays equal to 30, 70, 80, and 120 ms. The maximum available bandwidth $d_{\max}$ is set as 10,000 packets/s, and the upper bound of the overall source rate $u_{\max}$ is adjusted to 11,000 packets/s $= 1.1 d_{\max}$. We test the controller performance in response to two different bandwidth patterns. In Test 1, we focus on studying the response to abrupt changes in the networking conditions, whereas in the second scenario, we verify the controller operation in the presence of a stochastic bandwidth pattern.

Test 1. The function representing the available bandwidth used in Test 1 is depicted in Fig. 4.15. According to Theorem 4.4, the queue length will be greater than zero, thus ensuring full bandwidth usage, if the demand value of the queue length $y_{\mathrm{D}} > 750$ packets. In this simulation, $y_{\mathrm{D}} = 760$ packets is assumed, which according to Theorem 4.3 represents also the buffer capacity necessary to ensure no data loss.

The queue length $y(\cdot)$ resulting from the application of the control scheme described by relation (4.29) is shown in Fig. 4.16, curve a. It can be seen from

Fig. 4.15 Available bandwidth

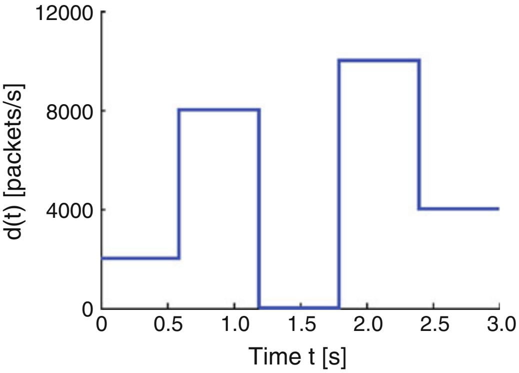
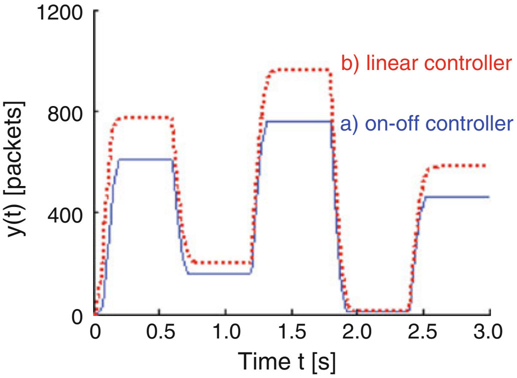


Fig. 4.16 Packet queue length: *a* SM on-off controller (4.29) ($y_D = 760$ packets) and *b* linear controller [18] (gain $= 50\ \mathrm{s}^{-1}$, $y_D = 960$ packets)



the figure that the queue length does not exceed the value of 760 packets and never drops to zero. These two properties imply no buffer overflow and full utilization of the bottleneck link bandwidth.

For the purpose of comparison, we repeat the test for a similar (but linear) controller based on the SP reported in [18]. We set the gain for controller [18] equal to $50\ \mathrm{s}^{-1}$ and y_D bigger than $d_{\max}\left(\sum_{p=1}^{m} \mathrm{RTT}_p\,/m\ + \mathrm{gain}^{-1}\right) = 10{,}000{\cdot}(0.075 + 0.02) = 950$ as 960 packets. The queue length $y(\cdot)$ resulting from the application of the linear controller is presented in Fig. 4.17, curve b. It is clear from the plots in Fig. 4.16 that the SM strategy performs better in terms of the queue length management, as it results in faster convergence to steady-state values and smaller buffer space. However, this is achieved at the expense of higher degree of oscillations of the control signal which are related to switching around the sliding plane. The overall transfer speed generated by the controller is illustrated in Fig. 4.17 and the switching variable in Fig. 4.18, curve a. We can see from the graph in Fig. 4.18 that in the case of multiple connections characterized by different delays, similarly as in the single flow scenario analyzed in Sect. 4.1, the switching surface $\sigma(t) = 0$ is attained in finite time and afterwards the system representative point does not leave the plane.

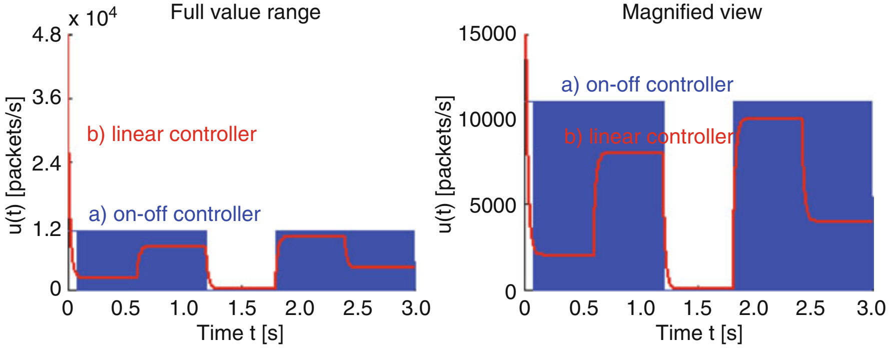
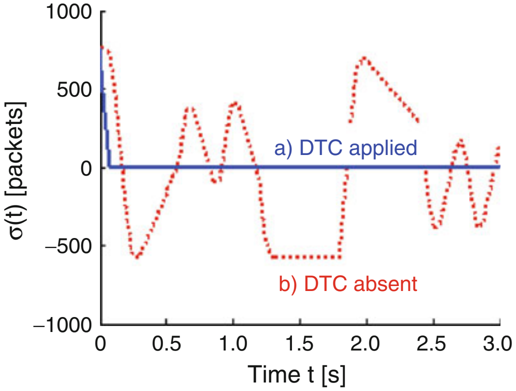


Fig. 4.17 Transmission rate: *a* SM on-off controller (4.29) ($y_D = 760$ packets) and *b* linear controller [18] (gain = 50 s^{-1}, $y_D = 960$ packets)

Fig. 4.18 Switching variable: *a* with DTC applied and *b* with DTC absent



The delay compensating term $\sum_{p=1}^{m} \int_{t-\mathrm{RTT}_p}^{t} u(\tau)\, \mathrm{d}\tau$ in the structure of the switching function has a decisive impact on the existence of the sliding motion. It is evident from the plot in Fig. 4.18, curve b, depicting the evolution of $\sigma(t) = y_D - y(t)$ that sliding motion cannot be guaranteed without appropriate handling of the delay effects. However, even though the sliding phase is not reached when the compensator is removed from (4.28), Figs. 4.19 and 4.20 illustrating the control signal and the output variable indicate that Lyapunov stability is maintained. Nevertheless, in order to eliminate packet losses, strategy (4.29) with the plane $\sigma(t) = y_D - y(t) = 0$ requires nearly doubled buffer capacity as compared to the case when $\sigma(t)$ given by (4.28) is applied. The increased buffer requirements are illustrated in Fig. 4.20.

Test 2. In the second simulation scenario, we verify the controller performance in the presence of highly variable stochastic bandwidth following the normal distribution with mean $d_\mu = 5{,}000$ packets/s and standard deviation $d_\delta = 3{,}000$

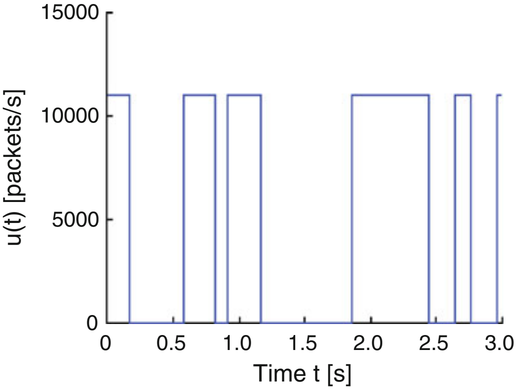


Fig. 4.19 Transmission rate established by controller (4.29) with switching function $\sigma(t) = y_D - y(t)$

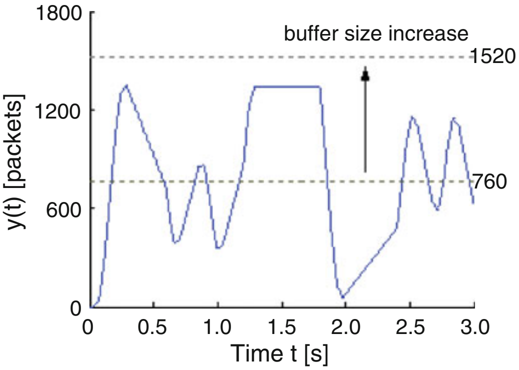


Fig. 4.20 Packet queue length when controller (4.29) operates with switching function $\sigma(t) = y_D - y(t)$

packets/s. The function $d(\cdot)$ used in the test is illustrated in Fig. 4.21. The mean bandwidth is much lower than the maximum value; therefore, we use $d_L = 5{,}000$ packets/s to calculate the demand queue length instead of $u_{\max}$ in (4.33) and set y_D as $380 > 375$ packets. We also appropriately reduce the demand queue length for strategy [18] setting $y_D = 480 > 475$ packets. The transmission rate is presented in Fig. 4.22 and the packet queue length in Fig. 4.23. It follows from the queue length evolution that both strategies guarantee that the assigned buffer space is not exceeded and packet losses are avoided. The nonlinear controller exhibits faster response and has smaller memory requirements than the linear scheme, which is obtained at the expense of the control signal subject to high-frequency switching.

Since the demand queue length was selected smaller than indicated by Theorem 4.4, the maximum throughput is no longer ensured. The bandwidth utilization approaches 92% in the case of the SM on-off controller (4.29) and 93% in the case of the linear one [18]. The bandwidth usage degradation comes at a propitious trade-off of reduced buffer space (buffer capacity is halved as compared to the fully robust case analyzed in Test 1). The missed opportunities for transferring the data due to empty buffer are shown in Fig. 4.24.

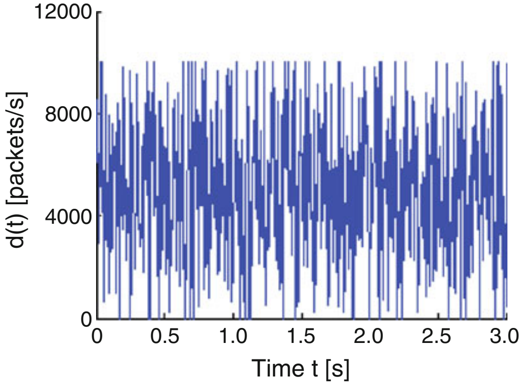


Fig. 4.21 Available bandwidth following normal distribution $D_{\text{norm}}(5{,}000, 3{,}000)$

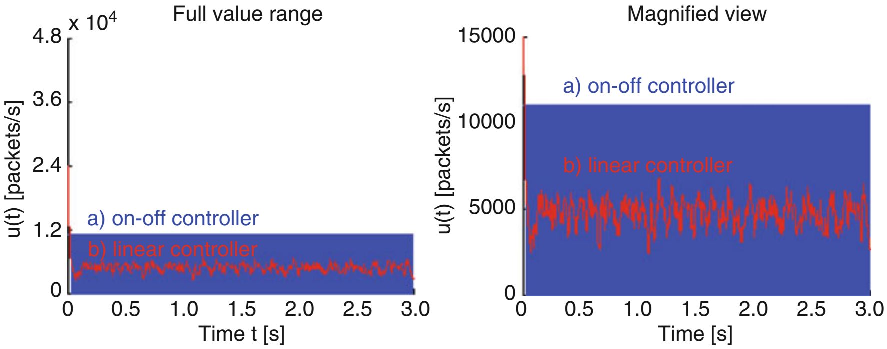


Fig. 4.22 Transmission rate: *a* SM on-off controller (4.29) ($y_D = 380$ packets) and *b* linear controller [18] (gain = 50 s^{-1}, $y_D = 480$ packets)

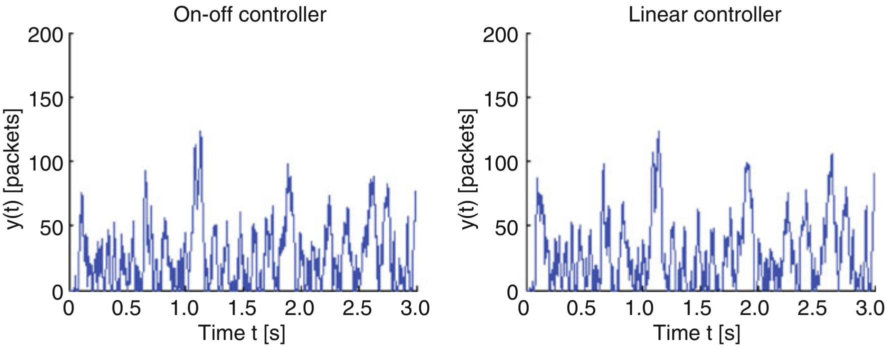


Fig. 4.23 Packet queue length: SM on-off controller (4.29) ($y_D = 380$ packets), linear controller [18] (gain = 50 s^{-1}, $y_D = 480$ packets)

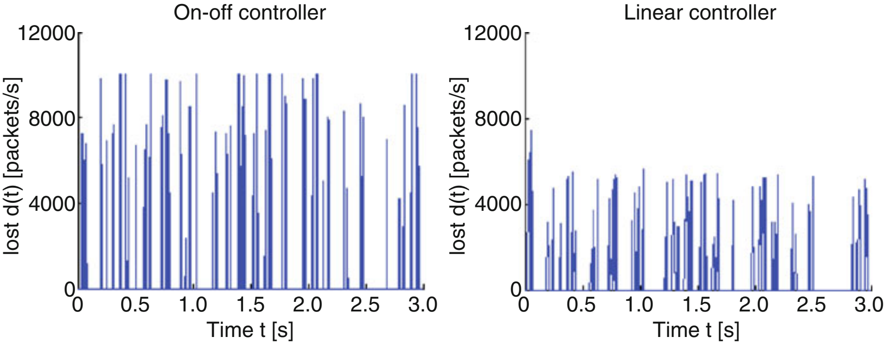
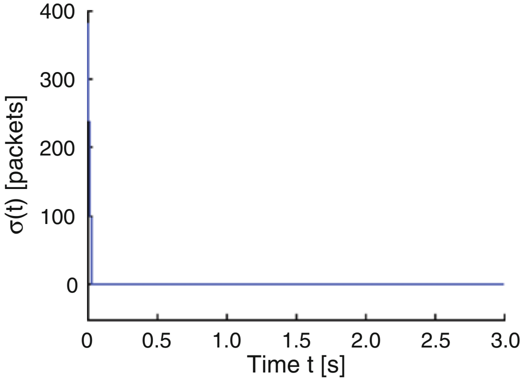


Fig. 4.24 Missed opportunities for transferring data

Fig. 4.25 Switching variable for controller (4.29)



The switching variable is shown in Fig. 4.25. The surface $\sigma(t) = 0$ is attained in finite time, and the system representative point remains on the surface for subsequent time instants implying ideal sliding motion.

4.3 Chapter Summary

In this chapter, we analyzed the basic process of transferring data in a communication network. In the first part of the chapter, we set the basis for the analysis of networking phenomena related to flow regulation in data transmission networks and solved the problem of efficient rate control of the data stream in a single connection. The controller, designed using the theory of SMC, ensures that packet losses are eliminated and all of the available bandwidth is used for the transmission of data. This is of particular importance for applications expecting low loss rate, such as banking transactions, or stock exchange feeds. The controller performance was compared with an outstanding linear controller developed for a similar network model [17], and it was shown that the nonlinear strategy presented in this work can guarantee the maximum throughput with smaller buffer requirements yet with a trade-off in the smoothness of rate transitions. The favorable properties of the proposed SM controller are mainly attributed to the appropriate handling of the effects of delay in the feedback loop by means of a state predictor included in the switching function. As far as the theoretical background is considered, the employed mechanism may be also analyzed in a broader context of the works on SM observers and controllers for systems with delay, for instance [5, 21, 23, 24], or the development of DT compensators for a general class of continuous-time systems [19]. After all, as discussed in [20], all the stabilizing controllers for the systems with delay should contain an observer-predictor structure. In its basic form, the structure resembles the SP, which is used to provide an estimate of the rational part of the controlled plant.

In the second part of the chapter, the problem of data flow control was extended to a more general case of m flows sharing the (single) bottleneck link. It was demonstrated that the features of the controller depend on the appropriate choice of the switching function which should account for multiple delays in the feedback path. The designed SM controller was shown to provide the maximum throughput in the considered network with smaller buffer requirements than the very successful SP-based strategy proposed in the past [18]. However, similarly as in the case of single flow data transfer, this is achieved at the expense of more abrupt changes of source transmission rate.

The primary drawback of the strategies described in this chapter (and the other controllers developed within a similar framework such as [17] and [18]) is the necessity of providing the information about the transfer rate to the data sources continuously in time. In real networks, the feedback information about the current condition of the transmission system is accessible at the sources only at discrete time instants (e.g., in ATM networks at the instant of RM cell arrival and in TCP networks upon the reception of an acknowledgement or timeout expiry). Therefore, in the further part of this work, we direct our attention towards discrete-time (Chaps. 5 and 6) and sampled-data (Chap. 7) systems, which explicitly account for the discrete nature of fundamental networking phenomena. However, it should be stressed that fluid-flow models, the example of which is the class of systems considered in this chapter, give a very good approximation of the essential network dynamics in many traffic scenarios [8]. As shown in the past, they may be successfully applied in the controller design for traffic regulation and serve as a good reference for studying the macroscopic behavior of even complex communication systems.

References

1. Altman E, Basar T (1995) Optimal rate control for high speed telecommunication networks. In: Proceedings of the 34th IEEE conference on decision and control, vol 2, New Orleans, USA, pp 1389–1394

2. Altman E, Basar T (1998) Multiuser rate-based flow control. IEEE Trans Commun 46:940–949
3. Altman E, Basar T, Srikant R (1999) Congestion control as a stochastic control problem with action delays. Automatica 35:1937–1950
4. Bartoszewicz A, Kaynak O, Utkin VI (eds) (2008) Sliding mode control in industrial applications. Special Issue: IEEE Trans Ind Electron 55:3806–4074
5. Basin M, Fridman L, Rodriguez-González J, Acosta P (2003) Optimal and robust sliding mode control for linear systems with multiple time delays in control input. Asian J Control 5:557–567
6. Benmohamed L, Meerkov SM (1993) Feedback control of congestion in packet switching networks: the case of a single congested node. IEEE/ACM Trans Netw 1:693–708
7. Blanchini F, Lo Cigno R, Tempo R (2002) Robust rate control for integrated services packet networks. IEEE/ACM Trans Netw 10:644–652
8. Chatté F, Ducourthial B, Nace D, Niculescu SI (2003) Fluid modelling of packet switching networks: perspectives for congestion control. Int J Syst Sci 34:585–597
9. Draženović B (1969) The invariance conditions in variable structure systems. Automatica 5:287–295
10. Imer OC, Compans S, Basar T, Srikant R (2001) Available Bit Rate congestion control in ATM networks. IEEE Control Syst Mag 21:38–56
11. Izmailov R (1995) Adaptive feedback control algorithms for large data transfers in high-speed networks. IEEE Trans Autom Control 40:1469–1471
12. Izmailov R (1996) Analysis and optimization of feedback control algorithms for data transfer in high-speed networks. SIAM J Control Optim 34:1767–1780
13. Jaffe JM (1981) Bottleneck flow control. IEEE Trans Commun 29:954–962
14. Katabi D, Handley M, Rohrs ChE (2002) Congestion control for high bandwidth-delay product networks. In: Proceedings of ACM SIGCOMM, Pittsburgh, USA, pp 89–102
15. Kaynak O, Bartoszewicz A, Utkin VI (eds) (2009) Sliding mode control in industrial applications. Special Issue: IEEE Trans Ind Elect 56:3271–3784
16. Kelly FP, Maulloo AK, Tan DKH (1998) Rate control for communication networks: shadow prices, proportional fairness and stability. J Oper Res Soc 49:237–252
17. Mascolo S (1999) Congestion control in high-speed communication networks using the Smith principle. Automatica 35:1921–1935
18. Mascolo S (2000) Smith's principle for congestion control in high-speed data networks. IEEE Trans Autom Control 45:358–364
19. Michiels W, Niculescu S (2007) Stability and stabilization of time-delay systems: an eigenvalue-based approach. SIAM, Philadelphia
20. Mirkin L, Raskin N (2003) Every stabilizing dead-time controller has an observer-predictor-based structure. Automatica 39:1747–1754
21. Nguang SK (2001) Comments on 'Robust stabilization of uncertain input-delay systems by sliding mode control with delay compensation'. Automatica 37:1677
22. Quet PF, Ataşlar B, İftar A, Özbay H, Kalyanaraman S, Kang T (2002) Rate-based flow controllers for communication networks in the presence of uncertain time-varying multiple time-delays. Automatica 38:917–928
23. Roh YH, Oh JH (1999) Robust stabilization of uncertain input-delay systems by sliding mode control with delay compensation. Automatica 35:1861–1865
24. Roh YH, Oh JH (2000) Sliding mode control with uncertainty adaptation for uncertain input-delay systems. Int J Control 73:1255–1260
25. Shtessel Y, Fridman L, Zinober AIS (2008) Advances in higher order sliding mode control. Special Issue Int J Robust Nonlinear Control 18:381–585
26. Sichitiu ML, Bauer PH, Premaratne K (2003) The effect of uncertain time-variant delays in ATM networks with explicit rate feedback: a control theoretic approach. IEEE/ACM Trans Netw 11:628–637

Chapter 5
Flow Control in a Single-Source Discrete-Time System

In this chapter, we direct our attention to the design of flow control algorithms for networks in which the feedback information about the current network state is accessible for source rate adaptation at discrete time instants only. In this type of networks, in addition to the effects of nonnegligible delay, the design procedures need to explicitly account for the phenomena related to finite sampling rate. Hence, in this chapter, both the modeling and the controller design are performed directly in discrete-time domain.

In the presented modeling concept, the effects of input-output delay are accounted for by appropriately augmenting the state space. In this way, we may overcome the obstacle of DT which limits the application of many attractive control techniques known to be efficient in systems without delay, for example, SMC, or optimal control. In the extended state space, the networks are modeled as discrete-time *n*th-order systems. Using the state-space representation, several flow control algorithms are designed, each based on sound, control-theoretic foundations. Since the work concentrates on robust control methods, the presented methodology resides in the application of advanced, robust control technique – SMC. Because the key issue in the design of SM controllers, especially in discrete-time domain [23], is a proper choice of the sliding plane, we devote much attention to the selection of the plane parameters. Several approaches are discussed, such as LQ optimization or dead-beat control combined with a reaching law. Each control algorithm is formulated in a closed form which is straightforward in software (or hardware) implementation and allows for good operational efficiency. In addition, the closed-form solution of the optimization problem enables us to conduct a detailed analytical study of the system properties and prove them mathematically.

As it has already been discussed, the congestion control problem can be analyzed from a single source-destination pair perspective, or taking into account the whole set of active connections. Therefore, similarly as in Chap. 4, both the network modeling and the controller design will be performed for either case separately. However, due to numerous points that require considerable attention, we split the

P. Ignaciuk and A. Bartoszewicz, *Congestion Control in Data Transmission Networks: Sliding Mode and Other Designs*, Communications and Control Engineering,
DOI 10.1007/978-1-4471-4147-1_5, © Springer-Verlag London 2013

discussion to two chapters. This chapter comprises the fundamental concepts of single-flow transmission, and Chap. 6 is entirely devoted to multisource topologies.

In this chapter, we study the phenomena accompanying the transmission of data with the emphasis placed on a single flow. First, in Sect. 5.1, we analyze the flow control problem in a network with constant delay and present a few control algorithms. The first two control laws are obtained by solving the LQ problem with different performance indices. The first one is obtained when the entire state vector is considered in the optimization procedure, while in the latter case, the derivation concentrates on the system output variable with extra weighting coefficient introduced into the cost functional for tuning purposes. Since the optimal control strategy may require large transmission rates in the initial phase of the control process, in the subsequent part of Sect. 5.1, we analyze various methods of constraining excessive input signal. We investigate three attractive techniques: (1) application of a time-varying sliding plane, (2) design based on a reaching law, and, finally, (3) a method incorporating a direct transmission rate limiter. We discuss the design trade-offs and differences among the techniques with respect to the efficiency of handling the flow of data and the tuning effort. Next, in Sect. 5.2, we proceed towards a very important class of problems in communication networks originating from latency variations. We present a consistent methodology for modeling the effects caused by delay fluctuations in the feedback and data channels. The developed methodology allows for an effective study of the phenomena related to unknown, time-varying delay. It also enables the design of control strategies via worst-case uncertainty approach. We propose two robust algorithms. The first one combines the benefits of SM and LQ optimal control with a saturation element to provide feasible rate allocations, whereas the second one uses a novel technique for compensating the effects of delay variations by means of input rate measurements.

The analytical study is supplemented with extensive numerical tests reported at the end of each section.

5.1 Flow Control in a Network with Constant Delay

In this section, we analyze the basic mechanisms of feedback information interchange in the networks in which the packet emission rate of data sources can be adjusted only at discrete time instants. We present a modeling concept involving an extended state space and derive a number of control algorithms for data flow regulation in a single connection. We begin the analysis with the network where the delay of the feedback information delivery remains constant during the whole control process. Afterwards, in Sect. 5.2, we move to a more complex scenario, where neither the delay in the feedback channel nor the latency in the data path can be assumed invariable, or known *a priori*.

5.1.1 Network Model

We study the data flow process in a communication network in which packets and feedback carriers follow a fixed route, established in the connection setup phase preceding the actual transmission. If feedback carriers are served with priority over data packets, then the delay in a flow following a fixed route can be assumed constant and equal to its estimate determined at the connection setup. Such situation takes place in certain connection-oriented networks, for example, ATM, where the feedback information is delivered in special, priority-served control units (in ATM networks, these were the RM cells). From now on, we will use the term "data packet" or simply "packet" when referring to the piece of user data in the stream generated by the source, and the term "control unit" when referring to the feedback carrier, such as an RM cell in ATM, or an acknowledgment in TCP/IP networks.

We consider the flow of data in a single connection passing through a series of nodes. One node offers the smallest transfer capabilities at the output link and is considered the bottleneck for the connection. The purpose of the control algorithm operating at the bottleneck node is to regulate intensity of the data stream generated by the source in such a way that the node buffer is not overflowed with packets when there is little bandwidth available at the output link. On the other hand, when there is much bandwidth available for data transfer at the link, then we want to ensure that it is not wasted due to slow incoming rate. The feedback mechanism for the input rate regulation is provided by means of control units emitted periodically by the source. These special units travel along the same path as data packets. However, unlike data packets, they are not stored in the queues at the intermediate nodes. Instead, once they appear at the node input link and the feedback information is incorporated, they are immediately transferred at the output port. As soon as control units reach destination, they are turned back to be retrieved at the origin and to be used for the transfer speed adjustment round-trip time after they were generated. The presented concept is illustrated in Fig. 5.1. Source **S** sends packets interleaved with control units along the established data path to destination **D**. The path leads through nodes 1, 2, and 3. Control units are served with priority over data packets and are not stored in the buffers at the nodes. Each node assigns rate for the source and records it in the received control units. Node 2 calculates the smallest rate, and hence, it is considered the bottleneck for the analyzed connection.

The schematic diagram of the connection is presented in Fig. 5.2. The source sends packets at discrete time instants in the amounts determined by the controller placed at the bottleneck node. The number of packets to be delivered by the source, which is recorded as the feedback information in every control unit passing through the node, will be denoted by $u(kT)$, where T is the discretization period and $k = 0, 1, 2, \ldots$. After forward delay, T_F packets reach the bottleneck node and are served according to the bandwidth availability at the output link. The remaining data accumulates in the buffer. The packet queue length in the buffer, which at time kT will be denoted as $y(kT)$, and its demand value $y_D > 0$, are used to calculate the

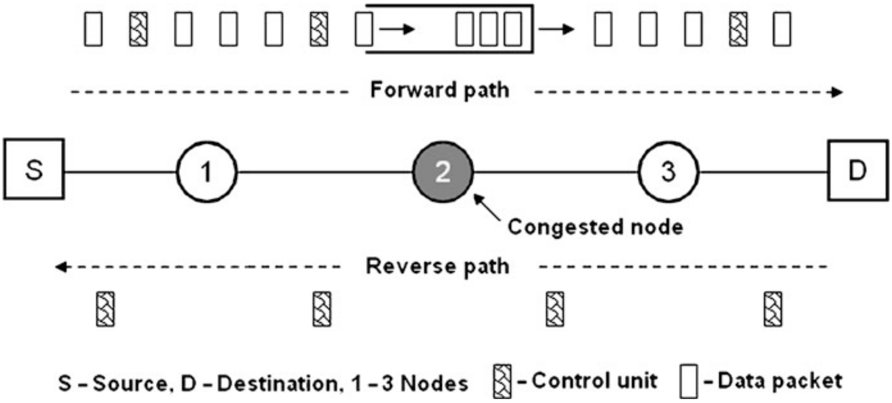


Fig. 5.1 Data transmission and feedback interchange concept

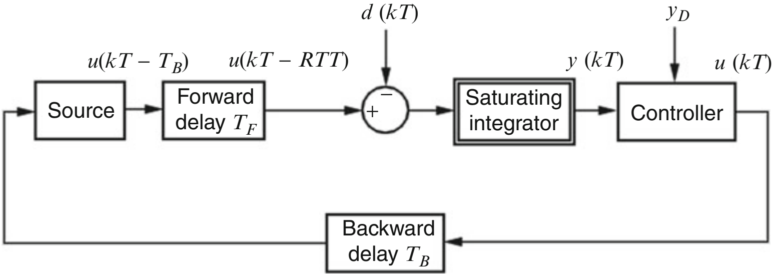


Fig. 5.2 Schematic diagram of single connection with constant round-trip time

current amount of data to be sent by the source $u(kT)$. Once control units appear at the end system, they are turned back to arrive at their origin with backward delay T_B after being processed by the bottleneck node. Since control units are not subject to queuing delays, the round-trip time $\text{RTT} = T_F + T_B = n_p T$, where n_p is a positive integer, remains constant for the duration of the connection. The RTT value does not depend on the location of the bottleneck node on the data path. The forward and backward delays may differ, but their sum constituting RTT remains unchanged, always equal to $n_p T$.

The available bandwidth (the number of packets which may leave the bottleneck node at each kT instant) is modeled as an *a priori* unknown, bounded function of time $d(kT)$

$$0 \le d\,(kT) \le d_{\max}. \tag{5.1}$$

Notice that this definition of the available bandwidth is general enough to capture any variations and traffic statistics typically analyzed in the considered problem.

If there are packets ready for transmission in the buffer, then the bandwidth actually consumed for data transfer $h(kT)$ (the number of packets actually leaving the node) will be equal to the available bandwidth. Otherwise, the output link is underutilized and the exploited bandwidth matches the data arrival rate at the node. Thus, we may write

$$0 \leq h\,(kT) \leq d\,(kT) \leq d_{\max}. \tag{5.2}$$

The rate of change of the queue length at any instant of time depends on the data arrival speed and on the consumed bandwidth $h(\cdot)$. Consequently, the queue length dynamics obeys the following simple retarded difference equation:

$$y\,[(k+1)\,T] = y\,(kT) + u\,(kT - \mathrm{RTT}) - h\,(kT)\,. \tag{5.3}$$

We assume that before the connection is established there are no packets in the buffer, i.e., $y(kT) = 0$ for $k \leq 0$. Then, for any $kT \geq 0$, the length of the queue at the node may be expressed in the alternative form as

$$y\,(kT) = \sum_{j=0}^{k-1} u\,(jT - \mathrm{RTT}) - \sum_{j=0}^{k-1} h\,(jT). \tag{5.4}$$

Applying the definition $\mathrm{RTT} = n_p T$, we can rewrite (5.4) as

$$y\,(kT) = \sum_{j=0}^{k-1} u\left(jT - n_p T\right) - \sum_{j=0}^{k-1} h\,(jT) = \sum_{j=-n_p}^{k-n_p-1} u\,(jT) - \sum_{j=0}^{k-1} h\,(jT). \tag{5.5}$$

Assuming that the controller determines the initial rate at the time instant $kT = 0$, the first packets arrive at the node at $kT = \mathrm{RTT}$, and $y(kT) = 0$ for $k \leq n_p$. Consequently, we get the following equation describing the evolution of packet queue length:

$$y\,(kT) = \sum_{j=0}^{k-n_p-1} u\,(jT) - \sum_{j=0}^{k-1} h\,(jT). \tag{5.6}$$

This relation will be used in the analytical study later in this section.

The interaction among the key network variables defined in the presented model is illustrated in Example 5.1.

Example 5.1. We analyze the flow of packets in a connection characterized by round-trip time RTT = 7T. The sequence of events in the initial phase of the control process is portrayed in Fig. 5.3, and the evolution of network variables $u(\cdot)$, $y(\cdot)$, and $h(\cdot)$ is depicted in Fig. 5.4. Forward delay T_{F} is assumed equal to $3T$ and backward delay $T_{\mathrm{B}} = 4T$. Suppose the demand queue length y_{D} is set as 9 packets

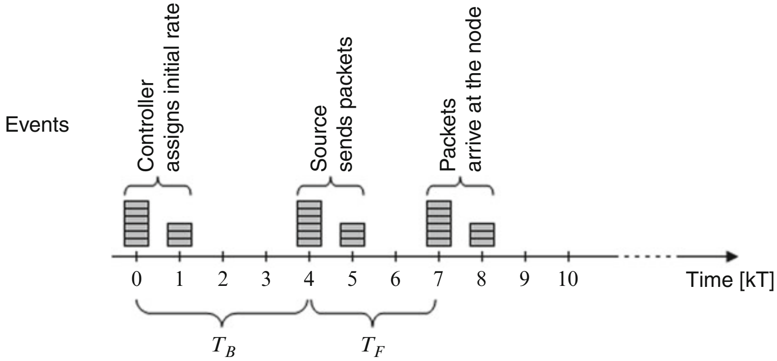


Fig. 5.3 Events in the initial phase of the control process

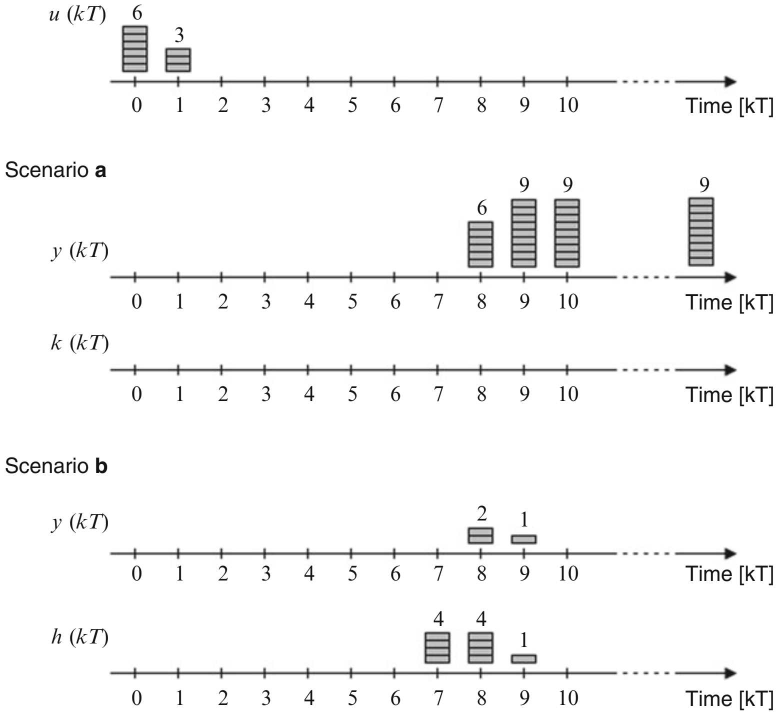


Fig. 5.4 Network variables for (**a**) $d(kT) \equiv 0$ and (**b**) $d(kT) \equiv 4$ packets

and the controller assigns the rate in the following way: $u(0) = 6$ packets, $u(T) = 3$ packets, and $u(kT) = 0$ for $k > 1$. We consider two cases: (a) $d(kT) \equiv 0$ and (b) $d(kT) \equiv 4$ packets per discretization period. In either case, the source acquires the feedback information with delay $T_B = 4T$ and sends 6 packets at instant $4T$ and 3 packets at instant $5T$. The packets arrive at the node at instants $7T$ and $8T$, respectively. In scenario (a), there is no bandwidth available to relay the received packets to the subsequent node on the data path. Therefore, the queue length calculated according to (5.3) is determined as $y(k \leq 7) = 0$, $y(8T) = 6$ packets, and $y(k \geq 9) = 9$ packets. The utilized bandwidth $h(kT) = 0$. In scenario (b), there is positive bandwidth available for transferring the data. Consequently, the evolution of the queue length follows a different pattern than in case (a). The first 6 packets arrive at instant $7T$, and since $d(kT) = 4$ packets, 4 packets are immediately transferred at the output interface. The remaining 2 packets are stored in the buffer, constituting the queue buildup $y(8T) = 2$ packets. Three more packets arrive at $kT = 8T$, and 4 packets are transferred at the node output interface. Consequently, according to (5.3), $y(9T) = y(8T) + u(T) – h(8T) = 2 + 3 – 4 = 1$ packet. No further packets are emitted by the source, thus $y(10T) = y(9T) + u(2) – h(9T) = 1 + 0 – 1 = 0$ packets, and $y(k > 10) = 0$. The utilized bandwidth $h(kT)$ takes on the following values:

- Zero for $k < 7$ – the buffer is initially empty, and $\mathrm{RTT} = 7T$ must elapse before the first packets to arrive at the bottleneck node.
- Four packets at $kT = 7T$, i.e., 6 packets reach the node at instant $7T$ and 4 packets are immediately transferred at the output interface according to the bandwidth availability $d(7T) = 4$ packets.
- Four packets at $kT = 8T$, i.e., 6 packets are taken from the queue, and two more packets are relayed directly from the pool of the incoming ones to fill up the available bandwidth of 4 packets.
- One packet at $kT = 9T$ – no packets arrive at the node, and only a single packet remaining in the buffer is transmitted, which lowers the bandwidth utilization to 1 packet.
- Zero for $k > 9$ – there are no more packets on route, neither there are packets assigned by the controller for the source to emit; in consequence, all the available bandwidth for $k > 9$ will be wasted unless the controller allows the source to send additional data.

As can be noticed from Fig. 5.4, in scenario (a), the queue length stays at the demand level of 9 packets following the initial phase, whereas in the second case (b), a further controller action will be required to bring the output variable to the target value y_{D}. This is due to the presence of positive available bandwidth, which acts as an external disturbance to the integrating action of the controlled plant (the node buffer). We will show further in the text (in Chap. 7) how to cope with the drift of the output variable from the target value due to the presence of persistent positive disturbance (by means of feed-forward bandwidth compensation).

In this chapter, we focus on maximizing the efficiency of data transfer in the network modeled as stated above. For this purpose, we will propose a number of control laws that guarantee the highest throughput despite the presence of delay and unknown variable bandwidth.

The discussed network model can also be presented in the state space. The state-space realization facilitates adaptation of formal design techniques, and therefore, it is selected as a basis for the control law derivation described in detail in the next section.

5.1.1.1 State-Space Representation

In order to proceed with a formal controller design, we describe the discrete-time model of the considered network in the state space

$$\begin{aligned} \mathbf{x}\left[(k+1)T\right] &= \mathbf{A}\mathbf{x}\left(kT\right) + \mathbf{b}u\left(kT\right) + \mathbf{v}h\left(kT\right), \\ y\left(kT\right) &= \mathbf{q}^{\mathrm{T}}\mathbf{x}\left(kT\right), \end{aligned} \tag{5.7}$$

where $\mathbf{x}(kT) = [x_1(kT)\ x_2(kT)\ x_3(kT)\ \ldots\ x_n(kT)]^{\mathrm{T}}$ is the state vector with $x_1(kT) = y(kT)$ representing the bottleneck queue length at instant kT, and the remaining state variables $x_j(kT) = u[(k-n+j-1)T]$ for any $j = 2, 3, \ldots, n$ equal to the delayed input signal u. $\mathbf{A}$ is $n \times n$ state matrix; $\mathbf{b}$, $\mathbf{v}$, and $\mathbf{q}$ are $n \times 1$ vectors

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} -1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}; \tag{5.8}$$

and the system order $n = \mathrm{RTT}/T + 1 = n_p + 1$. For convenience of the further analysis, we can rewrite the model in the alternative form

$$\begin{cases} x_1\left[(k+1)T\right] = x_1\left(kT\right) + x_2\left(kT\right) - h\left(kT\right), \\ x_2\left[(k+1)T\right] = x_3\left(kT\right), \\ x_3\left[(k+1)T\right] = x_4\left(kT\right), \\ \qquad \vdots \\ x_{n-1}\left[(k+1)T\right] = x_n\left(kT\right), \\ x_n\left[(k+1)T\right] = u\left(kT\right), \end{cases} \tag{5.9}$$

which clearly shows how the choice of the (extended) state space relates to the delay in the feedback loop. The desired system state is defined as

$$\mathbf{x_d} = \begin{bmatrix} x_{d1} \\ x_{d2} \\ \vdots \\ x_{dn-1} \\ x_{dn} \end{bmatrix} = \begin{bmatrix} x_{d1} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \tag{5.10}$$

where $x_{d1} = y_{\mathrm{D}}$ denotes the target value of the first state variable, i.e., the demand queue length. By choosing the desired state vector as

$$\mathbf{x_d} = \left[y_{\mathrm{D}}\ 0\ 0\ \ldots\ 0\right]^{\mathrm{T}},$$

we want the first state variable (the packet queue length) to reach the level y_{D} and to be kept at this level in the steady state. For this situation to take place, all the state variables $x_2, x_3, \ldots, x_n$ should be zero once $x_1(kT)$ becomes equal to y_{D}, exactly as dictated by (5.10).

In the next section, Eqs. (5.7)–(5.10) describing the system behavior and interactions among the principal network variables (transmission rate, queue length, and available bandwidth) will be used to develop a discrete-time SM flow control strategy.

5.1.2 SM Controller with LQ Optimal Sliding Plane

In this section, a control-theoretic approach is employed to design a discrete-time SM controller for the considered communication network. The emphasis is placed on the selection of the sliding plane, which has a decisive impact on the performance of the control process. We propose to apply dynamical optimization with quadratic quality criterion to obtain the plane parameters. Actually, two approaches for the selection of the plane parameters are considered, each focused on the minimization of a different cost functional. In the first optimization task, the whole state vector is taken into account in the control law derivation, while in the second one, we concentrate on the output variable. In the second optimization task, an additional weighting coefficient is introduced into the cost functional for tuning purposes. The presented procedures concentrate on the solution of a matrix Riccati equation for the considered nth-order discrete-time system. As the typical approaches for solving Riccati equations are mainly suitable for numerical implementations and systems with predefined dimensions (see, e.g., [1, 2, 6, 9, 16, 17, 20, 21]), an analytic method is developed to get the desired plane coefficients. The proposed method is based on iterative substitution of matrices obtained at the intermediate steps of the derivation. The derivation ends when all the elements of the unknown matrix in the Riccati equation can be expressed in terms of the system parameters. The analytical solution of optimization problem allows us to formulate the control law in a closed form and proceed with a successful proof of a number of its advantageous properties.

5.1.2.1 Controller Design

Let us denote the closed-loop system error as $\mathbf{e}(kT) = \mathbf{x_d} - \mathbf{x}(kT)$. We introduce a sliding hyperplane described by the following equation:

$$s\,(kT) = \mathbf{c}^{\mathrm{T}}\mathbf{e}\,(kT) = 0, \tag{5.11}$$

where $\mathbf{c}^{\mathrm{T}} = [c_1\ c_2\ c_3\ \ldots\ c_n]$ is such a vector that $\mathbf{c}^{\mathrm{T}}\mathbf{b} \neq 0$. The selection of this vector will be analyzed further in this section. Substituting (5.7) into equation $\mathbf{c}^{\mathrm{T}}\mathbf{e}[(k+1)T] = 0$ with the disturbance $h(kT) \equiv 0$, we get

$$\mathbf{c}^{\mathrm{T}}\mathbf{e}\left[(k+1)\,T\right] = \mathbf{c}^{\mathrm{T}}\left\{\mathbf{x_d} - \mathbf{x}\left[(k+1)\,T\right]\right\} = \mathbf{c}^{\mathrm{T}}\left[\mathbf{x_d} - \mathbf{Ax}\,(kT) - \mathbf{b}u\,(kT)\right] = 0, \tag{5.12}$$

which leads to the following feedback control law

$$u\,(kT) = \left(\mathbf{c}^{\mathrm{T}}\mathbf{b}\right)^{-1}\mathbf{c}^{\mathrm{T}}\left[\mathbf{x_d} - \mathbf{Ax}\,(kT)\right]. \tag{5.13}$$

Using (5.8) and (5.10), we can rewrite (5.13) in the following form:

$$u\,(kT) = {c_n}^{-1}\left\{c_1\left[y_{\mathrm{D}} - x_1\,(kT)\right] - \sum_{j=2}^{n} c_{j-1}x_j\,(kT)\right\}. \tag{5.14}$$

It is well known that properties of SM controllers are determined by an appropriate choice of the sliding plane parameters $c_1, c_2, \ldots, c_n$. We present two approaches to the selection of the elements of vector $\mathbf{c}$ so that SM control law, optimal in the LQ sense, is obtained.

Case 1. The aim of the control action can be defined as bringing the current system state to a desired one without excessive control effort. In alternative terms, we may specify the control objective as reducing the closed-loop error to zero using a reasonable data flow rate. Therefore, we seek for an optimal SM control $u_{\mathrm{opt}}(kT)$, which will minimize the quality criterion expressed by the quadratic cost functional

$$J(u) = \frac{1}{2}\sum_{k=0}^{\infty}\left[\mathbf{e}^{\mathrm{T}}\,(kT)\,\mathbf{Qe}\,(kT) + Ru^2\,(kT)\right], \tag{5.15}$$

where $\mathbf{Q}_{n\times n}$ is a symmetric positive semi-definite matrix and R is a positive constant (note that in the considered system the input is a scalar). Choosing $\mathbf{Q}$ as identity matrix $\mathbf{I}_n = \mathrm{diag}\{1, 1, \ldots, 1\}$ and $R = 1$, we get the following performance index:

$$J_1(u) = \frac{1}{2}\sum_{k=0}^{\infty}\left[\mathbf{e}^{\mathrm{T}}\,(kT)\,\mathbf{e}\,(kT) + u^2\,(kT)\right]. \tag{5.16}$$

Applying the standard framework for solving the LQ problems, described, for example, in [30, ch. 8], to system (5.7)–(5.10), the optimal control $u_{\mathrm{opt}}(kT)$ minimizing criterion (5.16) can be presented as

$$u_{\mathrm{opt}}\,(kT) = -\mathbf{gx}\,(kT) + r, \tag{5.17}$$

where

$$
\begin{aligned}
\mathbf{g} &= \mathbf{b}^{\mathrm{T}}\mathbf{K}\big(\mathbf{I}_n + \mathbf{b}\mathbf{b}^{\mathrm{T}}\mathbf{K}\big)^{-1}\mathbf{A},\\
r &= \mathbf{b}^{\mathrm{T}}\Big[\mathbf{K}\big(\mathbf{I}_n + \mathbf{b}\mathbf{b}^{\mathrm{T}}\mathbf{K}\big)^{-1}\mathbf{b}\mathbf{b}^{\mathrm{T}} - \mathbf{I}_n\Big]\mathbf{k},\\
\mathbf{k} &= -\mathbf{A}^{\mathrm{T}}\Big[\mathbf{K}\big(\mathbf{I}_n + \mathbf{b}\mathbf{b}^{\mathrm{T}}\mathbf{K}\big)^{-1}\mathbf{b}\mathbf{b}^{\mathrm{T}} - \mathbf{I}_n\Big]\mathbf{k} - \mathbf{x_d},
\end{aligned}
\tag{5.18}
$$

and symmetric matrix $\mathbf{K}_{n\times n}$, which is at least positive semidefinite $\mathbf{K} \geq 0$, is determined according to the following Riccati equation:

$$
\mathbf{K} = \mathbf{A}^{\mathrm{T}}\mathbf{K}\big(\mathbf{I}_n + \mathbf{b}\mathbf{b}^{\mathrm{T}}\mathbf{K}\big)^{-1}\mathbf{A} + \mathbf{I}_n. \tag{5.19}
$$

Note that the pair ($\mathbf{A}$, $\mathbf{b}$) is stabilizable. Indeed, the controllability matrix for the considered system

$$
[\mathbf{b}\ \mathbf{A}\mathbf{b}\ \ldots\ \mathbf{A}^{n-1}\mathbf{b}] = \begin{bmatrix} 0 & \ldots & 0 & 1\\ 0 & \ldots & 1 & 0\\ \vdots & \iddots & 0 & \vdots\\ 1 & 0 & \ldots & 0 \end{bmatrix} \tag{5.20}
$$

is of full rank. Hence, the system is full state controllable, which implies stabilizability of the pair ($\mathbf{A}$, $\mathbf{b}$). On the other hand, since $\mathbf{Q} = \mathbf{I}_n$ is positive definite, the pair ($\mathbf{A}$, $\sqrt{\mathbf{Q}}$) = ($\mathbf{A}$, $\mathbf{I}_n$) is observable. Consequently, as the system is stabilizable (the pair ($\mathbf{A}$, $\mathbf{b}$) stabilizable) and the state is observable by the performance index (the pair ($\mathbf{A}$, $\sqrt{\mathbf{Q}}$) observable), then there exists a positive definite solution, $\mathbf{K} > 0$, to algebraic Riccati Eq. (5.19). For a more detailed discussion on seeking the solution to optimal control problems, refer to [20, ch. 2], whereas notes on checking controllability and observability can be found, for example, in [30, ch. 6].

The classical approaches to solving (5.19) suggested in the literature, for example, [1, 20, 30], are mainly suitable for numerical calculations and systems with predefined dimensions. However, in order to perform a detailed analytical study of the system properties for arbitrary delay, it is desirable to find a closed-form expression for the developed control law. This requires analytical solution of the Riccati equation of order n. The novel method proposed in this work involves iterative substitution of $\mathbf{K}$ into the expression on the right-hand side of (5.19) and comparison with its left-hand side so that at each step the number of independent variables k_{ij}, where k_{ij} denotes the element in the ith row and jth column of $\mathbf{K}$, is reduced.

We begin with the most general form of matrix $\mathbf{K}$ which can be presented as

$$
\mathbf{K}_0 = \begin{bmatrix} k_{11} & k_{12} & \ldots & k_{1n}\\ k_{12} & k_{22} & \ldots & k_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ k_{1n} & k_{2n} & \ldots & k_{nn} \end{bmatrix}. \tag{5.21}
$$

In the first iteration, we place $\mathbf{K}_0$ directly in (5.19), and after substituting matrix $\mathbf{A}$ and vector $\mathbf{b}$ as defined by (5.8), we seek for similarities between the elements k_{ij} on either side of the equality sign in (5.19). In this way, we find the relations among the first four elements in the upper left corner of $\mathbf{K}$: $k_{12} = k_{11} - 1$ and $k_{22} = k_{11}$ (note that $k_{21} = k_{12}$ since $\mathbf{K}$ is symmetric). Consequently, after the first analytical iteration, we obtain the following form of $\mathbf{K}$:

$$\mathbf{K}_1 = \begin{bmatrix} k_{11} & k_{11}-1 & k_{13} & \dots & k_{1n} \\ k_{11}-1 & k_{11} & k_{23} & \dots & k_{2n} \\ k_{13} & k_{23} & k_{33} & \dots & k_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{1n} & k_{2n} & k_{3n} & \dots & k_{nn} \end{bmatrix}. \tag{5.22}$$

Now, we substitute $\mathbf{K}_1$ given by (5.22) into the expression on the right-hand side of (5.19) and compare with its left-hand side, which allows us to represent the elements k_{i3} ($i = 1, 2, 3$) in terms of k_{11}: $k_{13} = k_{23} = k_{11} - 2$ and $k_{33} = k_{11}$. This results in

$$\mathbf{K}_2 = \begin{bmatrix} k_{11} & k_{11}-1 & k_{11}-2 & k_{14} & \dots & k_{1n} \\ k_{11}-1 & k_{11} & k_{11}-2 & k_{24} & \dots & k_{2n} \\ k_{11}-2 & k_{11}-2 & k_{11} & k_{34} & \dots & k_{3n} \\ k_{14} & k_{24} & k_{34} & k_{44} & \dots & k_{4n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{1n} & k_{2n} & k_{3n} & k_{4n} & \dots & k_{nn} \end{bmatrix}. \tag{5.23}$$

We repeat the substitutions until all the elements of $\mathbf{K}$ can be expressed as functions of k_{11} and the system order n. The final closed-form expression for $\mathbf{K}$, given in terms of its first element k_{11} and the system order, is determined as

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{11}-1 & k_{11}-2 & \dots & k_{11}-n+1 \\ k_{11}-1 & k_{11} & k_{11}-2 & \dots & k_{11}-n+1 \\ k_{11}-2 & k_{11}-2 & k_{11} & \dots & k_{11}-n+1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{11}-n+1 & k_{11}-n+1 & k_{11}-n+1 & \dots & k_{11} \end{bmatrix}. \tag{5.24}$$

For matrix $\mathbf{K}$ given by (5.24), any $k_{11} \geq n - 1$ ensures that all the leading principal minors and the determinant are positive, and in consequence, it guarantees that $\mathbf{K}$ is positive definite. In order to determine k_{11}, we substitute (5.24) into the expression on the right-hand side of Eq. (5.19) and compare the first element in the upper left corner of the obtained matrices. This yields

$$k_{11} = \frac{2n - n^2}{(k_{11} + 1)}. \tag{5.25}$$

Equation (5.25) has two roots $k_{11}^{-} = (2n-1-\sqrt{4n+1})/2$ and $k_{11}^{+} = (2n-1+\sqrt{4n+1})/2$. Only $k_{11}^{+} > n-1$ guarantees that $\mathbf{K}$ is positive definite and constitutes the desired solution of (5.25). This concludes the solution of the Riccati equation in the optimal control problem (5.16) with constraint (5.7) and (5.8).

Having found $\mathbf{K}$, we may determine $\mathbf{g} = \mathbf{b}^{\mathrm{T}}\mathbf{K}(\mathbf{I_n} + \mathbf{b}\mathbf{b}^{\mathrm{T}}\mathbf{K})^{-1}\mathbf{A}$. Substituting (5.8) and (5.24) into the first equation in set (5.18), we obtain

$$\mathbf{g} = \left[1\ 1\ 1\ \ldots\ 1\right][1 - n/(k_{11}+1)]\,. \tag{5.26}$$

In a similar way, we calculate the elements of vector $\mathbf{k} = [k_1\, k_2\, k_3\, \ldots\, k_n]^T$. By substituting $\mathbf{K}$ given by (5.24) into the last equation in set (5.18), we get

$$\mathbf{k} = \left[k_1\ k_1 + y_{\mathrm{D}}\ k_1 + 2y_{\mathrm{D}}\ \ldots\ k_1 + (n-1)\,y_{\mathrm{D}}\right]^{\mathrm{T}}, \tag{5.27}$$

and, consequently,

$$k_1 = -y_{\mathrm{D}} n\left[1 + (k_{11} - n + 1)^{-1}\right]. \tag{5.28}$$

From the second equation in set (5.18), we find r:

$$r = \frac{-\left[k_1 + (n-1)\,y_{\mathrm{D}}\right]}{(k_{11}+1)}. \tag{5.29}$$

Substituting k_1 given by (5.28) into (5.29), we get $r = y_{\mathrm{D}}/(k_{11} - n + 1)$. Finally, using (5.26), the control $u_{\mathrm{opt}}(kT)$ can be presented in the following form:

$$u_{\mathrm{opt}}\,(kT) = -\left(1 - \frac{n}{k_{11}+1}\right)\sum_{j=1}^{n} x_j\,(kT) + \frac{y_{\mathrm{D}}}{k_{11} - n + 1}. \tag{5.30}$$

Substituting $k_{11} = (2n - 1 + \sqrt{4n+1})/2$ into (5.30), we obtain

$$\begin{aligned} u_{\mathrm{opt}}\,(kT) &= -\left(1 - \frac{2n}{2n + 1 + \sqrt{4n+1}}\right)\sum_{j=1}^{n} x_j\,(kT) + \frac{2y_{\mathrm{D}}}{\sqrt{4n+1}+1} \\ &= -\frac{\sqrt{4n+1}-1}{2n}\sum_{j=1}^{n} x_j\,(kT) + \frac{\left(\sqrt{4n+1}-1\right)y_{\mathrm{D}}}{2n}. \end{aligned} \tag{5.31}$$

Taking out the common term, we get

$$u_{\mathrm{opt}}\,(kT) = \frac{\sqrt{4n+1}-1}{2n}\left[y_{\mathrm{D}} - x_1\,(kT) - \sum_{j=2}^{n} x_j\,(kT)\right]. \tag{5.32}$$

Introducing

$$\gamma_1 = \frac{\left(\sqrt{4n+1} - 1\right)}{2n}, \tag{5.33}$$

we arrive at

$$u_{\text{opt}}\left(kT\right) = \gamma_1 \left[y_{\text{D}} - x_1\left(kT\right) - \sum_{j=2}^{n} x_j\left(kT\right) \right]. \tag{5.34}$$

If we compare this control law with SM controller (5.14), we get

$$\frac{c_1}{c_n} = \gamma_1 \quad \text{and} \quad c_1 = c_2 = \cdots = c_{n-2} = c_{n-1}. \tag{5.35}$$

Hence, the elements of vector **c**

$$\mathbf{c}^{\text{T}} = \left[\gamma_1\ \gamma_1\ \ldots\ \gamma_1\ 1\right] c_n, \tag{5.36}$$

and the LQ optimal SM control

$$u\left(kT\right) = \gamma_1 \left[y_{\text{D}} - x_1\left(kT\right) - \sum_{j=2}^{n} x_j\left(kT\right) \right]. \tag{5.37}$$

This concludes solution of the first optimization problem.

The relation between the controller gain and the system order is shown in Fig. 5.5. We can see from the plot that γ_1 monotonically decreases with the increase of n, which means that the designed controller (5.37) faster reacts to the bandwidth changes for the connections with smaller propagation delay. On the other hand, as $n \to \infty$, γ_1 approaches zero. This means that the controller based on performance index (5.16) may provide sluggish response for long-distance connections. In the second optimization problem stated below, we develop an enhanced control law which can ensure good responsiveness to the changing networking conditions irrespective of the delay range.

Remark 5.1. According to [30, ch. 8], the minimum value of the quality criterion can be determined from $J(u_{\text{opt}}) = 0.5\mathbf{e}^{\text{T}}(0)\mathbf{K}\mathbf{e}(0)$. Therefore, $J_1(u_{\text{opt}})$ equals

$$\frac{1}{2}\left[y_{\text{D}}\ 0 \ldots 0\right] \begin{bmatrix} k_{11} & k_{11}-1 & \ldots & k_{11}-n+1 \\ k_{11}-1 & k_{11} & \ldots & k_{11}-n+1 \\ \vdots & \vdots & \ddots & \vdots \\ k_{11}-n+1 & k_{11}-n+1 & \ldots & k_{11} \end{bmatrix} \begin{bmatrix} y_D \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \frac{1}{2} y_{\text{D}}^2 k_{11}. \tag{5.38}$$

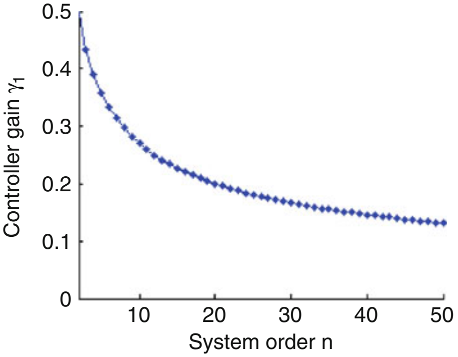


Fig. 5.5 Controller gain γ_1 vs. system order n

Substituting $k_{11} = k_{11}^{+} = (2n - 1 + \sqrt{4n + 1})/2$ into (5.38), we get

$$J_1\left(u_{\mathrm{opt}}\right) = \frac{1}{4} y_{\mathrm{D}}^2 \left(2n - 1 + \sqrt{4n + 1}\right). \tag{5.39}$$

Case 2. More commonly in the optimization problems, instead of considering the entire state vector, we analyze the situation when the minimum control effort is required to bring the output (or the controlled variable) to its desired value [1, 18, 20]. Thus, here in Case 2, we propose an alternative approach to the selection of the sliding plane parameters – by applying a different quality criterion. We will show that the gain of the resulting control law is independent of delay, and thus, the controller offers faster response to the changing networking conditions than the control law obtained in Case 1. Similarly as in Case 1, we consider the general quadratic cost functional (5.15) with input weight $R = 1$. However, the state weighting matrix $\mathbf{Q}$ is chosen to reflect only the error at the output. We choose

$$\mathbf{Q}_{n\times n} = w\mathbf{q}\mathbf{q}^{\mathrm{T}} = \begin{bmatrix} w & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}, \tag{5.40}$$

where w is a positive constant applied to adjust the influence of the controller command and the output variable on the value of the quality criterion. Thus, we consider the LQ optimal control problem with the following performance index:

$$J_2(u) = \frac{1}{2} \sum_{k=0}^{\infty} \left\{ w[y_{\mathrm{D}} - y\,(kT)]^2 + u^2\,(kT) \right\}. \tag{5.41}$$

The optimal control $u_{\mathrm{opt}}(kT)$ minimizing (5.41) can be presented as in (5.17). Vector $\mathbf{g}$ and constant r are obtained from the formulas already specified in (5.18).

However, in the case of criterion J_2 applied in the modified optimization problem, the last equation in set (5.18) needs to be changed to

$$\mathbf{k} = -\mathbf{A}^{\mathrm{T}}\left[\mathbf{K}\left(\mathbf{I}_n + \mathbf{b}\mathbf{b}^{\mathrm{T}}\mathbf{K}\right)^{-1}\mathbf{b}\mathbf{b}^{\mathrm{T}} - \mathbf{I}_n\right]\mathbf{k} - w\mathbf{q}y_{\mathrm{D}} \tag{5.42}$$

and the Riccati equation to

$$\mathbf{K} = \mathbf{A}^{\mathrm{T}}\mathbf{K}\left(\mathbf{I}_n + \mathbf{b}\mathbf{b}^{\mathrm{T}}\mathbf{K}\right)^{-1}\mathbf{A} + w\mathbf{q}\mathbf{q}^{\mathrm{T}}. \tag{5.43}$$

The pair $(\mathbf{A}, \mathbf{b})$ is stabilizable (see the derivation in (5.20)). The square root of $\mathbf{Q}$ equals

$$\left[\sqrt{w}\ 0 \ldots 0\right]. \tag{5.44}$$

Hence, the observability matrix for the pair $(\mathbf{A}, \sqrt{\mathbf{Q}})$,

$$\begin{bmatrix} \sqrt{\mathbf{Q}} \\ \sqrt{\mathbf{Q}}\mathbf{A} \\ \vdots \\ \sqrt{\mathbf{Q}}\mathbf{A}^{n-1} \end{bmatrix}, \tag{5.45}$$

can be presented as a lower triangular matrix

$$\begin{bmatrix} \sqrt{w} & 0 & \ldots & 0 \\ \sqrt{w} & \sqrt{w} & 0 & \vdots \\ \vdots & \vdots & \ddots & 0 \\ \sqrt{w} & \sqrt{w} & \ldots & \sqrt{w} \end{bmatrix}, \tag{5.46}$$

which has rank n. This implies that the pair $(\mathbf{A}, \sqrt{\mathbf{Q}})$ is fully observable. Consequently, since $(\mathbf{A}, \mathbf{b})$ is stabilizable and $(\mathbf{A}, \sqrt{\mathbf{Q}})$ observable, there exists a positive definite solution, $\mathbf{K} > 0$, to algebraic Riccati equation (5.43).

Similarly as in Case 1, we begin solving (5.43) with the most general form of $\mathbf{K}$ given by (5.21). In the first iteration, we obtain the relationship among the first four elements in the upper left corner of $\mathbf{K}$ getting $k_{12} = k_{22} = k_{11} - w$. In matrix form, this can be written as

$$\mathbf{K}_1 = \begin{bmatrix} k_{11} & k_{11} - w & k_{13} & \ldots & k_{1n} \\ k_{11} - w & k_{11} - w & k_{23} & \ldots & k_{2n} \\ k_{13} & k_{23} & k_{33} & \ldots & k_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{1n} & k_{2n} & k_{3n} & \ldots & k_{nn} \end{bmatrix}. \tag{5.47}$$

Now, we substitute $\mathbf{K}_1$ given by (5.47) into the expression on the right-hand side of (5.43) and compare with its left-hand side. This allows us to represent the elements k_{i3} $(i = 1, 2, 3)$ in terms of k_{11} as $k_{13} = k_{23} = k_{33} = k_{11} - 2w$. In matrix form,

$$\mathbf{K}_2 = \begin{bmatrix} k_{11} & k_{11} - w & k_{11} - 2w & k_{14} & \dots & k_{1n} \\ k_{11} - w & k_{11} - w & k_{11} - 2w & k_{24} & \dots & k_{2n} \\ k_{11} - 2w & k_{11} - 2w & k_{11} - 2w & k_{34} & \dots & k_{3n} \\ k_{14} & k_{24} & k_{34} & k_{44} & \dots & k_{4n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{1n} & k_{2n} & k_{3n} & k_{4n} & \dots & k_{nn} \end{bmatrix}. \tag{5.48}$$

We proceed with the substitutions until a general pattern is determined, i.e., until all the elements of $\mathbf{K}$ can be expressed as functions of k_{11} and the system order n. We get $k_{ij} = k_{11} - (j-1)w$ for $j \geq i$ (the upper part of $\mathbf{K}$) and $k_{ij} = k_{11} - (i-1)w$ for $j < i$ (the lower part of $\mathbf{K}$). Thus, matrix $\mathbf{K}$,

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{11} - w & k_{11} - 2w & \dots & k_{11} - (n-1)\,w \\ k_{11} - w & k_{11} - w & k_{11} - 2w & \dots & k_{11} - (n-1)\,w \\ k_{11} - 2w & k_{11} - 2w & k_{11} - 2w & \dots & k_{11} - (n-1)\,w \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{11} - (n-1)\,w & k_{11} - (n-1)\,w & k_{11} - (n-1)\,w & \dots & k_{11} - (n-1)\,w \end{bmatrix}. \tag{5.49}$$

If we substitute (5.49) into the right-hand side of Eq. (5.43) and compare the first element in the upper left corner of the matrices on either side of the equality sign, we get the expression from which we can determine k_{11}:

$$k_{11} = nw + 1 - [k_{11} - (n-1)\,w + 1]^{-1}. \tag{5.50}$$

Equation (5.50) has two roots

$$k_{11}^{\pm} = \frac{\sqrt{w}\left[(2n-1)\sqrt{w} \pm \sqrt{w+4}\right]}{2}. \tag{5.51}$$

Since $\det(\mathbf{K}) = w^{n-1}[k_{11} - (n-1)w]$, only $k_{11}^{+} \geq (n-1)w$ guarantees that $\mathbf{K}$ is positive definite. Consequently, we get matrix $\mathbf{K}$ (5.49) with $k_{11} = k_{11}^{+}$ given by (5.51). This concludes the solution of the Riccati equation.

Having found $\mathbf{K}$, we evaluate $\mathbf{g}$:

$$\mathbf{g} = \left[\,1\ 1\ 1\ \dots\ 1\,\right]\left\{1 - [k_{11} - (n-1)\,w + 1]^{-1}\right\}. \tag{5.52}$$

Vector **k** is determined by substituting matrix **K** given by (5.49) into (5.42). We obtain

$$\mathbf{k} = \left[k_1 \; k_1 + wy_\mathrm{D} \; k_1 + 2wy_\mathrm{D} \; \ldots \; k_1 + (n-1)\, wy_\mathrm{D}\right]^\mathrm{T}, \tag{5.53}$$

where

$$k_1 = -wy_\mathrm{D}\left\{n + \left[k_{11} - (n-1)\, w\right]^{-1}\right\}. \tag{5.54}$$

Then using the second equation from (5.18) and substituting (5.54), we calculate r:

$$r = -\frac{k_1 + (n-1)\, wy_\mathrm{D}}{k_{11} - (n-1)\, w + 1} = \frac{wy_\mathrm{D}}{k_{11} - (n-1)\, w}. \tag{5.55}$$

Finally, using (5.52) and (5.55), the optimal control $u_\mathrm{opt}(kT)$ can be presented in the following way:

$$u_\mathrm{opt}(kT) = -\left(1 - \frac{1}{k_{11} - (n-1)\, w + 1}\right)\sum_{j=1}^{n} x_j(kT) + \frac{wy_\mathrm{D}}{k_{11} - (n-1)\, w}. \tag{5.56}$$

Substituting $k_{11} = \frac{\sqrt{w}\left[(2n-1)\sqrt{w} + \sqrt{w+4}\right]}{2}$, we arrive at

$$u_\mathrm{opt}(kT) = \frac{\sqrt{w(w+4)} - w}{2}\left[y_\mathrm{D} - x_1(kT) - \sum_{j=2}^{n} x_j(kT)\right]. \tag{5.57}$$

Introducing

$$\gamma_2 = \frac{\sqrt{w(w+4)} - w}{2}, \tag{5.58}$$

we obtain

$$u_\mathrm{opt}(kT) = \gamma_2\left[y_\mathrm{D} - x_1(kT) - \sum_{j=2}^{n} x_j(kT)\right]. \tag{5.59}$$

Comparing (5.59) with (5.14), we get

$$\frac{c_1}{c_n} = \gamma_2 \quad \text{and} \quad c_1 = c_2 = \cdots = c_{n-2} = c_{n-1}, \tag{5.60}$$

which in vector form can be written as

$$\mathbf{c}^{\mathrm{T}} = \left[\, \gamma_2 \; \gamma_2 \; \ldots \; \gamma_2 \; 1 \,\right] c_n. \tag{5.61}$$

Therefore, the discrete-time SM control law (5.14) takes the following form:

$$u\,(kT) = \gamma_2 \left[y_{\mathrm{D}} - x_1\,(kT) - \sum_{j=2}^{n} x_j\,(kT) \right]. \tag{5.62}$$

Notice that in contrast to result (5.33) obtained in the first optimization problem (Case 1), the gain of controller (5.62), $\gamma_2 = (\sqrt{w(w+4)} - w)/2$, does not depend on the system order. This means that in contrast to the control law (5.37), the dynamics of controller (5.62) is insensitive to the value of delay. As a result, the optimal control law obtained by considering the modified, output-based quality criterion (5.41), can provide equally fast response for both the local and long-distance flows. This concludes the second optimization procedure.

For the sake of further analysis, the control laws obtained in Case 1 and 2 may be more conveniently written using parameters of the original model (5.1)–(5.6). Let γ denote either γ_1 or γ_2. From (5.9), the state variables x_j $(j = 2, 3, \ldots, n)$ may be expressed in terms of the control signal generated at the previous $n-1$ samples as

$$x_j\,(kT) = u\,[(k - n + j - 1)\,T]\,. \tag{5.63}$$

Recall that we introduced the notation $x_1(kT) = y(kT)$. Then, since $n = n_p + 1$, substituting (5.63) into either (5.37) or (5.62), we obtain

$$\begin{aligned} u\,(kT) &= \gamma\,\{y_{\mathrm{D}} - y\,(kT) - u\,[(k-n+1)\,T] + \cdots + u\,[(k-1)\,T]\} \\ &= \gamma \left\{ y_{\mathrm{D}} - y\,(kT) - \sum_{j=1}^{n-1} u\,[(k-j)\,T] \right\} \\ &= \gamma \left\{ y_{\mathrm{D}} - y\,(kT) - \sum_{j=1}^{n_p} u\,[(k-j)\,T] \right\}. \end{aligned} \tag{5.64}$$

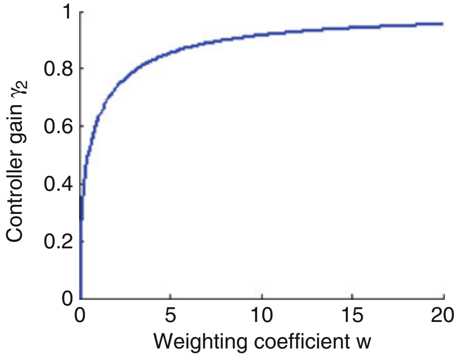


Fig. 5.6 Controller gain γ_2 vs. weighting coefficient w

This completes the design of LQ optimal SM flow control algorithm for the considered network.

Remark 5.2. When $w \to 0$, the influence of the error at the output on the value of cost functional J_2 decreases, and the controller gain γ_2 drops to zero. On the other hand, when $w \to \infty$, the term $y_{\mathrm{D}} - y(kT)$ prevails. In this situation, the error is to be reduced to zero as quickly as possible no matter the value of the controller command, and the proposed controller becomes a dead-beat scheme with the gain equal to one. The relation between γ_2 and the weighting coefficient w is illustrated in Fig. 5.6.

Remark 5.3. The minimum value of quality criterion $J_2 = 0.5\mathbf{e}^{\mathrm{T}}(0)\mathbf{K}\mathbf{e}^{\mathrm{T}}(0)$ is determined as $0.5y_{\mathrm{D}}^2 k_{11}$. Hence, substituting $k_{11} = k_{11}^{+}$ given by (5.51), we get

$$J_2\left(u_{\mathrm{opt}}\right) = \frac{y_{\mathrm{D}}^2}{4}\left[\sqrt{w\,(w+4)} + w\,(2n-1)\right]. \tag{5.65}$$

Stability Analysis

A discrete-time system is asymptotically stable if all the roots of the characteristic polynomial of its closed-loop state matrix $\mathbf{A}_{\mathbf{cl}} = [\mathbf{I}_n - \mathbf{b}(\mathbf{c}^{\mathrm{T}}\mathbf{b})^{-1}\mathbf{c}^{\mathrm{T}}]\mathbf{A}$ are located within the unit circle on the z-plane. The roots of the polynomial

$$\begin{aligned}\det\left(z\mathbf{I}_n - \mathbf{A}_{\mathbf{cl}}\right) &= z^n + \frac{c_{n-1}-c_n}{c_n}z^{n-1} + \frac{c_{n-2}-c_{n-1}}{c_n}z^{n-2} + \cdots + \frac{c_1-c_2}{c_n}z,\\ &= z^n + (\gamma-1)\,z^{n-1} = z^{n-1}\left[z-(1-\gamma)\right],\end{aligned} \tag{5.66}$$

where γ is either γ_1 or γ_2, are located inside the unit circle if $0 < \gamma < 2$. Since for every n and for every w both γ_1 and γ_2 satisfy the condition $0 < \gamma \leq 1$, the system is asymptotically stable. Moreover, since irrespective of the value of n and w the roots of (5.66) remain on the nonnegative real axis, no oscillations appear at the output. By changing w from 0 to ∞, the nonzero pole moves towards the origin of the

z-plane, which results in faster convergence to the demand state. In the limit case when $w = \infty$, all the closed-loop poles are at the origin ensuring the fastest achievable response of a linear controller in discrete-time systems offered by a dead-beat scheme.

5.1.2.2 Properties of the Proposed Controller

At any time instant $kT \geq 0$, the amount of data to be delivered by the source according to the command of the SM controller with LQ optimal sliding plane can be rewritten in the following way:

$$u(kT) = \gamma \left[y_{\mathrm{D}} - y(kT) - \sum_{j=k-n_p}^{k-1} u(jT) \right], \tag{5.67}$$

where controllers (5.37) and (5.62) obtained in Case 1 and 2, respectively, differ in the choice of the gain constant γ. Control law (5.37) assumes $\gamma = \gamma_1 = (\sqrt{4n+1} - 1)/2n$, whereas controller (5.62) uses $\gamma = \gamma_2 = (\sqrt{w(w+4)} - w)/2$.

In the further part of this section, the properties of the developed control algorithm (5.67) will be formulated as three theorems, and strictly proved. The first theorem specifies the memory requirements for the buffer at the bottleneck node which guarantee loss-free transmission. The second proposition imposes the constraint on the demand queue length necessary to obtain full resource usage in the network. Finally, the third theorem states that the transfer speed assigned to the source is always nonnegative and bounded, which is a critical prerequisite in the design of feasible network controllers.

Theorem 5.1. *If controller* (5.67) *is applied to system (5.7)–(5.10), then the queue length in the bottleneck node buffer is always upper-bounded, i.e.,*

$$\underset{k \geq 0}{\forall}\; y(kT) \leq y_{\mathrm{D}}. \tag{5.68}$$

Proof. The bottleneck node buffer is empty for any $kT \leq \mathrm{RTT} = n_p T$. Hence, it suffices to show that the proposition is satisfied for any $k > n_p$. Let us assume that for some integer $l > n_p$, $y(lT) \leq y_{\mathrm{D}}$. We will demonstrate that the theorem is also true for $l + 1$.

Substituting (5.6) into (5.67), we get

$$\begin{aligned} u(lT) &= \gamma \left[y_{\mathrm{D}} - \sum_{j=0}^{l-n_p-1} u(jT) + \sum_{j=0}^{l-1} h(jT) - \sum_{j=l-n_{\mathrm{p}}}^{l-1} u(jT) \right] \\ &= \gamma \left[y_{\mathrm{D}} - \sum_{j=0}^{l-1} u(jT) + \sum_{j=0}^{l-1} h(jT) \right]. \end{aligned} \tag{5.69}$$

Using (5.3), the queue length at the $(l+1)T$ time instant can be expressed as

$$y\left[(l+1)T\right] = y\left(lT\right) + u\left[\left(l-n_{\mathrm{p}}\right)T\right] - h\left(lT\right). \tag{5.70}$$

Applying (5.6) and (5.69), we get

$$\begin{aligned} y\left[(l+1)T\right] &= \sum_{j=0}^{l-n_p-1} u\left(jT\right) - \sum_{j=0}^{l-1} h\left(jT\right) \\ &\quad + \gamma\left[y_{\mathrm{D}} - \sum_{j=0}^{l-n_p-1} u\left(jT\right) + \sum_{j=0}^{l-n_p-1} h\left(jT\right)\right] - h\left(lT\right) \\ &= (1-\gamma)\left[\sum_{j=0}^{l-n_p-1} u\left(jT\right) - \sum_{j=0}^{l-1} h\left(jT\right)\right] \\ &\quad + \gamma y_{\mathrm{D}} - \gamma \sum_{j=l-n_p}^{l-1} h\left(jT\right) - h\left(lT\right) \\ &= (1-\gamma)\, y\left(lT\right) + \gamma y_{\mathrm{D}} - \gamma \sum_{j=l-n_p}^{l-1} h\left(jT\right) - h\left(lT\right). \end{aligned} \tag{5.71}$$

After adding and subtracting y_{D}, the term rearrangement in (5.71) leads to

$$y\left[(l+1)T\right] = y_{\mathrm{D}} - (1-\gamma)\left[y_{\mathrm{D}} - y\left(lT\right)\right] - \gamma \sum_{j=l-n_p}^{l-1} h\left(jT\right) - h\left(lT\right). \tag{5.72}$$

Since $0 < \gamma \leq 1$ and $h(\cdot)$ is always nonnegative, $y[(l+1)T] \leq y_{\mathrm{D}}$. Thus, using the principle of the mathematical induction, we conclude that the proposition is valid for any time instant $kT \geq 0$. This ends the proof. □

It comes from Theorem 5.1 that if for the considered VC the buffer of size y_{D} is assigned at the bottleneck node, then no data will be lost in the network as a result of congestion. Apart from low loss rate, successful flow control strategies for modern telecommunication systems are expected to achieve high level of resource utilization. The proposition formulated below indicates how the demand queue length should be selected so that the available bandwidth at the output link of the bottleneck node will be always entirely consumed by the data stream in the analyzed VC.

Theorem 5.2. *If controller (5.67) is applied to system (5.7)–(5.10), and the demand queue length satisfies*

$$y_{\mathrm{D}} > d_{\max}\left(n_p + 1/\gamma\right), \tag{5.73}$$

then for any $k \geq n_p + 1$, the queue length is strictly positive.

Proof. It follows from (5.6) and (5.71) that when y_{D} is selected according to (5.73), then $y(k = n_p + 1) > 0$. Let us assume that for some integer $l > n_p + 1$, the queue length is positive. We shall demonstrate that $y[(l+1)T]$ is also greater than zero. Since $\gamma \in (0, 1]$, then from (5.71), we get

$$\begin{aligned} y\left[(l+1)T\right] &= (1-\gamma)\, y(lT) + \gamma y_{\mathrm{D}} - \gamma \sum_{j=l-n_p}^{l-1} h(jT) - h(lT) \\ &\geq \gamma y_{\mathrm{D}} - \gamma \sum_{j=l-n_p}^{l-1} h(jT) - h(lT). \end{aligned} \tag{5.74}$$

It follows from (5.2) that for any time instant kT, $0 \leq h(kT) \leq d_{\max}$. Using assumption (5.73), we obtain

$$y\left[(l+1)T\right] \geq \gamma\left[y_{\mathrm{D}} - d_{\max}\left(n_p + 1/\gamma\right)\right] > 0. \tag{5.75}$$

Since l was chosen arbitrarily (greater than $n_p + 1$), the proposition is valid for any $k \geq n_p + 1$. This concludes the induction proof. □

Any properly designed flow control algorithm should guarantee that the assigned transmission rate is always nonnegative and bounded. The next theorem shows that the rate established by the analyzed algorithm is indeed greater than or equal to zero and that it is limited from above by a (precisely determined) finite value.

Theorem 5.3. *If controller (5.67) is applied to system (5.7)–(5.10), then the transmission rate generated by this controller is always nonnegative and bounded:*

$$\underset{k\geq 0}{\forall}\ 0 \leq u(kT) \leq \max\left\{\gamma y_{\mathrm{D}}, d_{\max}\right\}. \tag{5.76}$$

Proof. For $k = 0$, we have $u(0) = \gamma y_{\mathrm{D}}$, which means that since $\gamma > 0$ and $y_{\mathrm{D}} > 0$ the theorem holds at the initial time. Let us assume that (5.76) is true for some integer $l > 0$. We will prove that the proposition is valid also for $l + 1$. Using (5.69), we can present $u[(l+1)T]$ in the following way:

$$\begin{aligned} u\left[(l+1)T\right] &= \gamma\left[y_{\mathrm{D}} - \sum_{j=0}^{l} u\left(jT\right) + \sum_{j=0}^{l} h\left(jT\right)\right] \\ &= \gamma\left[y_{\mathrm{D}} - \sum_{j=0}^{l-1} u\left(jT\right) + \sum_{j=0}^{l-1} h\left(jT\right)\right] - \gamma\left[u\left(lT\right) - h\left(lT\right)\right] \\ &= u\left(lT\right) - \gamma u\left(lT\right) + \gamma h\left(lT\right) = \left(1-\gamma\right)u\left(lT\right) + \gamma h\left(lT\right). \end{aligned} \tag{5.77}$$

Since $\gamma \in (0, 1]$ and for any lT, $0 \le h(lT) \le d_{\max}$, then $0 \le u[(l+1)T] \le \max(\gamma y_{\mathrm{D}}, d_{\max})$. This ends the induction proof. □

Theorems 5.1–5.3 define the most important properties of the designed control strategy related to the communication system stability (nonnegative and bounded control signal, and finite queue length) and the network resource utilization (strictly positive queue length). In the remainder of Sect. 5.1.2, these features will be verified in a series of simulation tests.

5.1.2.3 Simulation Results

The model for simulation tests has been constructed according to the description given in Sect. 5.1.1. The network parameters are chosen in the following way: discretization period $T = 10$ ms, round-trip time $\mathrm{RTT} = n_p T = 100$ ms, and the maximum available bandwidth $d_{\max} = 10$ packets per period. Hence, the system order $n = n_p + 1 = 11$. Two series of simulation tests are conducted. In Test 1, the controller performance is verified in the situation when the available bandwidth exhibits sudden changes of large amplitude. Such scenario reflects the most adverse networking conditions and is well suited to corroborate the extreme signal values indicated by the theorems stated in the previous section. In Test 2, in turn, the system behavior is investigated in the presence of stochastic bandwidth variations, which are typical for many real-life networking scenarios. In both tests, several simulations are run for controller (5.67) with different gain settings. Its properties are compared with the on-off controller (4.9) adapted for the case of finite sampling considered here. The maximum rate for the on-off controller is set as $u_{\max} = 11$ packets $> d_{\max}$.

Test 1. In the first simulation example, the operation of the designed SM controller is verified in response to the available bandwidth depicted in Fig. 5.7.

The gain of the controller obtained in the first optimization procedure is set according to (5.33) as $\gamma_1 = (\sqrt{4 \cdot 10 + 1} - 1)/2 \cdot 10 = 0.259$. According to the choice of the tuning coefficient, we may consider various gain settings for the controller developed in the second optimization procedure. We will analyze the controller performance for $w = 1$ with the gain calculated according to (5.58) as 0.618. Notice that the gain $\gamma_1 = 0.259$ of the controller obtained in the first

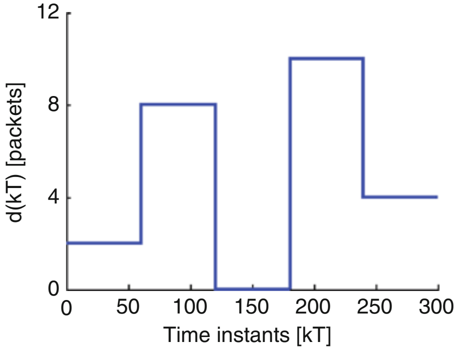


Fig. 5.7 Available bandwidth

Table 5.1 Controller parameters in Test 1

Controller	Weighting factor w	Controller gain γ	Demand queue length y_{D} [packets]
Linear controller (5.67)	0.09	0.259	$140 > 139$
	1	0.618	$120 > 116$
Nonlinear controller (4.9)	–	∞	$111 > 110$

optimization procedure corresponds to the case of $w = 0.09$ as analyzed in the context of the second optimization problem. The demand queue length y_{D} is adjusted according to the guidelines stated in Theorem 5.2. The gain and y_{D} settings are summarized in Table 5.1. In addition, we run the test for the nonlinear control law presented in Chap. 4 with y_{D} adjusted according to (4.15) as 111 packets $> u_{\max} n_p = 11 \cdot 10 = 110$ packets. This value is also stated in Table 5.1.

The transmission rate generated by the controllers is illustrated in Figs. 5.8 and 5.9 and the resulting queue length in Fig. 5.10. It can be seen from the plots in Figs. 5.8 and 5.9 that the rates generated by the linear controller (5.67) are nonnegative and bounded as indicated by Theorem 5.3. For the nonlinear control law (4.9), the established control signal is confined to the interval $[0, u_{\max}] = [0,$ 11 packets]. In each case, the queue length does not increase beyond the demand value and never drops to zero (for $kT \geq (n_p + 1)T = 11T$). This means that the buffer capacity is not exceeded, and all of the available bandwidth is used for data transfer. In consequence, the maximum throughput in the network is achieved. Moreover, since the gain of linear controller depends on the weighting coefficient, the choice of w influences the system dynamics. When w increases, the controller reacts faster to the fluctuations of available bandwidth, and as w is reduced, responsiveness to the changes drops. Furthermore, according to Theorems 5.1 and 5.2, the increase of w allows for allocation of smaller buffers while preserving the benefits of loss-free transmission. However, placing more impact on the output error elimination (large w) incurs bigger values of the initial source transfer speed, which can be too high for slow transmitters. Moreover, we can notice from the rate evolution in the initial phase shown in Fig. 5.9 that with the increase of w the smoothness of rate adjustments degrades, which is disadvantageous for the transmission consistency. Therefore, for a majority of practical applications, the weighting factor $w = 1$ would

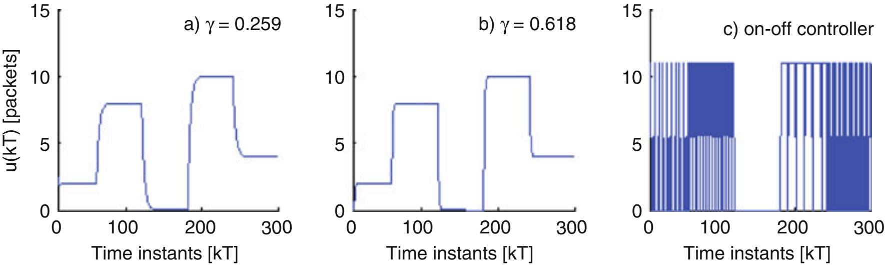
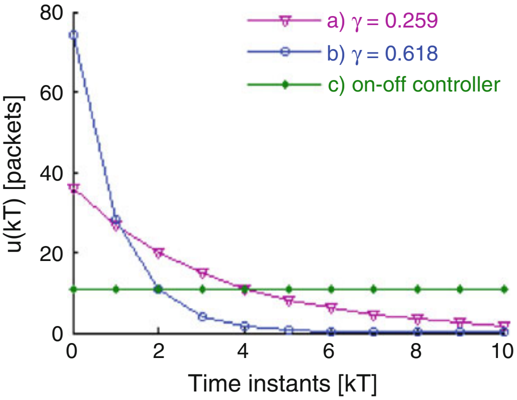


Fig. 5.8 Transmission rate linear controller *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, *c* nonlinear controller

Fig. 5.9 Initial transfer rate: linear controller *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, *c* nonlinear controller



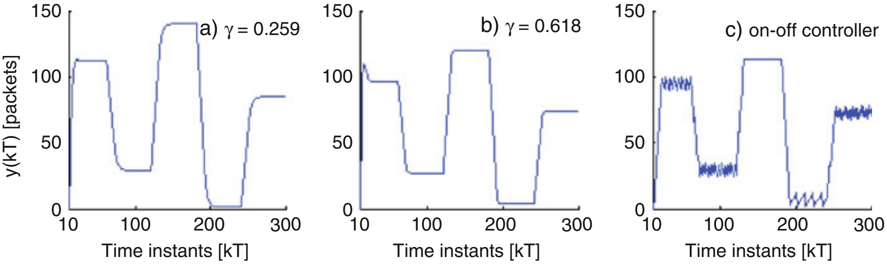


Fig. 5.10 Queue length: linear controller *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, *c* nonlinear controller

offer a fair trade-off between good system dynamics and smoothness of moderate transmission rates. Notice that according to (5.58), setting $w = 1$ corresponds to the gain $\gamma = (\sqrt{5} - 1)/2$. Consequently, the gain reciprocal (the system time constant $1/\gamma$) constitutes the golden ratio $(\sqrt{5}+1)/2$. We will refer to this dynamical configuration as the “golden-ratio controller.”

Figure 5.11 shows the evolution of the sliding variable. Notice that in the case of the controller with LQ optimal sliding plane (graphs a and b), $s(\cdot)$ immediately

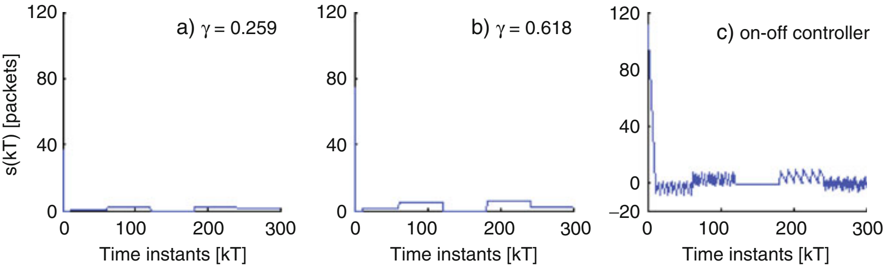


Fig. 5.11 Sliding variable: linear *a* $\gamma = 0.259$, *b* $\gamma = 0.618$; *c* nonlinear controller

decreases from its original value $s(0) = \gamma y_D$ to a relatively narrow band $s(\cdot) \in [0, \gamma c_n d_{\max} = \gamma d_{\max})$ and always remains in this band, which provides a clear evidence of a properly established quasi-sliding motion in a discrete-time system. In the case of the on-off controller (4.9), the reaching phase is extended over several periods; however, once the system representative point approaches the vicinity of the sliding plane $s(\cdot) = 0$, it stays in the band around the plane for all subsequent time. Comparing the plot in Fig. 5.11c with the one representing the sliding variable for the continuous-time system presented in Fig. 4.6, we can notice high-frequency switching around the plane, which occurs due to the finite sampling rate in discrete-time systems.

We also compare the controllers with respect to performance indices J_1 and J_2. Two tests are conducted: (i) with $y_D = 140$ packets adjusted the same for all controllers and (ii) with y_D set according to the values listed in Table 5.1. Consequently, in test (i), we verify the controller performance with the same control objective set for all the algorithms (to stabilize $y(\cdot)$ at the level of 140 packets), while in test (ii), we compare the indices for different control objectives (different values of y_D for each controller). The simulations are run in the absence of the disturbance, i.e., with $d(\cdot) \equiv 0$. We summarize the results for cases (i) and (ii) appropriately in Tables 5.2 and 5.3. For the LQ optimal SM controller, the choices of w (listed as the column headings in the second row) determine both the gain adjustment defined in Table 5.1 and J_2 tuning as defined in (5.41). For the nonlinear controller, the value indicated in the column heading reflects the value of w used to compute J_2. As expected, the controller with $w = 0.09$ achieves the smallest value of performance index J_1 (in case (i)), as this setting corresponds to the optimal gain $(\sqrt{4n+1} - 1)/2n$ with respect to J_1. However, this is no longer the case when a different control objective is set for each controller (case (ii)). In such circumstances, the golden-ratio controller with $w = 1$ achieves a smaller J_1 value and outperforms the first controller (with $w = 0.09$). This is due to much lower y_D level which allows for substantial reduction of the sum encountered in J_1. It follows from the tables that index J_2 grows as w is increased. Therefore, we may expect larger control effort for more responsive controllers. The nonlinear control law outperforms the optimal one in terms of J_1

Table 5.2 Performance index for identical y_D setting (case (i))

	LQ optimal controller		Nonlinear controller	
w	0.09	1	0.09	1
J_1	135,770	157,680	154,503	154,503
J_2	12,234	113,859	13,991	146,638

Table 5.3 Performance index for minimum y_D setting (case (ii))

	LQ optimal controller		Nonlinear controller	
w	0.09	1	0.09	1
J_1	135,770	115,847	106,779	106,779
J_2	12,234	83,649	19,338	100,123

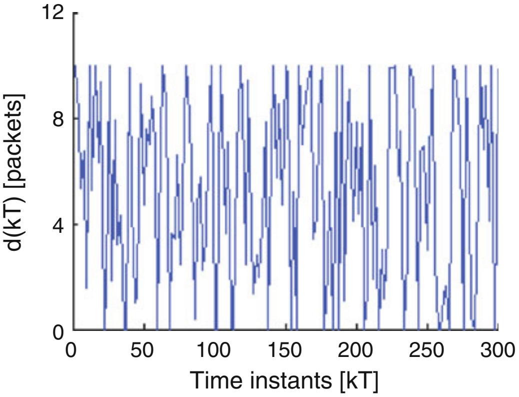


Fig. 5.12 Available bandwidth following normal distribution $D_{\mathrm{norm}}(5, 5)$

for $w = 1$ in test (i) and for each value of w in test (ii). However, it shows worse performance in terms of index J_2 when compared with the equivalent dynamical setting of the LQ optimal controller.

Test 2. In the second scenario, we investigate the behavior of controller (5.67) in the presence of highly variable stochastic bandwidth. Function $d(kT)$ following the normal distribution with mean $d_\mu = 5$ packets and standard deviation $d_\delta = 5$ packets, $D_{\mathrm{norm}}(5, 5)$, is illustrated in Fig. 5.12. In the first two simulations (curves a) and b) in the graphs), we apply the same controller parameters as in Test 1. However, since the mean available bandwidth in the stochastic pattern significantly differs from the maximum value, in the third simulation (curve c)), we adjust the demand queue length according to (5.73) with $d_{\max}$ replaced by $d_\mu = 5$ packets. With the gain 0.618, we have $y_D = 60 > 58$ packets in the third simulation run.

The generated transmission rates are depicted in Fig. 5.13 and the buffer occupancy in Fig. 5.14. We can see from the graphs in Fig. 5.13 that function $u(kT)$ established by the controller is nonnegative and bounded, thus ensuring feasible input rate adjustment at the data source. Moreover, one can notice that by decreasing the gain, the favorable rate smoothening in the presence of high-frequency bandwidth oscillations is obtained. Consequently, reduced responsiveness gives the benefit of rate signal which has a better chance of being reproduced by the source. Thus, the control fidelity in the actual system may be improved.

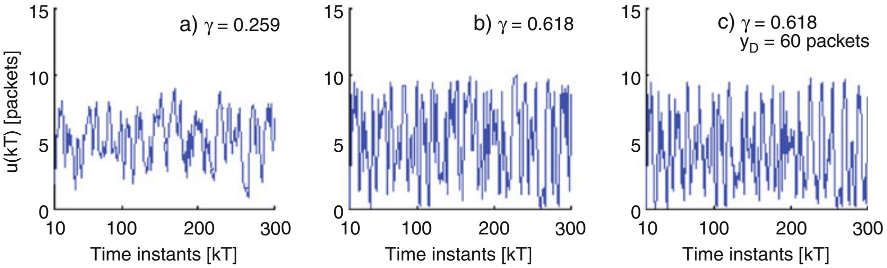


Fig. 5.13 Transmission rate: *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, *c* reduced y_D, $\gamma = 0.618$

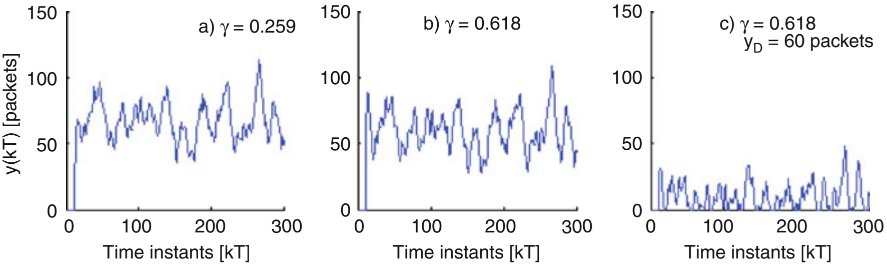


Fig. 5.14 Queue length: *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, *c* reduced y_D, $\gamma = 0.618$

The plots in Fig. 5.14 demonstrate that for each parameter, setting the queue length never grows beyond the demand value, which yields zero loss rate at the bottleneck node. In cases (a) and (b), $y(\cdot)$ is strictly positive (following the initial phase), which means that all of the available bandwidth is effectively utilized for the data transfer. In case (c), though, the queue length occasionally drops to zero, and certain part of the available bandwidth is left unused. Consequently, the obtained buffer savings (from the reduced value of y_D) come at the price of decreased bandwidth utilization which drops to 87%. The lost opportunities for packet transfer at the output interface due to empty buffer are shown in Fig. 5.15.

The evolution of the sliding variable is illustrated in Fig. 5.16. We can see from the plots that the system representative point reaches the vicinity of the sliding plane $s(\cdot) = 0$ and stays in this vicinity for all subsequent time. Thus, the stability of the sliding motion is ensured despite the presence of highly variable mismatched disturbance $d(\cdot)$.

5.1.3 Methods for Constraining Excessive Initial Flow Rates

A possible drawback of the SM controller with LQ optimal sliding plane presented in the previous section is the high initial flow rate that is required to quickly bring the communication system into the region of full bandwidth utilization (and

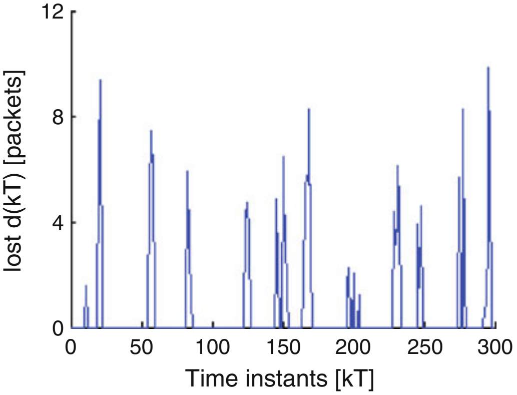


Fig. 5.15 Missed opportunities for packet transfer when $y_{\mathrm{D}} = 60$ packets ($\gamma = 0.618$)

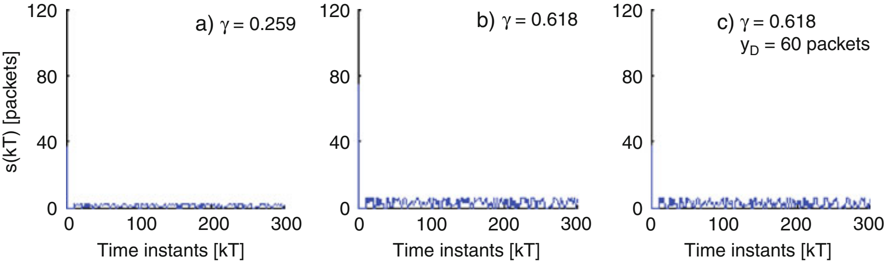


Fig. 5.16 Sliding variable: *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, *c* reduced y_{D}, $\gamma = 0.618$

consequently maximum throughput). It is clear from the plots shown in Fig 5.9 that with the increase in the controller gain (faster dynamics), the rate signal generated by the control algorithm grows and can be difficult to follow by rate-constrained sources. Therefore, it is desirable to throttle high input signals in the initial phase of transmission while maintaining good responsiveness to the changes of network state afterwards. Below, we present three methods for reducing excessive input signals without downgrade in the response speed to bandwidth fluctuations. The first method involves the use of a *time-varying sliding plane*, the second one is based on the so-called *reaching-law* approach, and the third technique employs rate clamping by means of a *saturation element*. The first method gives direct control over the duration of the initial phase and indirect control over the maximum transmission rate value. The other two approaches, in turn, place an explicit limitation on the maximum input signal and implicitly regulate the duration of the initial phase.

5.1.3.1 Application of a Time-Varying Hyperplane

The fundamental objective of sliding-mode control is to steer the system in such a way that its representative point is brought on the sliding surface (or to its close vicinity) in each successive control period. The control algorithm presented in

Sect. 5.1.2 uses a stationary plane, defined by $s(kT) = 0$ for all k. Such choice of the sliding plane with the requirement that the controller brings the representative point onto the plane in a minimal number of steps (in Sect. 5.1.2 assumed one period) usually implies high input signal at the beginning of the control process. This is due to the fact that significant control effort is required to cover the (large) initial distance from the plane. To alleviate this problem, one can introduce a time-varying plane which moves together with the system representative point towards the final, time-invariant position. As the control action is supposed to keep the representative point on the plane, and not to bring the point immediately onto the plane in its final position, the control effort $u(\cdot)$ may be reduced. In what follows, we discuss the selection of parameters of time-varying plane and the choice of plane dynamics so that desired system performance can be achieved.

Controller Design

In order to avoid excessive input signal magnitude at the beginning of the control process, we introduce a time-varying sliding hyperplane instead of the fixed one as was considered so far in this chapter. The plane is constructed in such a way that initially the system representative point belongs to the plane. Afterwards, the plane advances monotonically towards the origin of the error state space and stops moving after a predetermined time $k_{\mathrm{VP}}T$. Then, it remains fixed for the rest of the control process. The controller should track the plane position and keep the system representative point on the plane (or in its immediate vicinity) for all k. The design procedure consists of two steps. First, we decide on the plane dynamics so that smooth and stable movement to the plane final position is guaranteed. In order to ensure that the system error is reduced to zero in finite time, the plane in its final position needs to pass through the origin of the error state space. In the second step, we choose parameters of the plane so that fast system response to the changes in network state is ensured in the sliding phase.

Step 1. In this first part of the design, we choose the sliding plane dynamics. For any $k \geq 0$, the moving plane can be described by the following equation:

$$s\,(kT) = \mathbf{c}^{\mathrm{T}}\mathbf{e}\,(kT) + f\,(kT) = 0, \tag{5.78}$$

where $\mathbf{c}^{\mathrm{T}} = [c_1 \;\; c_2 \;\; c_3 \;\; \ldots \;\; c_n]$ is such a vector that $\mathbf{c}^{\mathrm{T}}\mathbf{b} \neq 0$, $\mathbf{e}(kT) = \mathbf{x_d} - \mathbf{x}(kT)$ denotes the system error, and $f(kT)$ is an *a priori* known function describing the plane advancement towards the final position $\mathbf{c}^{\mathrm{T}}\mathbf{e}(kT) = 0$. Function $f(\cdot)$ satisfies the following conditions:

- $f(0) = -\mathbf{c}^{\mathrm{T}}\mathbf{e}(0),$ (5.79)

- $f\,(kT)$ is strictly monotonic in the interval $[0, k_{\mathrm{VP}}T]\,,$ (5.80)

- $f\,(kT) = 0$ for any $k \geq k_{\mathrm{VP}}.$ (5.81)

Equation (5.79) implies that at the initial time the system representative point belongs to the sliding hyperplane (5.78). Conditions (5.80) and (5.81), in turn, guarantee that the plane reaches the origin of the error state space in finite time, $k_{\mathrm{VP}}T$, and it stops moving afterwards. Function $f(\cdot)$ and constant k_{VP} are selected in such a way that, on the one hand, the desired system dynamics is achieved, and on the other hand, the controller is able to follow the plane advancement without violating the transmission rate limitations. One possible definition of $f(kT)$ is

$$f\,(kT) = f_a\,(kT) = \begin{cases} -\frac{k-k_{\mathrm{VP}}}{k_{\mathrm{VP}}}\mathbf{c}^{\mathrm{T}}\mathbf{e}(0) & \text{for} \quad k = 0, 1, \ldots, k_{\mathrm{VP}}, \\ 0 & \text{for} \quad k > k_{\mathrm{VP}}; \end{cases} \tag{5.82}$$

which represents the movement towards the origin with constant velocity. Other choices of $f(kT)$ could be

$$f\,(kT) = f_b\,(kT) = \begin{cases} (-1)^{\rho_b+1}\left(\frac{k-k_{\mathrm{VP}}}{k_{\mathrm{VP}}}\right)^{\rho_b}\mathbf{c}^{\mathrm{T}}\mathbf{e}(0) & \text{for} \quad k = 0, 1, \ldots, k_{\mathrm{VP}}, \\ 0 & \text{for} \quad k > k_{\mathrm{VP}}, \end{cases}$$
$$\rho_b \in \mathbb{C}_+ \quad \text{and} \quad \rho_b > 1; \tag{5.83}$$

or

$$f\,(kT) = f_c\,(kT) = \begin{cases} \left[(k/k_{\mathrm{VP}})^{\rho_c} - 1\right]\mathbf{c}^{\mathrm{T}}\mathbf{e}(0) & \text{for} \quad k = 0, 1, \ldots, k_{\mathrm{VP}}, \\ 0 & \text{for} \quad k > k_{\mathrm{VP}}, \end{cases} \tag{5.84}$$
$$\rho_c \in \mathbb{C}_+.$$

Equation (5.83) corresponds to the plane movement with constant inclination and decreasing speed, and definition (5.84) reflects accelerated plane dynamics. The example plots of function $f(\cdot)$ given by (5.82), (5.83), and (5.84) are shown in Fig. 5.17, and the plane displacement in a hypothetical second-order system (x_1, x_2) for each case is illustrated schematically in Fig. 5.18 (note that in the second-order system the sliding hyperplane reduces to a line).

Although function $f(\cdot)$ can be selected in various ways, giving different properties in the initial phase of the control process, the most attractive choice out of possibilities (a)–(c) from the point of view of the transmission efficiency seems to be the linear one (5.82). This is due to the fact that constant plane velocity adjusted to maintain the maximum input signal will typically result in the largest permissible transmission rate in the initial phase. As a consequence, the linear plane dynamics will usually allow for a bigger number of packets to be sent at the beginning of transmission than in the case of function $f_b(\cdot)$ or $f_c(\cdot)$. This result, in turn, provides higher initial throughput and opens a possibility for completing the data transfer in a shorter time interval. We discuss these issues in Example 5.2.

Example 5.2. Let us analyze rate assignments in the interval [0, $k_{\mathrm{VP}}T$] in the presence of constraint $u_{\max} = 8$ packets per period. The controller tracks the position of time-varying plane and attempts to keep the system representative point on the

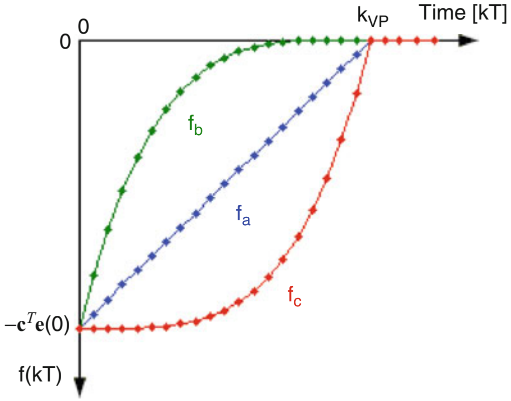


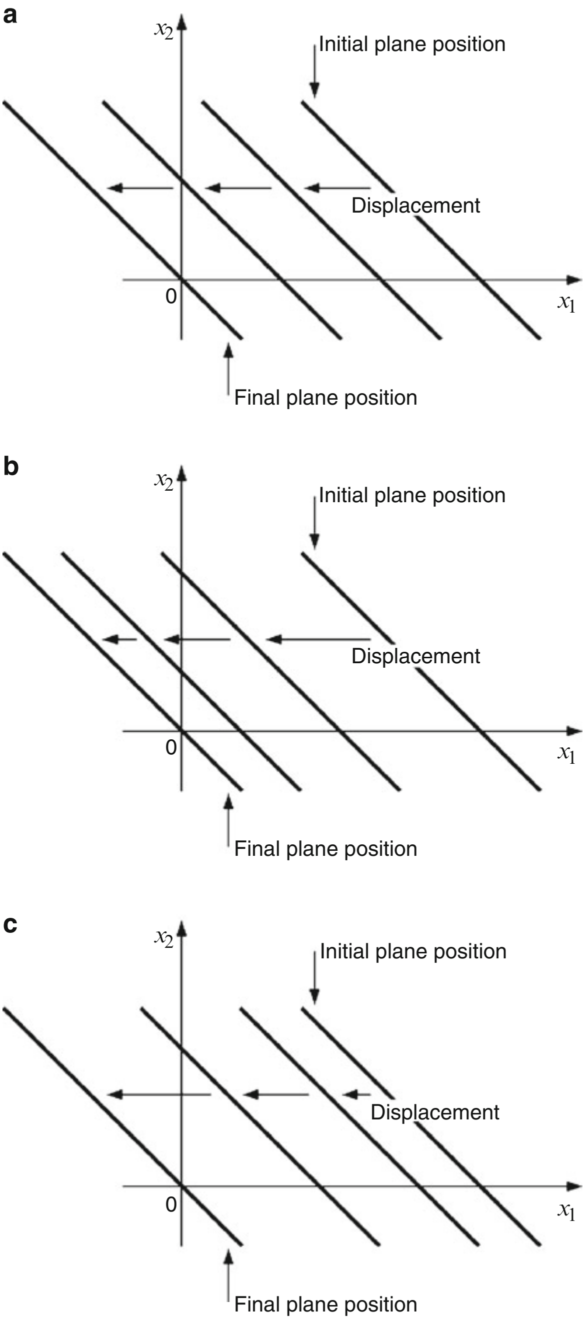
Fig. 5.17 Example choices of function $f(\cdot)$

plane for all time. For this purpose, it uses the information about the plane movement contained in the *a priori* known function $f(kT)$ and at each kth time instant generates such control signal that the representative point is on the plane at instant $(k+1)T$. Consequently, the current control action depends on the next position of the sliding hyperplane determined by $f[(k+1)T]$. Constant k_{VP} is selected so that the maximum rate $u_{\max}$ is never exceeded. We assume here $k_{\text{VP}} = 8$. For the purpose of exposition, it also assumed that the plane movement terminates within the first round-trip time; i.e., it is assumed that $k_{\text{VP}} < n_p$. In this way, the presence of the available bandwidth does not interfere with rate calculations in the initial phase. We study rate evolution for three choices of plane dynamics determined by the shape of $f(\cdot)$: (a) constant plane velocity described by linear function (5.82), (b) plane dynamics with a negative acceleration, and (c) plane dynamics with a positive acceleration. The rate assignment in each case is illustrated in Fig. 5.19.

Linear function (a) and the one characterized by negative acceleration (b) result in the maximum allowed rate assigned to the source at the instant $kT = 0$. In case a), the plane dynamics is constant in the interval $[0,\ k_{\text{VP}}T]$, which means that the maximum allowable rate can be maintained from 0 to $(k_{\text{VP}} - 1)T$. The decelerated plane movement (case b)) implies decreasing rate assignments in the interval $[0,\ k_{\text{VP}}T)$. This is easily explained if we notice that a smaller magnitude of the control signal is required to bring the system representative point onto the plane when the plane position changes in smaller steps at successive time instants. The plane dynamics characterized by positive acceleration, in turn, results in a growing rate signal in the analyzed interval $[0,\ k_{\text{VP}}T)$ with the maximum of $u_{\max}$ attained at instant $(k_{\text{VP}} - 1)T$. In this case, the plane displacement increases at each step, and the control signal of growing magnitude is necessary to keep the representative point on the plane. We may perceive this last case as chasing the plane which departs with increasing velocity.

As a consequence of various choices of plane dynamics, the overall number of packets permitted for sending into the network differs among the controllers considered in cases (a), (b), and (c). In the analyzed example, the number of packets

Fig. 5.18 Plane dynamics in second-order system (**a**) plane displacement with constant velocity (**b**) plane displacement with negative acceleration (**c**) plane displacement with positive acceleration



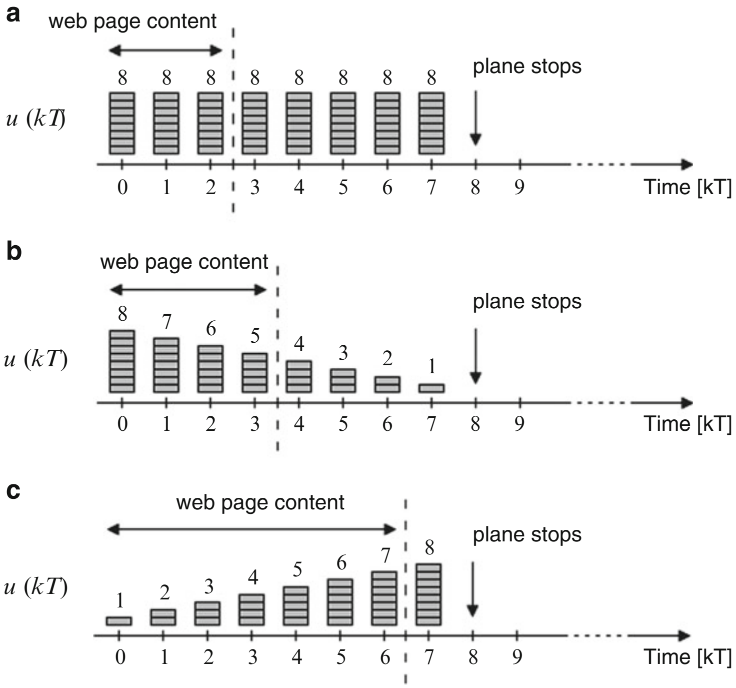


Fig. 5.19 Example rate function for different choice of $f(\cdot)$ (**a**) constant plane velocity (**b**) negative acceleration (**c**) positive acceleration

allowed into the network amounts to 64 in case (a) and 36 in cases (b) and (c). Notice also that from the point of view of the considered application (data transfer in a communication network), the accelerated plane movement is the least attractive solution. To see this fact, let us suppose that a user (the traffic destination) wants to retrieve web page content from a server operating at the source. We assume that the web page can be conveyed in 24 packets. Consequently, in the case of linear plane movement (a), three periods suffice to transmit the page, the decelerated case (b) requires four periods, whereas the accelerated plane movement prolongs the page content retrieval to seven periods. This clearly shows why the plane displacement with a constant velocity is favored over other choices of plane dynamics, (b) and (c), for typical transmission scenarios in communication networks.

Once the plane dynamics has been determined, the next step in the design of a flow rate controller is to select the elements of vector **c**, which defines the inclination of the plane $s(kT) = \mathbf{c}^{\mathrm{T}}\mathbf{e}(kT) = 0$. The plane parameters can be chosen in various ways as long as the condition $\mathbf{c}^{\mathrm{T}}\mathbf{b} \neq 0$ is met. For instance, one could apply the pole placement technique [11], or use LQ optimization, as it was considered in the previous section. However, in this section, we try a different approach to the plane design; namely, we choose the elements of vector **c** for a dead-beat scheme. In this way, the highest responsiveness to changing networking conditions will be achieved.

The problem of high control signal usually implied by dead-beat control in the initial phase is tackled by adjusting the plane dynamics through function $f(\cdot)$. The selected function $f(\cdot)$ ought to satisfy criteria (5.79)–(5.81).

Step 2. We intend to find such parameters of the plane which will ensure that the error is eliminated in the minimal number of steps after a change in the available bandwidth. Neglecting for the moment the effects of the disturbance, and substituting (5.7) into the equation describing the sliding plane at instant $(k+1)T$,

$$\mathbf{c}^{\mathrm{T}}\mathbf{e}\left[(k+1)\,T\right] + f\left[(k+1)\,T\right] = 0, \tag{5.85}$$

we get

$$\begin{aligned} &\mathbf{c}^{\mathrm{T}}\left\{\mathbf{x_d} - \mathbf{x}\left[(k+1)\,T\right]\right\} + f\left[(k+1)\,T\right] \\ &\quad = \mathbf{c}^{\mathrm{T}}\left\{\mathbf{x_d} - \mathbf{A}\mathbf{x}\,(kT) - \mathbf{b}u\,(kT)\right\} + f\left[(k+1)\,T\right] = 0, \end{aligned} \tag{5.86}$$

which leads to

$$u\,(kT) = \left(\mathbf{c}^{\mathrm{T}}\mathbf{b}\right)^{-1}\left\{\mathbf{c}^{\mathrm{T}}\left[\mathbf{x_d} - \mathbf{A}\mathbf{x}\,(kT)\right] + f\left[(k+1)\,T\right]\right\}. \tag{5.87}$$

The characteristic polynomial of the closed-loop state matrix with this control applied is determined as

$$\det\left(z\mathbf{I}_n - \mathbf{A_{cl}}\right) = z^n + \frac{c_{n-1} - c_n}{c_n}z^{n-1} + \frac{c_{n-2} - c_{n-1}}{c_n}z^{n-2} + \cdots + \frac{c_1 - c_2}{c_n}z. \tag{5.88}$$

Notice that the assumption $\mathbf{c}^{\mathrm{T}}\mathbf{b} \neq 0$ with vector $\mathbf{b}$ defined as $[0\ 0\ 0 \ldots\ 1]^{\mathrm{T}}$ (identity (5.8)) guarantees that $c_n \neq 0$, and relation (5.88) makes sense. For dead-beat control, the characteristic polynomial needs to satisfy $\det(z\mathbf{I}_n - \mathbf{A_{cl}}) = z^n$. This is achieved when, simultaneously,

$$\begin{aligned} c_{n-1} - c_n &= 0, \\ c_{n-2} - c_{n-1} &= 0, \\ &\ \vdots \\ c_2 - c_3 &= 0, \\ c_1 - c_2 &= 0, \end{aligned} \tag{5.89}$$

which implies

$$c_n = c_{n-1} = c_{n-2} = \cdots = c_3 = c_2 = c_1 \tag{5.90}$$

and vector describing parameters of the sliding plane

$$\mathbf{c}^{\mathrm{T}} = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \end{bmatrix} c_n . \tag{5.91}$$

Using (5.91) in (5.87), we get the following control law:

$$u(kT) = y_{\mathrm{D}} - x_1(kT) - \sum_{j=2}^{n} x_j(kT) + \frac{f[(k+1)T]}{c_n}. \tag{5.92}$$

Substituting $x_1(kT) = y(kT)$ and $x_j(kT) = u[(k-n+j-1)T]$ for $j = 2, 3, \ldots, n$, we can rewrite (5.92) as

$$u(kT) = y_{\mathrm{D}} - y(kT) - \sum_{j=k-n_{\mathrm{p}}}^{k-1} u(jT) + \frac{f[(k+1)T]}{c_n}. \tag{5.93}$$

The amount of data the source is allowed to transmit in each review period is calculated by the proposed nonlinear algorithm according to (5.93) and is accessible at the source T_{B} later. If we compare (5.93) with the LQ optimal controller (5.67), we can notice a similar structure of both algorithms. The first two terms account for the current error at the output, whereas the third term $\sum_{j=k-n_{\mathrm{p}}}^{k-1} u(jT)$ accumulates the information about the packets on route. Consequently, when the gain constant γ is set equal to 1 in Eq. (5.67), both algorithms are equivalent for $k \geq k_{\mathrm{VP}} - 1$.

This concludes the design of the flow control algorithm for the considered network. In the next section, we describe several important properties of the obtained control law and substantiate each property with a formal proof.

Properties of the Proposed Controller

The properties of the designed nonlinear controller (5.93) will be stated in a lemma and three theorems. The lemma specifies a relation between the control signal $u(kT)$ and the utilized bandwidth. Afterwards, the findings provided in the lemma are used to prove other controller properties, stated in the theorems. The first theorem shows that controller (5.93) always generates a nonnegative and bounded transmission rate. The second proposition specifies the conditions that must be satisfied to eliminate the risk of data loss (which could occur as a consequence of exceeding the bottleneck node buffer capacity). Finally, the third theorem provides a condition for full bandwidth utilization at the bottleneck link.

First, let us notice that

$$u(0) = y_{\mathrm{D}} + f(T)/c_n . \tag{5.94}$$

Then, for $k \geq 1$, the control signal satisfies the relation given in the following lemma.

Lemma 5.4. *If controller (5.93) is applied to system (5.7)–(5.10) with function f(·) satisfying conditions (5.79)–(5.81), then for any $k \geq 1$,*

$$u\,(kT) = h\,[(k-1)\,T] + \{f\,[(k+1)\,T] - f\,(kT)\}\,/c_n\,. \tag{5.95}$$

Proof. Substituting (5.6) into (5.93) yields

$$\begin{aligned} u\,(kT) =& y_{\mathrm{D}} - y\,(kT) - \sum_{j=k-n_p}^{k-1} u\,(jT) + f\,[(k+1)\,T]\,/c_n \\ =& y_{\mathrm{D}} - \sum_{j=0}^{k-n_p-1} u\,(jT) + \sum_{j=0}^{k-1} h\,(jT) - \sum_{j=k-n_p}^{k-1} u\,(jT) + f\,[(k+1)\,T]\,/c_n \\ =& y_{\mathrm{D}} - \sum_{j=0}^{k-1} u\,(jT) + \sum_{i=0}^{k-1} h\,(jT) + f\,[(k+1)\,T]\,/c_n\,. \end{aligned} \tag{5.96}$$

For $k = 1$, it follows immediately from (5.96) that

$$u(T) = y_{\mathrm{D}} - u(0) + h(0) + f\,(2T)\,/c_n \;= h(0) + [f\,(2T) - f(T)]\,/c_n\,, \tag{5.97}$$

which shows that (5.95) is indeed satisfied for $k = 1$. Let us assume that (5.95) is true for all integers up to some $l > 1$. Using this assumption, from (5.96), the transmission rate generated at instant $(l+1)T$ can be expressed as

$$\begin{aligned} u\,[(l+1)\,T] &= y_{\mathrm{D}} - \sum_{i=0}^{l} u\,(iT) + \sum_{i=0}^{l} h\,(iT) + f\,[(l+2)\,T]\,/c_n \\ &= y_{\mathrm{D}} - u(0) - \sum_{i=1}^{l} u\,(iT) + \sum_{i=0}^{l} h\,(iT) + f\,[(l+2)\,T]\,/c_n \\ &= -\frac{f(T)}{c_n} - \sum_{i=1}^{l} h\,[(i-1)\,T] - \frac{1}{c_n}\sum_{i=1}^{l}\{f\,[(i+1)\,T] - f\,(iT)\} \\ &\quad + \sum_{i=0}^{l} h\,(iT) + f\,[(l+2)\,T]\,/c_n \\ &= h\,(lT) + \{f\,[(l+2)\,T] - f\,[(l+1)\,T]\}\,/c_n\,. \end{aligned} \tag{5.98}$$

This ends the proof. □

Theorem 5.5. *If controller (5.93) is applied to system (5.7)–(5.10) with function $f(\cdot)$ satisfying conditions (5.79)–(5.81), then the transmission rate is always nonnegative and bounded, i.e., for any $k \geq 0$,*

$$0 \leq u\,(kT) \leq d_{\max} + y_{\mathrm{D}}. \tag{5.99}$$

Proof. Initially, the system representative point belongs to the sliding hyperplane, and according to (5.79), $f(0) = -\mathbf{c}^{\mathrm{T}}\mathbf{e}(0) = -c_n y_{\mathrm{D}}$. Since $f(\cdot)$ is assumed to be strictly monotonic and $f(k_{\mathrm{VP}}T) = 0$, the identity $f(0) = -c_n y_{\mathrm{D}}$ implies that for any $k \in [0;\, k_{\mathrm{VP}})$, $f(kT)$ and c_n have opposite signs. Consequently, for any $k \geq 0$, the following set of inequalities is satisfied:

$$-y_{\mathrm{D}} \leq \frac{1}{c_n} f\,(kT) \leq 0, \tag{5.100}$$

$$\text{and} \quad 0 \leq \frac{1}{c_n}\{f\,[(k+1)\,T] - f\,(kT)\} \leq y_{\mathrm{D}}. \tag{5.101}$$

It follows from the algorithm definition that $u(0) = y_{\mathrm{D}} + f(T)/c_n$. Hence, using (5.100), we get $u(0) \geq 0$ and $u(0) \leq y_{\mathrm{D}} \leq y_{\mathrm{D}} + d_{\max}$, which means that the theorem is satisfied for $k = 0$. On the other hand, for any $k > 0$, $u(kT)$ satisfies the relation given in Lemma 5.4. Since for any k the utilized bandwidth $h(kT)$ is nonnegative and bounded from above by $d_{\max}$, directly from (5.95) and (5.101), we obtain

$$0 \leq u\,(kT) = h\,[(k-1)\,T] + \{f\,[(k+1)\,T] - f\,(kT)\}\,/c_n \leq d_{\max} + y_{\mathrm{D}}. \tag{5.102}$$

This ends the proof. □

The next theorem shows that when the source injects packets into the network with the intensity regulated by the nonlinear control strategy (5.93), then the buffer queue length remains finite.

Theorem 5.6. *If controller (5.93) is applied to system (5.7)–(5.10) with function $f(\cdot)$ satisfying conditions (5.79)–(5.81), then the queue length is always upper-bounded by its demand value y_{D}.*

Proof. It follows from the system initial conditions that $y(0) = 0$. Since the control process commences at $kT = 0$ (the first nonzero rate signal is issued at $kT = 0$), due to the delay, the first packets arrive at the bottleneck node at $kT = \mathrm{RTT} = n_p T$. This means that according to (5.3), $y(kT) = 0$ for $k \leq n_p$, and it is sufficient to demonstrate the proposition for $k > n_p$.

Applying relation (5.6), we may present the queue length in the following form:

$$y\left(kT\right)=\sum_{j=0}^{k-n_p-1}u\left(jT\right)-\sum_{j=0}^{k-1}h\left(jT\right)=u\left(0\right)+\sum_{j=1}^{k-n_p-1}u\left(jT\right)-\sum_{j=0}^{k-1}h\left(jT\right)$$
$$=y_{\mathrm{D}}+\frac{1}{c_n}f\left(T\right)+\sum_{j=1}^{k-n_p-1}u\left(jT\right)-\sum_{j=0}^{k-1}h\left(jT\right). \tag{5.103}$$

Using Lemma 5.4, we get

$$y\left(kT\right)-y_{\mathrm{D}}=\frac{1}{c_n}f\left(T\right)+\sum_{j=1}^{k-n_p-1}\left\{h\left[\left(j-1\right)T\right]+\frac{1}{c_n}\{f\left[\left(j+1\right)T\right]\right.$$
$$\left.-f\left(jT\right)\}\right\}-\sum_{j=0}^{k-1}h\left(jT\right)$$
$$=\sum_{j=0}^{k-n_p-2}h\left(jT\right)-\sum_{j=0}^{k-1}h\left(jT\right)+\frac{1}{c_n}\left\{\sum_{j=1}^{k-n_p}f\left(jT\right)-\sum_{j=1}^{k-n_p-1}f\left(jT\right)\right\}$$
$$=-\sum_{j=k-n_p-1}^{k-1}h\left(jT\right)+\frac{1}{c_n}f\left[\left(k-n_p\right)T\right]. \tag{5.104}$$

Since function $h(\cdot)$ is always nonnegative, and $\forall k\, f(kT)$ and c_n have opposite signs, we conclude that $y(kT)$ given by (5.104) never exceeds the demand value. This ends the proof. □

Theorem 5.7. *If controller (5.93) is applied to system (5.7)–(5.10) with function $f(\cdot)$ satisfying conditions (5.79)–(5.81), and the demand queue length satisfies the following inequality:*

$$y_{\mathrm{D}}>d_{\max}\left(n_p+1\right), \tag{5.105}$$

then for any $k\geq k_{\mathrm{VP}}+n_p+1$, the queue length is strictly positive.

Proof. Notice that we deal with the time instants kT such that $k\geq k_{\mathrm{VP}}+n_p+1$. Consequently, in the considered time range function $f(\cdot)=0$. Since the utilized bandwidth represented by function $h(\cdot)$ is nonnegative and upper-bounded by $d_{\max}$, we obtain from (5.106) the following estimate of the queue length for $k\geq k_{\mathrm{VP}}+n_p+1$:

$$y\left(kT\right)=y_{\mathrm{D}}-\sum_{j=k-n_p-1}^{k-1}h\left(jT\right)\geq y_{\mathrm{D}}-d_{\max}\left(n_p+1\right). \tag{5.106}$$

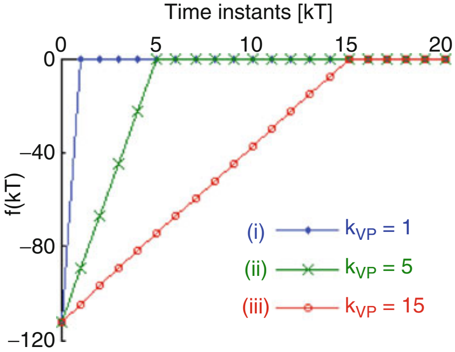


Fig. 5.20 Function $f(\cdot)$ used in the tests: (i) $k_{\mathrm{VP}} = 1$, (ii) $k_{\mathrm{VP}} = 5$, and (iii) $k_{\mathrm{VP}} = 15$

Therefore, using assumption (5.105), we get $y(kT) > 0$ for all $k \geq k_{\mathrm{VP}} + n_p + 1$. This completes the proof. □

The fundamental properties of the presented strategy stated in the theorems have been verified in a simulation scenario described in the following section.

Simulation Results

In order to facilitate the comparison with the other control algorithms described so far in this work, parameters of the network model are set identical to those defined in Sect. 5.1.2.3. Consequently, we select the discretization period $T = 10$ ms, the round-trip time $\mathrm{RTT} = n_p T = 10T$, and the maximum available bandwidth $d_{\max} = 10$ packets per period. We test performance of controller (5.93) with three choices of the terminal condition for function $f(\cdot)$ given by (5.82): (i) $k_{\mathrm{VP}} = 1$, (ii) $k_{\mathrm{VP}} = 5$, and (iii) $k_{\mathrm{VP}} = 15$. Notice that case (i) corresponds to a time-invariant plane and reflects the operation of linear controller (5.67) with the gain set equal to 1, i.e., the dead-beat scheme for the analyzed network. The evolution of $f(\cdot)$ for cases (i)–(iii) with $c_n = 1$, and $-\,\mathbf{c}^{\mathrm{T}}\mathbf{e}(0) = -c_n y_{\mathrm{D}} = -112$ packets, is illustrated in Fig. 5.20.

Two series of simulation tests are run: one for the bandwidth pattern illustrated in Fig. 5.7 and another for the stochastic bandwidth D_{norm}(mean, standard deviation) $= D_{\mathrm{norm}}$ (5, 5) depicted in Fig. 5.12.

Test 1. In order to guarantee full bandwidth usage, in the first series of simulations, we set y_{D} according to (5.105) as $112 > 110$ packets. The results of the test for the bandwidth from Fig. 5.7 are shown in Figs. 5.21, 5.22, and 5.23: the generated transmission rate in Fig. 5.21, the packet queue length in Fig. 5.22, and the sliding variable in Fig. 5.23. We can see from the plot in Fig. 5.21 that by extending the

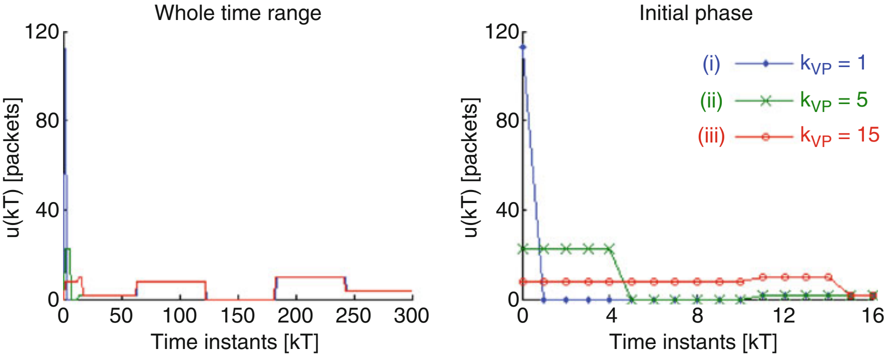
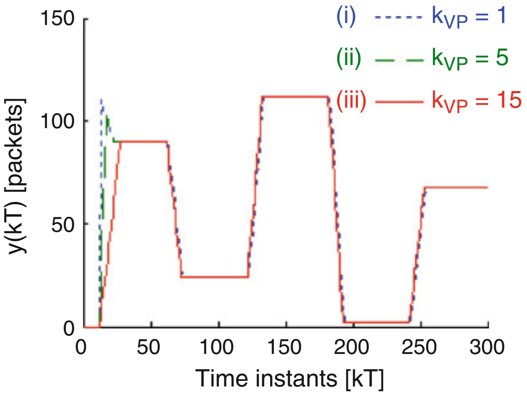


Fig. 5.21 Transmission rate: (i) $k_{\mathrm{VP}} = 1$, (ii) $k_{\mathrm{VP}} = 5$, and (iii) $k_{\mathrm{VP}} = 15$

Fig. 5.22 Queue length: (i) $k_{\mathrm{VP}} = 1$, (ii) $k_{\mathrm{VP}} = 5$, and (iii) $k_{\mathrm{VP}} = 15$



duration of the plane movement towards the final position (larger k_{VP}), we obtain a reduced initial rate value. For the fixed plane (marked as (i) $k_{\mathrm{VP}} = 1$ in the graph), the initial rate value equals 112 packets, whereas in case (ii), $k_0 = 5$, we get the initial rate of 22.4 packets, and in case (iii), $k_{\mathrm{VP}} = 15$, we obtain 7.5 packets. Following the primary phase of the control process, the curves overlap, and each controller provides the fastest reaction to the available bandwidth changes provided by a dead-beat scheme. The queue length evolution depicted in Fig. 5.22 demonstrates that the buffer size set equal to the demand queue length $y_{\mathrm{D}} = 112$ packets is not exceeded, which means that packet losses do not occur. Following the initial phase, the queue length does not fall to zero, implying that all of the available bandwidth is used for data transfer. Consequently, by applying the nonlinear control law with a time-varying plane instead of the linear controller with a fixed one, we have similar conditions for obtaining the maximum throughput, yet with a more realistic rate assignment.

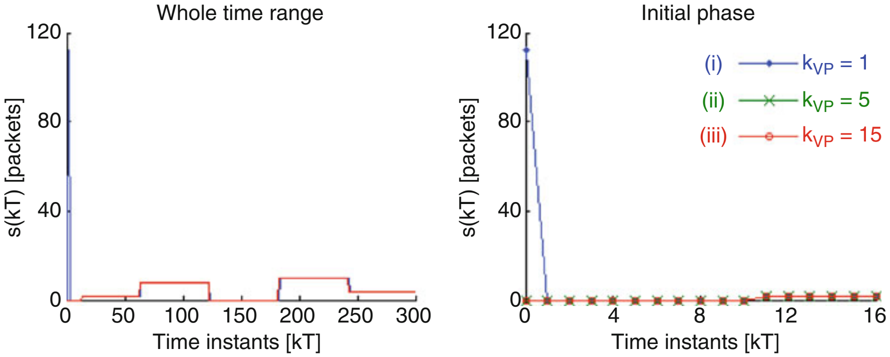


Fig. 5.23 Sliding variable: (i) $k_{\mathrm{VP}} = 1$, (ii) $k_{\mathrm{VP}} = 5$, and (iii) $k_{\mathrm{VP}} = 15$

It follows from Fig. 5.23 that the sliding variable, which reflects the distance from the sliding plane, is equal to zero in the case of the moving plane. This means that the system representative point is maintained on the plane precisely as it was assumed in the design. In the case of time-invariant plane (i), the initial large distance from the plane passing through the origin of the error state space is required to be covered in one period, which is the reason for high initial rate value. The three cases (i)–(iii) become equivalent for $k > 15$, resulting in a stable quasi-sliding motion in a discrete-time system perturbed by the mismatched exogenous signal $d(\cdot)$.

Test 2. In the second series of simulations, we verify the controller properties for the stochastic bandwidth pattern $D_{\mathrm{norm}}(5, 5)$ shown in Fig. 5.12. Since the mean bandwidth is much lower than the maximum one, in the tests, we use $d_{\mathrm{L}} = 5$ packets instead of $d_{\max} = 10$ packets in formula (5.105) and adjust y_{D} as $56 > 55$ packets. Consequently, the bandwidth may not always be entirely consumed by the stream of packets, but the buffer capacity will be reduced. The results of the test are shown in Figs. 5.24–5.27: the generated transmission rate in Fig. 5.24, the packet queue length in Fig. 5.25, the lost opportunities for data transfer in Fig. 5.26, and the sliding variable in Fig. 5.27.

Since the buffer is initially empty and the first packets arrive at the bottleneck node at $kT = n_pT = 10T$, no data is transferred at the output interface for $kT < 10T$. This means that the disturbance actually affects the system for $k \geq 10$, and in the interval [0, 9], the response of the system to the stochastic bandwidth is similar to the one considered in Test 1 and presented in Figs. 5.21, 5.22, and 5.23. The discrepancy in numerical results occurs due to a different y_{D} setting, assumed in Test 2 to be equal 56 packets, which is half of the value considered in Test 1. The initial rate in the case of the fixed plane (i) equals 56 packets, whereas for time-varying plane, it amounts to 11.2 packets (case (ii)) and 3.7 packets (case (iii)), respectively. The three controllers generate similar rate assignments in the time following the initial period. The controllers respond immediately to bandwidth

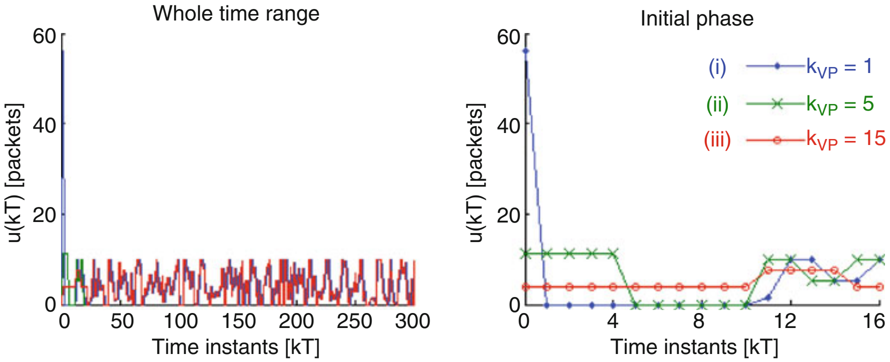


Fig. 5.24 Transmission rate: (i) $k_{VP} = 1$, (ii) $k_{VP} = 5$, and (iii) $k_{VP} = 15$

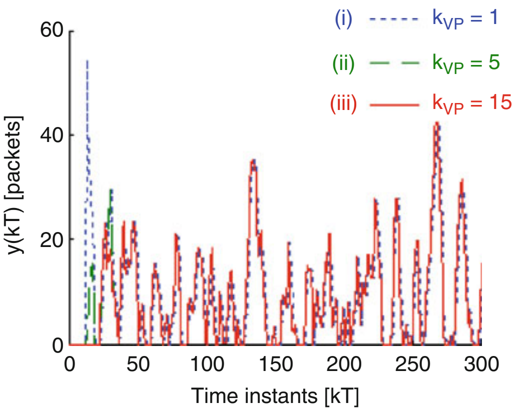


Fig. 5.25 Queue length: (i) $k_{VP} = 1$, (ii) $k_{VP} = 5$, and (iii) $k_{VP} = 15$

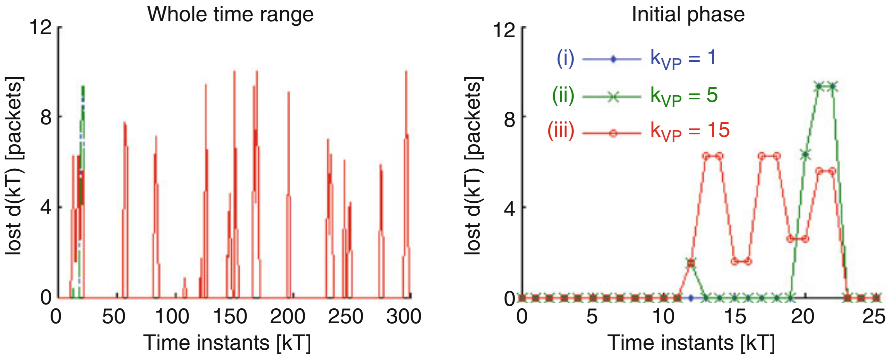


Fig. 5.26 Missed opportunities for transferring data: (i) $k_{VP} = 1$, (ii) $k_{VP} = 5$, and (iii) $k_{VP} = 15$

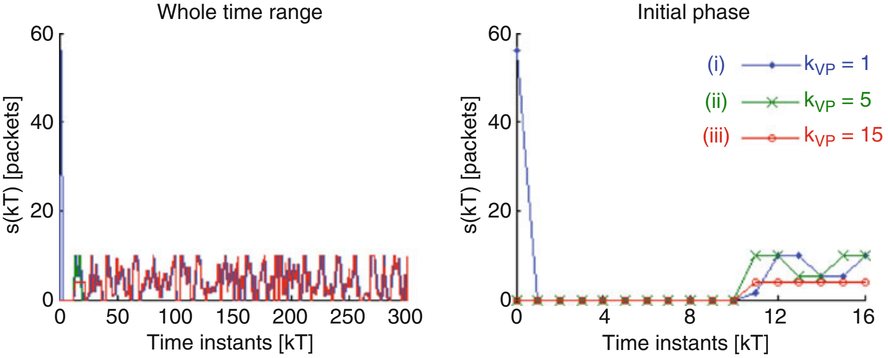


Fig. 5.27 Sliding variable: (i) $k_{\mathrm{VP}} = 1$, (ii) $k_{\mathrm{VP}} = 5$, and (iii) $k_{\mathrm{VP}} = 15$

variations. It follows from the queue length evolution presented in Fig. 5.25 that the buffer is depleted on several occasions ($y(kT) = 0$), which yields decreased level of bandwidth utilization 86% in cases (i) and (ii) and 84% in case (iii). The lost opportunities for transferring the data depicted in Fig. 5.26 indicate differences among the three controllers only for $k < 25$, which is the result of nonequal initial rate assignment. The lowered bandwidth utilization in Test 2, as compared to the case of the maximum throughput achieved in Test 1, is the effect of a trade-off between the bandwidth usage and buffer capacity savings.

5.1.3.2 Application of a Reaching Law

We showed in the previous section that by applying a nonlinear SM controller with a moving plane instead of the linear one with a fixed plane, we can obtain fast response to the changing networking conditions without generating excessive control signals. In this section, we will use an alternative method of SM controller design which will result in a similar set of advantageous properties related to handling the flow of data as controller (5.93) but will provide a direct control over the value of the input signal in the initial phase. The design procedure presented below is based on the so-called reaching-law approach [12]. In this approach, the sliding plane is fixed during the whole control process. In order to account for a possibly high initial magnitude of the control signal, it is no longer required to bring the system representative point onto the plane (or in its close vicinity) in a single step. Instead, the reaching phase is extended over several periods. In this way, large control effort needed to immediately overcome a significant distance from the plane, as was the case in Sect. 5.1.2, can be relieved. The controller employing the idea of reaching law steers the system dynamics so that the representative point approaches the plane by covering only a part of the initial distance from the plane in subsequent time intervals. Consequently, a smaller control signal suffices to ensure the desired

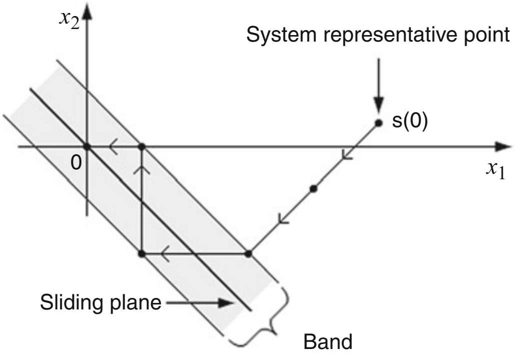


Fig. 5.28 Concept of reaching law proposed by Gao et al. [12]

system trajectory. In what follows, we will demonstrate that the favorable properties of the control strategies considered so far in this work can be achieved with a properly engineered reaching law.

The Concept of Reaching Law

The idea of reaching law in discrete-time systems originating from the seminal work of Gao et al. [12] is illustrated in Fig. 5.28 for the case of a hypothetical second-order system. The sliding plane (line in the analyzed system) remains fixed in the entire time span of the control process. The controller moves the system representative point towards the plane in the prescribed way (reaching phase) and maintains the point within a band around the plane afterwards (sliding phase). The way the point is supposed to approach the plane in the reaching phase, and to be kept in the vicinity of the plane in the sliding phase, is governed by reaching law. Mathematically, the reaching law is expressed as a function of the distance of the system representative point from the sliding plane. The law proposed by Gao et al. [12] brings the system representative point into the band around the plane and, provided that conditions given in [3] are satisfied, changes the point position from one side of the plane to the other in each successive control interval. This results in a quasi-sliding motion in the vicinity of the plane schematically sketched in Fig. 5.28.

The idea of Gao et al. proved inspiring for many researchers who proposed different approaches for the design of reaching law (see [23] for an excellent review of various solutions presented in a consistent framework). Two designs, developed in [4, 15], seem particularly attractive from the point of view of the application considered in this work. The idea introduced in [4], with a more efficient choice of the switching function reported later in [15], is based on the observation that moving the system representative point from one side of the plane to the other in each successive interval requires larger control effort than using a strategy aimed at keeping the point on the plane itself. This idea has been illustrated in Fig. 5.29.

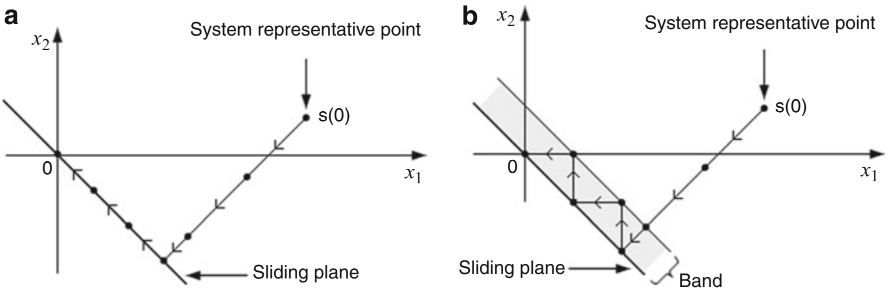


Fig. 5.29 Improved reaching law: (**a**) motion along sliding plane in unperturbed system, (**b**) motion confined to one side of sliding plane in system with perturbations

The situation shown in graph (a) in Fig. 5.29 reflects the case of a disturbance-free system, whereas graph (b) illustrates the system behavior in the presence of external perturbation. In the absence of disturbances, the improved reaching law maintains the point on the plane. When a perturbation affects the system dynamics, in turn, the system representative point may be pushed off the sliding plane, but then it is directed onto the plane (and not to its other side) at a subsequent control interval. Notice that since we are dealing with a discrete-time system, the corrective action of the controller that will move the point back to the plane takes effect with at least one period delay. Once the disturbance vanishes, the system representative point remains on the plane in a quasi-sliding motion. The reaching laws [4, 15], by avoiding the enforced switching at the boundaries of the band, give several important benefits in the considered system over the law of Gao et al. [12]. First of all, the magnitude of the control signal can be decreased since it takes less control effort to move the system representative point onto the plane than to its other side – a shorter distance needs to be covered in each step. Secondly, the degree of chattering can be reduced. Finally, probably the most important advantage of moving the representative point along the plane with regard to the application considered in this monograph is that the changes of polarity of the control signal may be eliminated. Indeed, if the disturbance is unipolar (assumes only nonpositive or only nonnegative values), the system representative point always remains in a band on one side of the plane only, as shown in Fig. 5.29b. This allows one to keep the network variables (such as the transmission rate, or packet queue length) in the feasible region of nonnegative values for all nonnegative bandwidth patterns $0 \le d(\cdot) \le d_{\max}$. From the control theory perspective, the discussed feature makes the system positive [19].

Controller Design

The dynamics of the controller with LQ optimal sliding plane developed in Sect. 5.1.2 is adjusted through a tuning coefficient w. By increasing w, one can obtain faster reaction to the changes in the available bandwidth and reduce the buffer capacity while maintaining the conditions of loss-free transmission. The price to be paid for achieving good responsiveness is large input signal in the initial phase required to quickly bring the system to the state of maximum efficiency (and maximum throughput). In Sect. 5.1.3.1, we showed that by using a time-varying plane with appropriately adjusted velocity and duration of the movement phase (determined by constant k_{VP}), one can achieve fast reaction to the bandwidth fluctuations and at the same time limit excessive transmission rates. The extreme value of the control signal generated by strategy (5.93) was adjusted indirectly by the choice of k_{VP}. In this section, we will use an alternative approach to SM controller design, which results in a similar set of properties as the algorithm employing time-varying plane (5.93) but gives direct control over the extreme value of the generated transmission rate. The approach presented here employs the reaching law proposed in [15].

We analyze the situation when the control signal is subject to the constraint

$$0 \le u\,(kT) \le u_{\max}, \tag{5.107}$$

where $u_{\max} > d_{\max}$. The reaching law proposed by Golo and Milosavljević [15] can be synthesized in the following way:

$$s\,[(k+1)\,T] - s\,(kT) = -\Phi\,[s\,(kT)]\,, \tag{5.108}$$

where

$$\Phi\,[s\,(kT)] = \min\,(|s\,(kT)|\,,\delta)\ \mathrm{sgn}\,[s\,(kT)] \tag{5.109}$$

and $\delta > 0$. Function sgn$(\cdot)$ is defined in the same way as in continuous-time domain, i.e., for argument x, $\mathrm{sgn}(x) = -1$ if $x \le 0$, and $\mathrm{sgn}(x) = 1$ for $x > 0$. With this law applied, the system representative point is guaranteed to reach the hyperplane $\mathbf{c}^{\mathrm{T}}\mathbf{e}(kT) = 0$ monotonically in a finite number of steps in a way determined by the choice of coefficient δ. For the purpose of further analysis, we introduce an auxiliary variable, $s_{\mathrm{A}}(kT)$,

$$s_{\mathrm{A}}\,(kT) = s\,(kT) + f_{\mathrm{RL}}\,(kT) = \mathbf{c}^{\mathrm{T}}\mathbf{e}\,(kT) + f_{\mathrm{RL}}\,(kT) = 0, \tag{5.110}$$

where the strictly monotonic function $f_{\mathrm{RL}}(\cdot)$, reflecting the distance from the plane $\mathbf{c}^{\mathrm{T}}\mathbf{e}(kT) = 0$, is defined as

$$\begin{cases} f_{\mathrm{RL}}\,[(k+1)\,T] = f_{\mathrm{RL}}\,(kT) + \delta\,\mathrm{sgn}\,[s\,(kT)] & \text{for} \quad k < k_{\mathrm{RL}}, k_{\mathrm{RL}} \in C_+, \\ f_{\mathrm{RL}}\,[(k+1)\,T] = 0 & \text{for} \quad k \ge k_{\mathrm{RL}}. \end{cases} \tag{5.111}$$

The positive integer k_{RL} in (5.111) corresponds to the duration of the reaching phase. We assume that $f_{\mathrm{RL}}(0) = -\mathbf{c}^{\mathrm{T}}\mathbf{e}(0) = -c_n y_{\mathrm{D}}$. Since $f_{\mathrm{RL}}(\cdot)$ is strictly monotonic, this assumption also implies that for any $k \in [0; k_{\mathrm{RL}})$, $f_{\mathrm{RL}}(\cdot)$ and c_n have opposite signs. With such representation of the reaching law, we may conveniently split the design procedure into two phases. First, the sliding plane parameters are selected for a dead-beat controller, and afterwards, the reaching law is chosen to satisfy the explicit input constraint (5.107).

Step 1. In the first part of the design, we choose vector $\mathbf{c}$ defining the orientation of the sliding plane $s(kT) = \mathbf{c}^{\mathrm{T}}\mathbf{e}(kT) = 0$. We opt for the highest responsiveness provided by dead-beat scheme. Thus, we want the closed-loop characteristic polynomial $\det(z\mathbf{I}_n - \mathbf{A}_{\mathbf{cl}})$ to be equal to z^n. Due to the similarity between the representation of the reaching law given by (5.110) and the mathematical description of time-varying plane (5.78), this design step can be conducted in a similar way, as it was done in Sect. 5.1.3.1, Step 2. Consequently, on the basis of (5.85)–(5.93), we get vector $\mathbf{c}$

$$\mathbf{c}^{\mathrm{T}} = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \end{bmatrix} c_n,$$

and the control law

$$u(kT) = y_{\mathrm{D}} - y(kT) - \sum_{j=k-n_{\mathrm{p}}}^{k-1} u(jT) + f_{\mathrm{RL}}[(k+1)T]/c_n. \tag{5.112}$$

Step 2. We need to select the parameter of the reaching law, $\delta > 0$, such that the resultant control signal will never exceed $u_{\max}$.

First, notice that $u(0) = y_{\mathrm{D}} + f_{\mathrm{RL}}(T)/c_n$. For $k \geq 1$, the control signal satisfies the relation defined in the following lemma.

Lemma 5.8. *If controller (5.112) with function $f_{\mathrm{RL}}(\cdot)$ defined by (5.111) is applied to system (5.7)–(5.10), then for any $k \geq 1$,*

$$u(kT) = h[(k-1)T] + \{f_{\mathrm{RL}}[(k+1)T] - f_{\mathrm{RL}}(kT)\}/c_n. \tag{5.113}$$

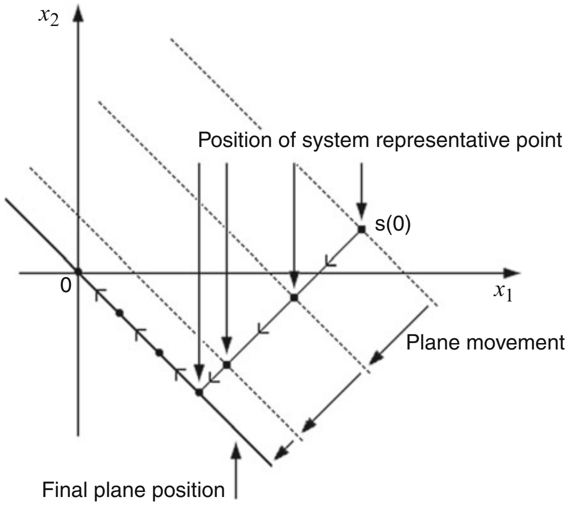
Proof. Since $f_{\mathrm{RL}}(\cdot)$ is strictly monotonic and satisfies the same initial conditions as it was assumed in (5.79), the lemma is true as a direct consequence of the reasoning presented in (5.96)–(5.98). This concludes the proof of the lemma. □

Using the definition of function $f_{\mathrm{RL}}(\cdot)$ (5.111), we can represent control law (5.113) as

$$\begin{cases} u(kT) = h[(k-1)T] + \delta \operatorname{sgn}[s(kT)]/c_n & \text{for} \quad k < k_{\mathrm{RL}}, \\ u(kT) = h[(k-1)T] & \text{for} \quad k \geq k_{\mathrm{RL}}. \end{cases} \tag{5.114}$$

It follows from (5.2) that $\forall k$, $0 \leq h(kT) \leq d_{\max}$. Therefore, the control signal given by (5.114) is nonnegative and bounded by $d_{\max} < u_{\max}$ for any $k \geq k_{\mathrm{RL}}$. Due to the delay in the feedback loop, the first packets may arrive at the node no sooner

Fig. 5.30 Comparison between representative point displacement under reaching law and with time-varying plane



than at $kT = n_pT$. Since the buffer is assumed to be empty before the control process commences at $kT = 0$, then $h(k < n_p) = 0$. As a consequence, in order to ensure that control signal (5.114) conforms to inequalities (5.107) for all $k < k_{\mathrm{RL}}$, parameter δ should satisfy the following constraint:

$$\begin{cases} \delta \leq |c_n|\, u_{\max} & \text{for} \quad 0 \leq k \leq n_p, \\ \delta \leq |c_n|\, (u_{\max} - d_{\max}) & \text{for} \quad k > n_p. \end{cases} \tag{5.115}$$

This ends selection of the reaching law and concludes the design procedure. The obtained controller calculates the transmission rate from (5.112) with function $f_{\mathrm{RL}}(\cdot)$ defined by (5.111) subject to constraint (5.115).

The analysis of the mathematical formulation of the designed reaching-law-based algorithm (5.110)–(5.113) in comparison with the findings related to the controller using a time-varying plane (5.93) indicates functional similarities of both strategies. In fact, in the considered application, if we describe the movement of the time-varying plane by (5.111) with the initial condition $f_{\mathrm{RL}}(0) = -\mathbf{c}^{\mathrm{T}}\mathbf{e}(0)$, the two approaches become equivalent. This is apparent if we consider the trajectory of the system representative point sketched for a hypothetical second-order discrete-time system in Fig. 5.30. The reaching law changes the position of the representative point so that it approaches a fixed sliding plane (line in the example depicted in Fig. 5.30). We can imagine that the plane is not fixed but moves from $s(0)$ to the final position passing through the origin of the error state space. Then, if the control law is formulated so that it maintains the system representative point on the moving plane, the point trajectory matches the one imposed by reaching law. Consequently, in both cases, we obtain identical system dynamics. When comparing the two approaches to SM controller design, it is also valuable to confront the most

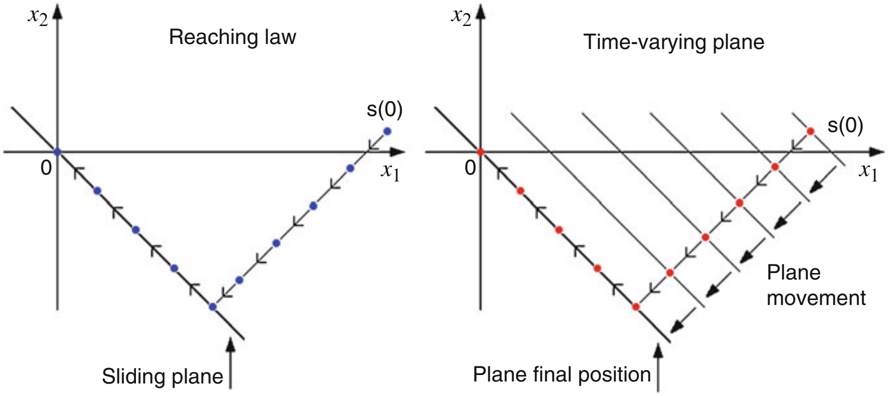


Fig. 5.31 Representative point trajectory – reaching law vs. time-varying plane

efficient function governing the movement of time-varying plane (5.82) with the one implied by reaching law (5.111). It is done in Example 5.3.

Example 5.3. Let us analyze the rate assignments in the interval $[0, k_0T]$, where k_0 is either k_{VP} or k_{RL}, for a data source constrained by the maximum rate $u_{\max} = 8$ packets per period. Let us assume the system initial state – $\mathbf{c}^{\mathrm{T}}\mathbf{e}(0) = -35$ packets. This implies that the reaching phase should last for at least $k_0 = 5$ periods. The function describing the displacement of time-varying sliding plane (5.82) partitions the distance from the plane final position (passing through the origin of the state space) into segments of equal size, yielding the constant transmission rate of seven packets per period. In contrast, function (5.111) defining the reaching law proposed by Golo and Milosavljević [15] allows for the maximum rate in the first $k_0 - 1$ steps and covers the remaining distance from the final position of the plane in the last period. The trajectory of the system representative point resulting from the execution of the strategies being compared is illustrated in Fig. 5.31. The rate assignments are shown in Fig. 5.32. Let us imagine a communication scenario in which a web page consisting of 24 packets is to be retrieved from a server operating at the source. The reaching-law-based controller allows us to transmit the page in three periods, whereas the controller with time-varying plane described by (5.82) requires at least four periods. This shows that in certain cases, the reaching-law-based controller (5.112) may exploit the network capabilities in a more efficient way than the one using time-varying plane (5.93).

Properties of the Proposed Controller

The properties of the proposed nonlinear controller developed using reaching-law approach will be formulated as two theorems. The first one shows that the controller does not cause buffer overflow. The second theorem, in turn, defines the minimum value of the demand queue length (which also constitutes the lower bound of the

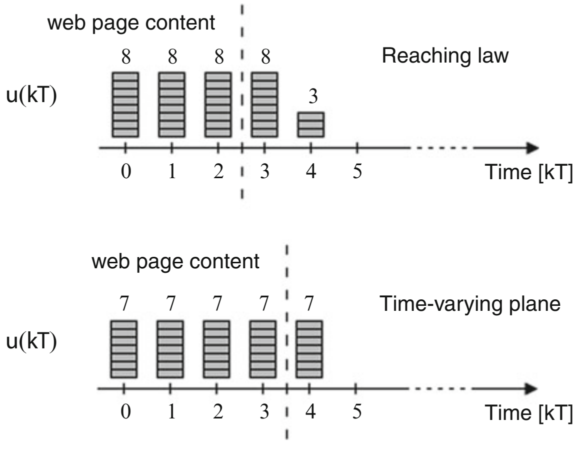


Fig. 5.32 Rate assignment in Example 5.3

buffer capacity) which will guarantee that the available bandwidth is entirely used for data transfer.

Theorem 5.9. *If controller (5.112) with function $f_{\mathrm{RL}}(\cdot)$ defined by (5.111) is applied to system (5.7)–(5.10), then the queue length in the bottleneck node buffer is always upper-bounded by its demand value y_{D}.*

Proof. The buffer at the bottleneck node is empty for any $kT \leq \mathrm{RTT} = n_p T$. Hence, it suffices to show that the proposition is satisfied for any $k > n_p$. Using Lemma 5.8, the queue length given by (5.6) can be presented as (see also (5.106))

$$\begin{aligned} y\left(kT\right) = u(0) + \sum_{j=1}^{k-n_p-1} h\left[(j-1)\,T\right] - \sum_{j=0}^{k-1} h\left(jT\right) \\ + \sum_{j=1}^{k-n_p-1} \left\{f_{\mathrm{RL}}\left[(j+1)\,T\right] - f_{\mathrm{RL}}\left(jT\right)\right\}/c_n \\ = y_{\mathrm{D}} - \sum_{j=k-n_p-1}^{k-1} h\left(jT\right) + f_{\mathrm{RL}}\left[\left(k-n_p\right)T\right]/c_n\,. \end{aligned} \tag{5.116}$$

Since the utilized bandwidth $h(\cdot)$ is always nonnegative, and for any k, $f(kT)$ and c_n have opposite signs, $y(kT)$ given by (5.116) never exceeds the demand value y_{D}. This ends the proof. □

Theorem 5.10. *If controller (5.112) with function $f_{\mathrm{RL}}(\cdot)$ defined by (5.111) is applied to system (5.7)–(5.10), and the demand queue length satisfies the inequality $y_{\mathrm{D}} > d_{\max}(n_p + 1)$, then for any $k \geq k_{\mathrm{RL}} + n_p + 1$, the queue length is strictly positive.*

Proof. It follows from (5.111) that for $k > k_{\mathrm{RL}}$ function $f_{\mathrm{RL}}(kT) = 0$. Consequently, for $k \geq k_{\mathrm{RL}}$, controller (5.112) becomes equivalent to control law (5.93), whose action influences the packet queue length for $k \geq k_{\mathrm{RL}} + n_p + 1$. Since both controllers incorporate the rate history in exactly the same way, then taking into account relation (5.105), the proposition is valid as a direct consequence of Theorem 5.6. This completes the proof. □

The comparison of Theorems 5.9 and 5.10, defining the crucial properties of the SM controller employing reaching law, with Theorems 5.5 and 5.6 formulated for the one with time-varying plane, confirms the functional equivalence of both solutions in the considered system. The principal difference between the two strategies, apart from distinct conceptual foundations, relates to the method of fulfilling constraint (5.107). The strategy developed using the idea of time-varying plane attempts to satisfy this constraint by adjusting the duration of the plane movement (reminiscent of the reaching phase). As a result, it gives indirect control over the actual extreme rate value. Controller (5.112), on the other hand, allows for explicit definition of the maximum permitted rate and provides implicit control over the duration of the reaching phase.

The properties of the reaching-law-based strategy have been verified in a simulation scenario discussed in the following section.

Simulation Results

Performance of nonlinear controller (5.112) is verified for the network model described in Sect. 5.1.2. We leave the model parameters unchanged: discretization period $T = 10$ ms, round-trip time $\mathrm{RTT} = n_p T = 10T$, and the maximum available bandwidth $d_{\max} = 10$ packets per period. Similarly as before, we run two series of simulation tests: one for the slowly varying bandwidth shown in Fig. 5.7 and another for the stochastic pattern D_{norm}(mean, standard deviation) $= D_{\mathrm{norm}}$(5 packets, 5 packets) illustrated in Fig. 5.12. In both tests, we assume the maximum allowed rate $u_{\max} = 15$ packets $> d_{\max}$. Parameter δ governing the reaching law is set according to (5.115) as $|c_n| u_{\max} = 15$ packets for $0 \leq k \leq 10$, and $|c_n|(u_{\max}–d_{\max}) = 15–10 = 5$ packets for $k > 10$.

Test 1. In order to ensure that all of the available bandwidth is used for data transfer, we set y_{D} in the first series of simulations according to the guidelines provided by Theorem 5.10 as $112 > 110$ packets. The results of the simulation for the available bandwidth depicted in Fig. 5.7 are given in Figs. 5.33–5.35: the transmission rate established by the controller in Fig. 5.33, the buffer occupancy in Fig. 5.34, and the sliding variable in Fig. 5.35. We can see from the plot in Fig. 5.33 that the application of a properly tuned reaching law guarantees that the input rate constraint is satisfied. The controller assigns the maximum rate of 15 packets in the first 7 periods, and 7 more packets in the last step (step 7) before going to zero at $kT = 8T$ when the reaching phase terminates. This is confirmed by the analysis of the sliding variable presented in Fig. 5.35. It follows from this figure that $s(kT)$,

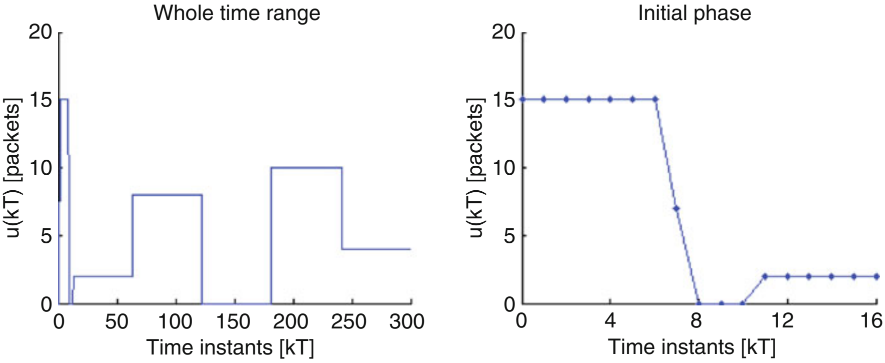
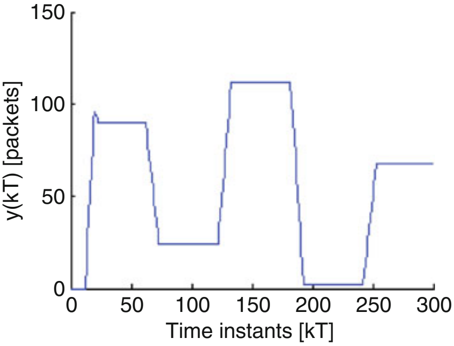


Fig. 5.33 Transmission rate

Fig. 5.34 Queue length



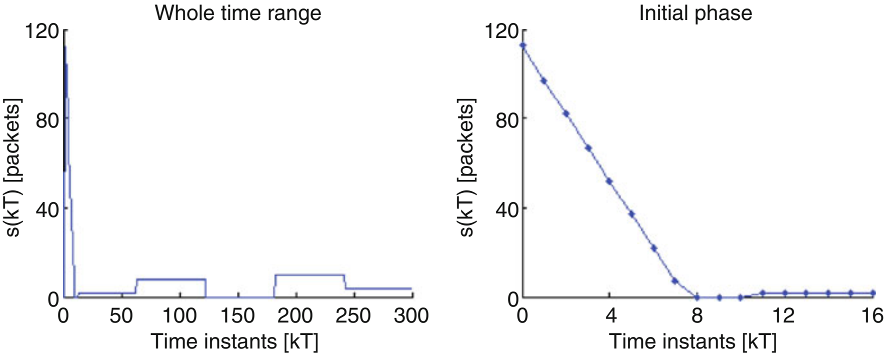


Fig. 5.35 Sliding variable

which constitutes a measure of the distance from the sliding plane, decreases from the initial value $s(0) = \mathbf{c}^{\mathrm{T}}\mathbf{e}(0) = 112$ packets to zero in 8 steps. In the first 7 periods, the distance diminishes at the rate of 15 packets per period, and in the last step before

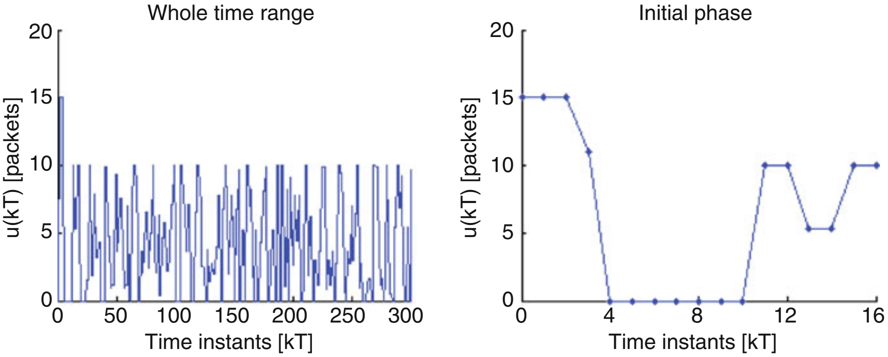


Fig. 5.36 Transmission rate

reaching the plane $s(kT) = 0$, the system representative point covers the remaining difference of 7 packets. Consequently, the reaching phase terminates at $kT = 8T$ and the corresponding $k_{\mathrm{RL}} = 8$. Afterwards, the system is maintained in a stable quasi-sliding motion. Indeed, despite the mismatched external disturbance $d(\cdot)$, the system representative point never leaves the one-sided band of 10 packets from the plane. Moreover, the buffer occupancy presented in Fig. 5.34 demonstrates that the queue length never exceeds its demand value $y_{\mathrm{D}} = 112$ packets, which means that packet losses are indeed eliminated. Furthermore, following the initial phase, the queue length does not fall to zero, implying that all of the available bandwidth is used for data transfer. These two observations indicate that the maximum throughput in the analyzed communication system is achieved.

Test 2. In the second run of simulations, we verify the controller performance in the presence of the stochastic bandwidth fluctuations $D_{\mathrm{norm}}(5, 5)$ shown Fig. 5.12. Since the mean bandwidth is much lower than the maximum one $d_{\max}$, in the tests, we use $d_{\mathrm{L}} = 5$ packets instead of $d_{\max} = 10$ packets in the formula specified in Theorem 5.10 and adjust y_{D} as $56 > 55$ packets. As a consequence, we may expect lower throughput than in Test 1 but at a propitious trade-off in buffer size. The test results are illustrated in Figs. 5.36–5.38: the rate allocation in Fig. 5.36, the packet queue length in Fig. 5.37, and the sliding variable in Fig. 5.38.

The controller allocates the maximum rate of 15 packets in the first three periods (0–2) and 11 packets in period 3. Period 3 is the last one before the system representative point meets the sliding plane (at instant $kT = 4T$) when the reaching phase terminates. This is in agreement with the evolution of the sliding variable depicted in Fig. 5.38. The value of $s(kT)$ decreases from $s(0) = \mathbf{c}^{\mathrm{T}}\mathbf{e}(0) = 56$ packets to zero in four steps. In the first three periods, the distance drops by 15 packets per period, and in the last step before reaching the plane $s(kT) = 0$, the system representative point advances by 11 packets. Therefore, the reaching phase indeed ends at $kT = 4T$ ($k_{\mathrm{RL}} = 4$). In the subsequent part of the transmission, the system representative point never leaves the one-sided band of 10 packets from the plane,

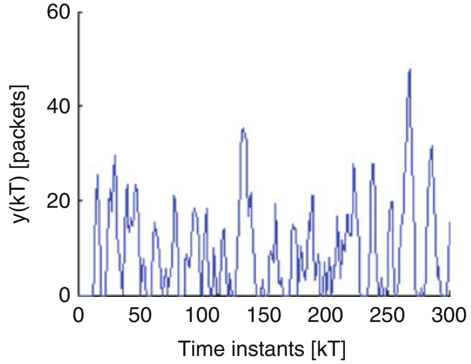


Fig. 5.37 Queue length

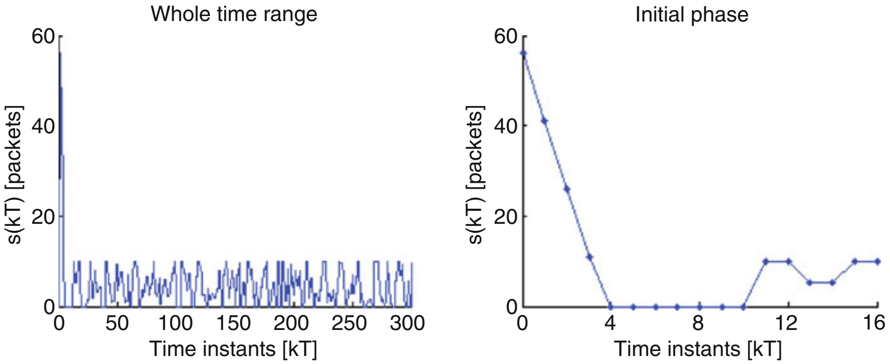


Fig. 5.38 Sliding variable

which demonstrates a properly established quasi-sliding motion in discrete-time domain. If we analyze the queue length evolution presented in Fig. 5.37, we will notice that $y(kT)$ decreases to zero in several periods. This means that certain part of the available bandwidth may be left unused. The numerical computations indicate that bandwidth utilization degrades to 86% as compared with the results of Test 1.

The plot of function $f_{\mathrm{RL}}(kT)$ resulting from the application of reaching law (5.108) for the analyzed case of $c_n = 1$ is depicted in Fig. 5.39. Curve (a) in this figure was obtained in Test 1, whereas curve (b) represents $f_{\mathrm{RL}}(kT)$ from Test 2.

5.1.3.3 Application of a Saturation Element

In the previous sections, we considered two methods for constraining excessive input rate. The presented techniques, strongly related to the theory of sliding-mode control, shape the trajectory of the system representative point so that the control effort exerted in a single period can be reduced. The first technique

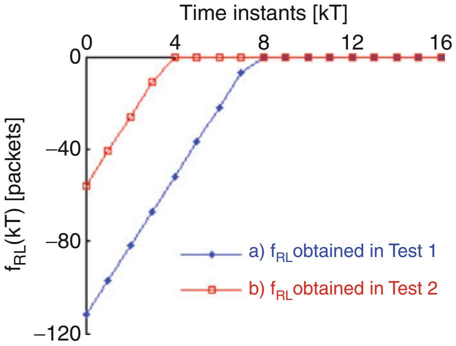


Fig. 5.39 Function $f_{\mathrm{RL}}(\cdot)$

employs a time-varying hyperplane instead of the typically considered fixed one and appropriately adjusts the hyperplane dynamics in order to meet (indirectly) the imposed input constraint. The second method uses a fixed hyperplane but enforces behavior of the system in the reaching phase such that the representative point approaches the plane in a prescribed manner according to the limitations of the input signal. The first technique gives implicit control over the maximum transmission rate value and explicit one over the duration of the reaching phase. The second method, in turn, allows for direct control over the maximum permitted transmission rate in the system and adapts the duration of the reaching phase indirectly. In this section, we present another technique used to ensure that the control signal never exceeds the specified upper bound. This method involves the use of a saturation element, which clamps down any transmission rate generated by the controller exceeding the specified upper limit. In this way, similarly as in the reaching-law-based approach presented in the previous section, explicit control over the maximum value of the control signal is provided. This direct approach would probably be favored by telecommunications and software practitioners due to the smallest intricacy at the implementation level. Moreover, it might prove more tractable for control engineers not necessarily familiar with sophisticated, though interesting, theory of discrete sliding-mode control.

Proposed Control Strategy

Similarly as in the previous section, we assume that the control signal is limited from above by a finite value $u_{\max} > 0$, precisely as it was specified by inequalities (5.109). The amount of data to be sent by the source at time kT is determined by the controller according to the following formula:

$$u(kT) = \min\{\omega(kT), u_{\max}\}, \tag{5.117}$$

where

$$\omega(kT) = y_D - y(kT) - \sum_{j=k-n_p}^{k-1} u(jT). \tag{5.118}$$

In order to adequately respond to the changing networking conditions and to ensure the maximum throughput when the available bandwidth $d(kT) = d_{\max}$, it is assumed that $u_{\max} > d_{\max}$.

Properties of the Proposed Strategy

The feasibility conditions for any flow control algorithm developed for a data transmission network require the source rate of sending packets into the network to be finite and nonnegative. The first condition of a bounded rate for the analyzed control algorithm is satisfied by definition (5.117). We will prove that the transmission rate calculated according to (5.117) and (5.118) is always nonnegative so that the second requirement is also fulfilled.

It follows directly from definition (5.117) that at any time instant $kT \geq 0$ the generated transmission rate satisfies the following inequality:

$$u(kT) \leq \omega(kT). \tag{5.119}$$

We show in the lemma formulated below that the rate $u(kT)$ is indeed nonnegative for any $kT \geq 0$.

Lemma 5.11. *If controller (5.117) with function $\omega(\cdot)$ defined by (5.118) is applied to system (5.7)–(5.10), then for any $k \geq 0$, the generated transmission rate is nonnegative, i.e.,*

$$\underset{k\geq 0}{\forall}\ u(kT) \geq 0. \tag{5.120}$$

Proof. At the initial time, function $\omega(0) = y_D$. Therefore, the flow rate $u(0) = \min\{y_D,\ u_{\max}\}$ equals either y_D or $u_{\max}$. Consequently, since both y_D and $u_{\max}$ are assumed to be positive, inequality (5.120) is satisfied for $k = 0$. On the other hand, at any time instant $kT > 0$, if the amount of data to be sent by the source is $u_{\max}$, then the flow rate u(kT) is also strictly positive. Hence, in order to complete the proof, it is only necessary to show that (5.120) is satisfied for any $k > 0$ when $u(kT) = \omega(kT)$. On the basis of (5.3), we can write the following relation for the queue length at instant kT:

$$y(kT) = y[(k-1)T] + u\big[\big(k - n_p - 1\big)T\big] - h[(k-1)T]. \tag{5.121}$$

Since in the analyzed case $u(kT) = \omega(kT)$ and the bottleneck queue length satisfies (5.121), we obtain

$$
\begin{aligned}
u(kT) = \omega(kT) &= y_{\mathrm{D}} - y(kT) - \sum_{j=k-n_p}^{k-1} u(jT) \\
&= y_{\mathrm{D}} - y[(k-1)T] - u\left[\left(k-n_p-1\right)T\right] + h[(k-1)T] - \sum_{j=k-n_p}^{k-1} u(jT) \\
&= y_{\mathrm{D}} - y[(k-1)T] - \sum_{j=k-n_p-1}^{k-1} u(jT) + h[(k-1)T] \\
&= y_{\mathrm{D}} - y[(k-1)T] - \sum_{j=k-n_p-1}^{k-2} u(jT) - u[(k-1)T] + h[(k-1)T] \\
&= \omega[(k-1)T] - u[(k-1)T] + h[(k-1)T]. \qquad (5.122)
\end{aligned}
$$

Taking into account inequality (5.119) and the fact that the utilized bandwidth is always nonnegative, we obtain

$$u(kT) \geq h[(k-1)T] \geq 0, \qquad (5.123)$$

which shows that inequality (5.120) indeed holds at any time instant $kT > 0$ when $u(kT) = \omega(kT)$. This conclusion ends the proof of the lemma. □

It follows from the analysis presented above that the input constraint is always satisfied; i.e., the presented algorithm guarantees that for any $k \geq 0$, $0 \leq u(kT) \leq u_{\max}$.

In the further part of this section, we present three theorems stating important properties of the proposed flow control scheme. The first one gives the condition which must be satisfied in order to eliminate the risk of data loss as a consequence of exceeding the bottleneck node buffer capacity. Afterwards, the second theorem provides a sufficient condition for full bottleneck link bandwidth utilization. Finally, a relation between the control signal $u(kT)$ and the consumed bandwidth is formulated in the third theorem.

Theorem 5.12. *If controller (5.117) with function $\omega(\cdot)$ defined by (5.118) is applied to system (5.7)–(5.10), then the queue length in the bottleneck node buffer is always upper-bounded by its demand value.*

Proof. As it has already been proved, data transmission rate $u(\cdot)$ is nonnegative at any time instant kT. On the other hand, by the definition, $u(kT)$ is smaller than or equal to $\omega(kT)$. Therefore, the following relation holds for any time $kT \geq 0$:

$$y_{\mathrm{D}} - y\,(kT) - \sum_{j=k-n_p}^{k-1} u\,(jT) = \omega\,(kT) \geq u\,(kT) \geq 0. \tag{5.124}$$

Hence, the queue length satisfies

$$y\,(kT) \leq y_{\mathrm{D}} - \sum_{j=k-n_p}^{k-1} u\,(jT). \tag{5.125}$$

Again taking into account that $u(kT)$ is always nonnegative, one concludes that the queue length indeed never exceeds its demand value. This ends the proof of the theorem. □

Another desirable property of the analyzed system is full bottleneck link bandwidth utilization. Since the bottleneck link bandwidth $d(kT)$ is fully used if the queue length $y[(k+1)T]$ is strictly greater than zero, then the next theorem specifies a condition which guarantees that the queue length in our scheme is always strictly positive.

Theorem 5.13. *If controller (5.117) with function $\omega(\cdot)$ defined by (5.118) is applied to system (5.7)–(5.10), the maximum rate $u_{\max} > d_{\max}$, and the demand value of the queue length y_{D} satisfies the following inequality:*

$$y_{\mathrm{D}} > u_{\max}\left(n_p + 1\right), \tag{5.126}$$

then for any $k \geq n_p + 1$, the queue length in the bottleneck node buffer is strictly positive.

Proof. Let us introduce an auxiliary function $\varphi(\cdot)$ defined for any instant kT as

$$\varphi\,(kT) = y\,(kT) + \sum_{j=k-n_p}^{k-1} u\,(jT). \tag{5.127}$$

This function, reminiscent of (4.11) in continuous-time domain, represents the sum of three quantities:

1. The number of packets currently waiting in the bottleneck buffer $y(kT)$.
2. The number of "in-flight" packets, i.e., those which have already been sent by the source but have not yet arrived at the bottleneck node.
3. The number of packets still required to be sent by the source (this is because the controller has already sent a command signal to the source).

The sum $\sum_{j=k-n_p}^{k-1} u\,(jT)$ in (5.127) accounts for the overall number of packets defined in 2 and 3. Substituting (5.6) into (5.127), one can express function $\varphi(kT)$ as

$$\varphi\left(kT\right)=\sum_{j=0}^{k-n_p-1}u\left(jT\right)-\sum_{j=0}^{k-1}h\left(jT\right)+\sum_{j=k-n_p}^{k-1}u\left(jT\right)=\sum_{j=0}^{k-1}u\left(jT\right)-\sum_{j=0}^{k-1}h\left(jT\right). \tag{5.128}$$

Hence, taking into account assumption (5.126), we have

$$\varphi(0)=0<y_{\mathrm{D}}-u_{\max}\left(n_p+1\right)<y_{\mathrm{D}}-u_{\max}. \tag{5.129}$$

Furthermore, if for some k the following inequality $\varphi(kT)<y_{\mathrm{D}}-u_{\max}$ is satisfied, then

$$\omega\left(kT\right)=y_{\mathrm{D}}-y\left(kT\right)-\sum_{j=k-n_{\mathrm{p}}}^{k-1}u\left(jT\right)=y_{\mathrm{D}}-\varphi\left(kT\right)>u_{\max}, \tag{5.130}$$

which implies according to (5.117) that the transmission rate actually assigned to the source $u(kT)=u_{\max}$. Consequently, since $u_{\max}>d_{\max}$, we conclude that if $\varphi(kT)<y_{\mathrm{D}}-u_{\max}$, then function $\varphi(\cdot)$ increases at least at the rate $u_{\max}-d_{\max}$. Moreover, since for any time $kT<n_pT$ the consumed bandwidth $h(kT)=0$, then if $\varphi(kT)<y_{\mathrm{D}}-u_{\max}$ and condition (5.126) is satisfied, $\varphi(\cdot)$ increases exactly at the rate $u_{\max}$ at each time instant $kT=T,\ \ldots,\ (n_p-1)T$ reaching $n_pu_{\max}$ at the time n_pT. On the other hand, the consumed bandwidth for any time $kT\geq n_pT$ satisfies inequality $h(kT)\leq d_{\max}$. This implies that $\varphi(\cdot)$, which is the sum of packets waiting in the buffer and those scheduled for arrival at the node in the next n_p periods, may decrease at most at the rate $d_{\max}$.

Further, we will show that function $\varphi(\cdot)$ after reaching $n_pu_{\max}$ never decreases below this value; i.e., we will demonstrate that the following inequality holds for any $kT\geq n_pT$:

$$\varphi\left(kT\right)\geq n_{\mathrm{p}}u_{\max}. \tag{5.131}$$

In order to prove this statement, we will apply the principle of the mathematical induction. Let us first check whether (5.131) holds for $k=n_p+1$. If condition (5.126) is satisfied, then $\varphi(n_pT)=n_pu_{\max}<y_{\mathrm{D}}-u_{\max}$. This implies, according to (5.130) and (5.117), that $u(n_pT)=u_{\max}$. Consequently, we have

$$\begin{aligned}\varphi\left[\left(n_p+1\right)T\right]&=\sum_{j=0}^{n_p}u\left(jT\right)-\sum_{j=0}^{n_p}h\left(jT\right)=\sum_{j=0}^{n_p-1}u\left(jT\right)-\sum_{j=0}^{n_p-1}h\left(jT\right)\\&\quad+u\left(n_pT\right)-h\left(n_pT\right)=\varphi\left(n_pT\right)+u\left(n_pT\right)-h\left(n_pT\right)\\&=n_pu_{\max}+u_{\max}-h\left(n_pT\right)\geq n_pu_{\max}+u_{\max}-d_{\max}>n_pu_{\max},\end{aligned} \tag{5.132}$$

which shows that (5.131) is indeed true for $k = n_p + 1$. Now, let us assume that (5.131) holds for some integer $l > n_p + 1$. We will show that this implies that (5.131) is also satisfied for $l + 1$. For this purpose, we will consider the following two cases: the first one when $u(lT) = \omega(lT)$ and the second one when $u(lT) = u_{\max}$.

Case 1. When it occurs that $u(lT) = \omega(lT)$, then taking into account the relation $\omega(lT) = y_{\mathrm{D}} - \varphi(lT)$ and inequality $u_{\max} > d_{\max}$, we obtain

$$\begin{aligned}\varphi\left[(l+1)T\right] &= \varphi(lT) + u(lT) - h(lT) = \varphi(lT) + \omega(lT) - h(lT) \\ &= \varphi(lT) + y_{\mathrm{D}} - \varphi(lT) - h(lT) = y_{\mathrm{D}} - h(lT) \\ &\geq y_{\mathrm{D}} - d_{\max} > y_{\mathrm{D}} - u_{\max} > n_p u_{\max}. \end{aligned} \tag{5.133}$$

Case 2. In the second situation, i.e., when $u(lT) = u_{\max}$, we can write

$$\begin{aligned}\varphi\left[(l+1)T\right] &= \varphi(lT) + u(lT) - h(lT) = \varphi(lT) + u_{\max} - h(lT) \\ &\geq \varphi(lT) + u_{\max} - d_{\max} > \varphi(lT) > n_p u_{\max}. \end{aligned} \tag{5.134}$$

Therefore, applying the principle of the mathematical induction, we conclude that relation (5.131) actually holds for any time $kT > n_pT$.

Finally, taking into account relations (5.128) and (5.131) and the fact that the flow rate generated by the analyzed controller is always upper-bounded by $u_{\max}$, for any time $kT > n_pT$, we get

$$y(kT) = \varphi(kT) - \sum_{j=k-n_p}^{k-1} u(jT) > n_p u_{\max} - n_p u_{\max} = 0. \tag{5.135}$$

This completes the proof. □

Theorem 5.13 shows that by using strategy (5.117) with condition (5.126), one can ensure full bottleneck link bandwidth utilization for any time $kT > n_pT$. Further, in the next theorem, a relation between the flow rate and the consumed bandwidth is stated and proved.

Theorem 5.14. *If controller (5.117) with function $\omega(\cdot)$ defined by (5.118) is applied to system (5.7)–(5.10), the demand queue length $y_{\mathrm{D}} > u_{\max}$ and the maximum flow rate $u_{\max} > d_{\max}$, then there exists a nonnegative integer k_ω satisfying*

$$k_\omega < \frac{y_{\mathrm{D}} - u_{\max}}{u_{\max} - d_{\max}} + 1, \tag{5.136}$$

such that for any $k > k_\omega$, the following relation holds:

$$u(kT) = h\left[(k-1)T\right]. \tag{5.137}$$

Furthermore, when $y_{\mathrm{D}} \leq u_{\max}$, relation (5.137) is satisfied for any $k \geq 1$.

Proof. First, we will consider the situation when the demand queue length is smaller than or equal to the maximum allowed rate, i.e., the situation when the inequality $y_D \leq u_{\max}$ holds. It will be shown that in such circumstances the following relation is always satisfied:

$$\underset{k\geq 0}{\forall}\ \omega\left(kT\right) \leq u_{\max}, \tag{5.138}$$

which from the algorithm definition (5.117) directly implies $u(kT) = \omega(kT)$.

In order to prove that relation (5.138) is indeed satisfied for any time $kT \geq 0$, we apply the principle of the mathematical induction. At the initial time, $\omega(0) = y_D \leq u_{\max}$. Therefore, inequality (5.138) holds for $k = 0$. Now, let us assume that (5.138) is true for some integer $l > 0$, and we will show that it is also satisfied for $l + 1$. Using equations (5.118) and (5.121), and taking into account the fact that $u(kT) = \omega(kT)$, we get

$$\begin{aligned}\omega\left[\left(l+1\right)T\right] &= y_D - y\left[\left(l+1\right)T\right] - \sum_{j=l-n_p+1}^{l} u\left(jT\right)\\ &= y_D - y\left(lT\right) - u\left[\left(l-n_p\right)T\right] + h\left(lT\right) - \sum_{j=l-n_p+1}^{l} u\left(jT\right)\\ &= y_D - y\left(lT\right) - \sum_{j=l-n_p}^{l-1} u\left(jT\right) - u\left(lT\right) + h\left(lT\right)\\ &= \omega\left(lT\right) - u\left(lT\right) + h\left(lT\right).\end{aligned} \tag{5.139}$$

The utilized bandwidth $h(\cdot)$ never exceeds $d_{\max}$. Since we assumed that $u(lT) = \omega(lT)$, we have

$$\omega\left[\left(l+1\right)T\right] = \omega\left(lT\right) - u\left(lT\right) + h\left(lT\right) = h\left(lT\right) \leq d_{\max} < u_{\max}, \tag{5.140}$$

which ends the induction proof of inequality (5.138).

Since it follows from (5.138) that at any time instant $kT \geq 0$ the flow rate $u(kT) = \omega(kT)$, then using expression (5.122), for any $k \geq 1$, we obtain

$$u\left(kT\right) = \omega\left[\left(k-1\right)T\right] - u\left[\left(k-1\right)T\right] + h\left[\left(k-1\right)T\right] = h\left[\left(k-1\right)T\right]. \tag{5.141}$$

Equation 5.141 shows that if $y_D \leq u_{\max}$, then (5.137) indeed holds for any positive integer k.

Now, let us consider the complementary case, i.e., the situation when $y_D > u_{\max}$. In order to demonstrate relations (5.136) and (5.137), we will use the auxiliary function $\varphi(\cdot)$ defined by (5.127). On the basis of the argument presented in the proof of Theorem 5.13, we know that if for some k the inequality $\varphi(kT) < y_D - u_{\max}$ is satisfied, then it follows from equation (5.128) and the assumption $u_{\max} > d_{\max}$ that function $\varphi(kT)$ increases at least at the rate $u_{\max} - d_{\max}$. Thus, there exists such a finite time instant $k_\omega T$ when the following condition

$$\varphi\,(kT) \geq y_{\mathrm{D}} - u_{\max} \tag{5.142}$$

is satisfied for the first time.

We will determine the latest instant when inequality (5.142) can be satisfied for the first time. Since function $\varphi(kT)$ is smaller than the difference $y_{\mathrm{D}} - u_{\max}$ until $k < k_\omega$, then

$$\varphi\,[(k_\omega - 1)\,T] = \sum_{j=0}^{k_\omega-2} u\,(jT) - \sum_{j=0}^{k_\omega-2} h\,(jT) < y_{\mathrm{D}} - u_{\max}. \tag{5.143}$$

Moreover, since the flow rate for any $k < k_\omega$ is equal to $u_{\max}$, then inequality (5.143) can be rewritten as

$$(k_\omega - 1)\,u_{\max} - \sum_{j=0}^{k_\omega-2} h\,(jT) < y_{\mathrm{D}} - u_{\max}. \tag{5.144}$$

Number k_ω in this equation is the biggest when for any time from 0 up to $(k_\omega - 2)T$, the consumed bandwidth is at its maximum $d_{\max}$. Consequently, from relation (5.144), we get the following inequality:

$$(k_\omega - 1)\,(u_{\max} - d_{\max}) < y_{\mathrm{D}} - u_{\max}, \tag{5.145}$$

which gives the estimate of k_ω specified by relation (5.136).

We will now demonstrate that for any time $kT > k_\omega T$, condition (5.142) is indeed satisfied. For that purpose, we take into account some $k > k_\omega$, and we consider two cases: the first one when $\omega(kT) \leq u_{\max}$ and the second one when $\omega(kT) > u_{\max}$.

Case 1. From the definition of functions $\omega(\cdot)$ and $\varphi(\cdot)$ (relations (5.118) and (5.127), respectively), if $\omega(kT) \leq u_{\max}$, then we have

$$\omega\,(kT) = y_{\mathrm{D}} - \varphi\,(kT) \leq u_{\max}. \tag{5.146}$$

From this inequality, it can be easily noticed that condition (5.142) actually holds for any $k > k_\omega$.

Case 2. Now let us consider the second case, i.e., the situation when $\omega(kT) > u_{\max}$. In this situation, in order to show that condition (5.142) holds for any $k > k_\omega$, one can apply the principle of the mathematical induction. We have already demonstrated that there exists such a moment $k_\omega T$ when inequality (5.142) is satisfied. Now, let us assume that for some instant $lT > k_\omega T$ the considered condition holds, and we will show that this implies that the condition is also satisfied at the time instant

$(l+1)T$. Since in the analyzed case $\omega(lT) > u_{\max}$, then $u(lT) = u_{\max}$. Taking into account equation (5.128) and inequality $u_{\max} > d_{\max}$, we get

$$\begin{aligned}\varphi\left[(l+1)\,T\right] &= \varphi\left(lT\right) + u\left(lT\right) - h\left(lT\right) = \varphi\left(lT\right) + u_{\max} - h\left(lT\right)\\ &\geq \varphi\left(lT\right) + u_{\max} - d_{\max} \geq \varphi\left(lT\right) \geq y_{\mathrm{D}} - u_{\max}.\end{aligned} \tag{5.147}$$

Consequently, we may conclude that for any $k > k_\omega$ inequality (5.142) is always satisfied.

Condition (5.142) implies that for any $kT > k_\omega T$, $\omega(kT) \leq u_{\max}$ and $u(kT) = \omega(kT)$. Therefore, it immediately follows from equation (5.122) that relation (5.137) is indeed satisfied for any $k > k_\omega$. This ends the proof. □

Theorem 5.14 shows a direct relation between the rate generated by the algorithm and the number of packets actually sent on the output interface of the bottleneck node at the previous time instant. Moreover, if we compare strategy (5.119) with the controllers presented in Sects. 5.1.3.1 and 5.1.3.2, we can notice that, in addition to constraining high initial flow rate, all three algorithms guarantee that the packet queue length in the bottleneck node buffer does not exceed the demand level y_{D} (see Theorems 5.5, 5.9, and 5.12). However, it comes from Theorems 5.6, 5.10, and 5.13 that the algorithms differ in the value of y_{D} which is proved to ensure full bandwidth utilization. The first two strategies, given by (5.93) and (5.112), are demonstrated to achieve this goal with smaller y_{D} value than controller (5.119). According to the theorems, strategies (5.93) and (5.112) require $y_{\mathrm{D}}^{*} > d_{\max}(n_p + 1)$, whereas controller (5.119) demands $y_{\mathrm{D}}^{**} > u_{\max}(n_p + 1)$ which is bigger than y_{D}^{*} since $u_{\max} > d_{\max}$. Consequently, controller (5.119) requires larger buffer capacity to be reserved at the node to guarantee loss-free and maximally efficient data transfer. The smaller buffer requirements of controllers (5.93) and (5.112) come at a price. It follows from Theorems 5.6, 5.10, and 5.13 that the state of full bandwidth utilization is guaranteed after longer time period since the beginning of the control process (which starts at $kT = 0$). In the case of controller (5.119), the maximum efficiency is ensured for $k = k^{**} > n_p + 1$, whereas in the case of controller (5.93), it is guaranteed for $k \geq k_{\mathrm{VP}} + n_p + 1 > k^{**}$ and in the case of controller (5.112) for $k \geq k_{\mathrm{RL}} + n_p + 1 > k^{**}$ (note that both k_{VP} and k_{RL} are larger than zero).

The properties of the nonlinear controller (5.119) are verified in a series of simulation tests described in the remainder of this section.

Simulation Results

The simulations are run for the same network model which was considered previously in this chapter, i.e., for the connection characterized by the delay consisting of $n_p = 10$ periods, and a single bottleneck node on the data route with bandwidth on the output interface limited by $d_{\max} = 10$ packets per period. Performance of controller (5.119), referred to as the saturating controller in the simulation description, is compared with the other algorithms reported so far in

Table 5.4 Demand queue length selection in Test 1

Controller	y_D [packets]
Linear with LQ optimal plane	$302 > 300$
Nonlinear with time-varying plane	$112 > 110$
Nonlinear employing reaching law	$112 > 110$
Nonlinear with saturation element	$167 > 165$

this chapter, i.e., SM controller with LQ optimal sliding plane (5.67), nonlinear SM controller with time-varying plane (5.93), and reaching-law-based SM controller (5.112). It is assumed that either the channel constraints or the source limitations do not permit transmission rate bigger than 15 packets per period. Hence, the maximum rate which can be assigned by the algorithms is set as $u_{\max} = 15$ packets. Two series of simulations are run for the bandwidth patterns illustrated in Figs. 5.7 and 5.12. The results obtained in the first scenario are presented in section Test 1, and those pertaining to the second scenario in the Test 2 part.

Test 1. First, we verify the controller performance in the presence of the available bandwidth evolving according to the plot shown in Fig. 5.7. We choose parameters of each controller so that (1) the rate assigned to the source does not exceed the allowed maximum of 15 packets and (2) all of the available bandwidth is consumed for data transfer. Consequently, the gain of the SM controller with time-invariant LQ optimal plane (a) is selected as $\gamma = 0.049$ and the demand queue length, according to Theorem 5.2, as $302 > 300$ packets. Notice that the gain selection $\gamma = 0.049$, according to (5.58), corresponds to the weighting factor $w = 0.0026$ used in performance index (5.41). The duration of the plane movement for controller (5.93) (curve b) in the graphs) is chosen as $k_{\mathrm{VP}} = 8$ and the demand queue length, according to Theorem 5.7, as $112 > 110$ packets. Parameter δ governing the reaching law applied in controller (5.112) is set according to (5.115) as $|c_n|u_{\max} = 15$ packets for $0 \le k \le 10$ and $|c_n|(u_{\max} - d_{\max}) = 15 - 10 = 5$ packets for $k > 10$. The demand queue length for this algorithm is adjusted according to Theorem 5.10 as $112 > 110$ packets. Finally, the saturation limit of controller (5.119) is set directly as $u_{\max} = 15$ packets. The demand queue length resulting in the maximally efficient bandwidth usage for this controller is selected according to Theorem 5.13 as $167 > 165$ packets. For convenience, y_D (which also constitute the minimum buffer capacity requirement) are grouped in Table 5.4.

The transmission rate established by the controllers is shown in Fig. 5.40 and the buffer occupancy in Fig. 5.41. The plots depicted in Fig. 5.40 demonstrate that the rate generated by the algorithms is indeed nonnegative and never exceeds 15 packets. We can also notice from these plots how each of the nonlinear controllers (5.93), (5.112), and (5.119) outperforms the linear one (5.67) in the response speed to bandwidth changes. The nonlinear controllers are also superior over the linear one with respect to the buffer capacity requirements. It follows from the curves shown in Fig. 5.40 that in each case the queue length does not grow beyond the demand value given in Table 5.4, thus ensuring loss-free transmission. However, the largest queue level results from the application of the linear scheme (5.67). The largest

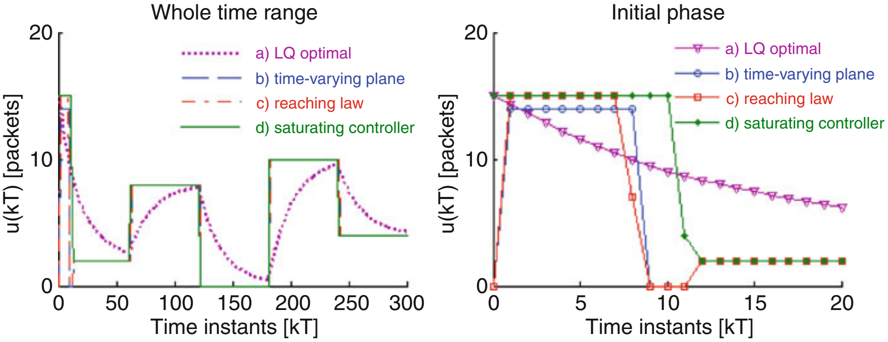
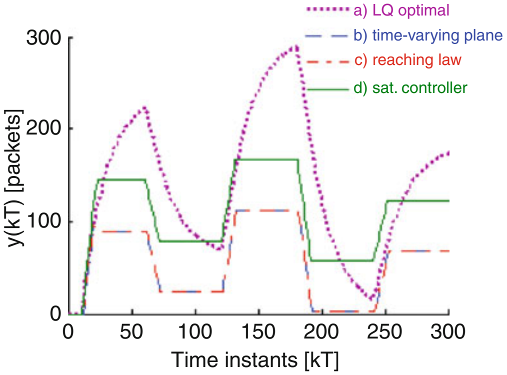


Fig. 5.40 Transmission rate: *a* linear controller (5.67), *b–d* nonlinear controllers (5.93), (5.112), and (5.119)

Fig. 5.41 Queue length: *a* linear controller (5.67), *b–d* nonlinear controllers (5.93), (5.112), and (5.119)



queue length also means that, on average, packets will need to wait for a longer time interval in the buffer before being transmitted towards the destination. This degrades the measure of quality of service represented by mean transfer latency.

We also compare the controllers with respect to performance index (5.41) (with $w = 0.0026$) for a disturbance-free scenario $d(kT) \equiv 0$. The numerical values shown in Table 5.5 indicate that nonlinear controllers perform better than the optimal one with respect to the considered quality criterion. This somewhat counterintuitive observation comes from the fact that the considered nonlinear algorithms exert more efficient control over the maximum rate value and provide better buffer size policy. Indeed, it comes from Theorems 5.2, 5.7, 5.10, and 5.13 that when the considered nonlinear control laws are applied to regulate the flow of data, the demand queue length value can be reduced as compared with the linear controller, but still the conditions of full bandwidth utilization are maintained.

Test 2. We can see from the results of Test 1 that all three nonlinear controllers perform better than the LQ optimal one with respect to buffer capacity savings and quadratic quality criterion. However, a possible drawback of all these nonlinear

Table 5.5 Controller comparison based on performance index (5.41)

Controller	J_2
Linear with LQ optimal plane	3,600
Nonlinear with time-varying plane	1,018
Nonlinear employing reaching law	1,043
Nonlinear with saturation element	1,761

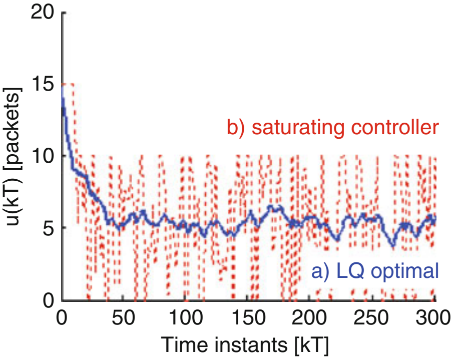


Fig. 5.42 Transmission rate: *a* linear controller (5.67) and *b* nonlinear controller (5.119)

control schemes is propagating high-frequency bandwidth fluctuations into the control signal which is thus more difficult to follow by the transmitters. In this simulation scenario, we compare performance of the LQ optimal controller with the nonlinear one incorporating a saturation element (note that all three nonlinear controllers do not apply rate smoothening, and their response to highly variable bandwidth is quite similar). Therefore, in the second scenario, we run the simulations for the stochastic bandwidth pattern illustrated in Fig. 5.12. The controller parameters are set exactly the same as in Test 1.

The transmission rate generated by the controllers is depicted in Fig. 5.42 and the buffer occupancy in Fig. 5.43. The plots given in Fig. 5.42 show that the rate generated by the algorithms is always confined to the interval [0, 15 packets]. The linear controller provides a smoother control signal in response to the oscillating bandwidth than the nonlinear one. In fact, in the case of the saturating controller, the bandwidth variations are directly translated to the fluctuations of the transmission rate, which is inconvenient since the assigned rate function is more difficult to follow by the data source. Since the linear controller with LQ optimal sliding plane can reduce the degree of oscillations, we conclude that from the point of view of preserving the transmission consistency, it is a preferred solution over the fast-responsive nonlinear controllers. The nonlinear controllers, however, are superior to the linear one with respect to the buffer capacity requirements, as indicated in Table 5.4. It can be seen from Fig. 5.43 that the queue length under the control of the linear scheme is larger than the one resulting from the operation of the saturating controller. Consequently, the linear controller is expected to provide longer average packet transfer delay than the nonlinear strategy. However, in both

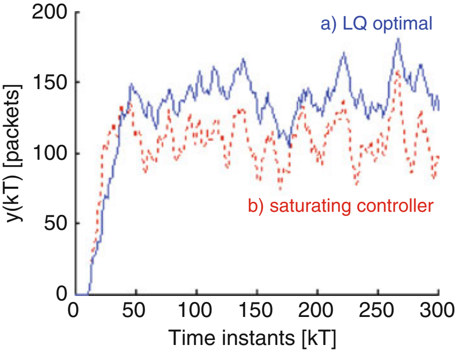


Fig. 5.43 Queue length: *a* linear controller (5.67) and *b* nonlinear controller (5.119)

cases, the queue length remains within the assigned buffer space and is positive. Since we assumed in the simulations that the only source of packet drops in the considered communication system is the congestion, a positive, finite queue length implies that efficient, loss-free packet transfer is ensured.

5.2 Flow Control in a Network with Variable Delay

In the preceding discussion in this chapter, we considered the communication system in which the delay in the feedback loop could be assumed constant. This is well justified in situations where the feedback information is provided by priority-served control units traveling along a fixed path through the network. The transmission along a fixed path implies constant propagation delay. In turn, when control units are not waiting in the node buffers interleaved with packets but are transferred immediately at the output interfaces after the feedback information is incorporated, then their queuing delay is zero. Therefore, when control units are treated with priority over data packets and travel along a fixed route, then their round-trip time, and consequently the delay in the feedback loop, can be considered constant. In fact, the only situation when the queuing delay of control units would be positive in the considered configuration is when the time for calculating and writing the feedback information exceeds the discretization period. In such a case, a control unit arriving at a node would meet the previous one in process by the flow control algorithm. However, in order to keep the network overhead related to the management traffic (the transfer of control units) low, in practical systems, the interval between generating subsequent control units is set much bigger than their processing time.

Unfortunately, in real networks, the delay in the feedback loop rarely remains constant and equal to the nominal value known to the control algorithm. There are at least two important factors which make the delay vary during the transmission. First of all, the route is rarely fixed. Node and link failures, or path optimization procedures, cause modifications in the routing (or switching) tables at the nodes.

As the routing tables governing the path selection are dynamically adapted to the current network topological state, the path taken by packets and feedback carriers does not remain invariant. It may change during the transmission, usually implying different propagation latencies at various stages of the control process. Secondly, if the actual bottleneck for the connection manifests itself, at least temporarily, at another node (not at the one where the flow control algorithm operates), then packets will be subject to time-varying queuing delay imposed by that node. This means that they will arrive at the controlling node with different delay than was expected by the algorithm, posing a serious stability threat. Therefore, in order to describe the network behavior in a more accurate way, and open possibility for robustness improvements, delay variations should be explicitly accounted for both in the network model, and in the controller design procedure.

In this section, we will extend the network model presented in Sect. 5.1 so that it will adequately reflect the phenomena related to delay variations. Next, we will propose two robust controllers which can drive the network into the state of maximum throughput despite unknown latency fluctuations. The first algorithm effectively combines the benefits of LQ optimal SM controller (5.67) with a saturation element in the form of transmission rate limiter. The second robust algorithm extends the idea of the first controller by additionally incorporating a novel compensation mechanism, specifically designed for eliminating the adverse effects of delay variations. We will show that with appropriately tuned controller parameters, both algorithms are able to maintain fast system dynamics without the risk of instability in the presence of arbitrary bandwidth and delay variation patterns. Moreover, we will show that when it is possible to implement the proposed compensation mechanism, the controller designed for the system with constant delay will operate in the system with unknown, time-varying delay, as if the delay was actually fixed.

5.2.1 Network Model

Similarly as in Sect. 5.1, we consider a single flow in the communication network in which the transmission rate of the data source is determined by the controller placed at a network node. The packets emitted by the source pass through a series of nodes operating in the store-and-forward mode without the traffic prioritization to be finally delivered to the destination. Since the position of the bottleneck may change in the configuration considered in this section, in order to avoid the ambiguity, the node at which the control algorithm operates will be termed the controlling node (and not the bottleneck node as considered so far). The feedback information used for rate adjustment is provided by the network to the sources by means of control units (such as acknowledgments in TCP/IP-based networks, or RM cells in ATM). Packets, as well as control units, experience delay as they pass through the nodes and travel on the internode links. In contrast to the previously considered scenario, we no longer expect the delay to remain constant during the transmission. In this section,

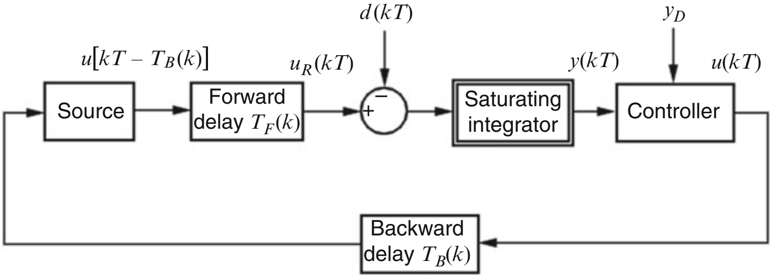


Fig. 5.44 Network model with variable delay – single virtual circuit

it is assumed that the delay of packets and control units may exhibit fluctuations affected by the current network state and selected transmission path, potentially causing instability.

The transmission process in the network with variable delay is illustrated in Fig. 5.44. The source sends packets at discrete instants of time kT, where T is the discretization period and $k = 0, 1, 2, \ldots,$ in the amounts determined by the algorithm operating at the controlling node. Packets travel through the network experiencing variable queuing and propagation latency, and after forward delay $T_\mathrm{F}(k)$, they appear at the controlling node. Then, they are transferred at the node output interface towards the destination in the amounts permitted by the bandwidth at the outgoing link $d(kT)$. The remaining packets are stored in the buffer where they wait for more bandwidth to become available in subsequent time periods. The controller uses the information about past rates and the indication of buffer occupancy $y(kT)$ in relation to the demand queue length y_D to determine the current amount of data $u(kT)$ to be sent by the source. The information about the current rate is extracted at the source from control units with backward delay $T_\mathrm{B}(k)$ after being processed by the controlling node. The time-varying round-trip time $\mathrm{RTT}(k) = T_\mathrm{B}(k) + T_\mathrm{F}(k)$ is assumed to be a multiple of the discretization period, i.e., $\mathrm{RTT}(k) = n_p(k)T$, where $n_p(k)$ and its nominal value $\bar{n}_p$ are positive integers satisfying

$$(1 - \beta)\,\bar{n}_p \le n_p(k) \le (1 + \beta)\,\bar{n}_p. \tag{5.148}$$

Parameter $\beta \in [0, 1)$ represents the tolerance of delay variation occurring, for instance, due to variable queuing latency at the nodes on the data path or path changes enforced by routing algorithm. The relative measure of variation tolerance was chosen instead of the absolute one since in real networks the packets traveling longer distances (and traversing more nodes) are potentially subject to bigger absolute fluctuations. As a result of delay variations, certain assignments (reaching the data source), as well as packets traveling along different paths, may appear out of order and concurrently with other pieces which is not uncommon in today networks [8, 29]. In such situation, it is assumed that all the assignments arriving at the source (and packets reaching the node) in the same period are added to each other so that the congestion problem cannot be resolved by some accidental yet advantageous control or input signal distortion. In this aspect, the proposed model can be regarded more general than those presented in [33, 35], which assumed that the sequence of the emitted packets and feedback carriers is preserved.

The available bandwidth, $d(kT)$, and the utilized one, $h(kT)$, are defined in the same way as in Sect. 5.1.1, i.e., as the *a priori* unknown bounded functions of discrete-time variable kT. The queue length $y(kT)$ evolves according to the intensity of the incoming, $u_{\mathrm{R}}(kT) = u[kT - \mathrm{RTT}(k)]$, and outgoing streams of packets (represented in the model by function $h(\cdot)$). In the situation when RTT is not fixed during the transmission, the dynamics of the queue length $y(\cdot)$ can be defined by means of the following equation:

$$y\left[(k+1)\,T\right] = y\left(kT\right) + u\left[kT - \mathrm{RTT}(k)\right] - h\left(kT\right). \tag{5.149}$$

Therefore, for any $kT \geq 0$, the packet queue length at the controlling node may be expressed as

$$y\left(kT\right) = \sum_{j=0}^{k-1} u\left[jT - \mathrm{RTT}(j)\right] - \sum_{j=0}^{k-1} h\left(jT\right). \tag{5.150}$$

Applying the definition $\mathrm{RTT}(k) = n_p(k)T$, we can rewrite (5.150) as

$$y\left(kT\right) = \sum_{j=0}^{k-1} u\left[jT - n_p(j)T\right] - \sum_{j=0}^{k-1} h\left(jT\right). \tag{5.151}$$

Let us introduce a function

$$\xi\left(kT\right) = \xi_{+}\left(kT\right) - \xi_{-}\left(kT\right), \tag{5.152}$$

where

$$\xi_{+}\left(kT\right) = \sum_{j \in (0, \beta \bar{n}_p]: n_p(k+j) \leq \bar{n}_p - j} u\left[\left(k - \bar{n}_p + j\right)T\right] \tag{5.153}$$

represents the sum of these surplus packets which arrive at the node by the time kT and earlier than expected since their true latency experienced in the neighborhood of kT is smaller than the nominal one. In other words, function $\xi_{+}(kT)$ accounts for all the packets which in the system with constant delay would reach the node after the time instant kT, but in the considered network with *time-varying* delay, they add to the queue buildup by kT since their actual delay is smaller than the nominal one $\bar{n}_p T$. Similarly,

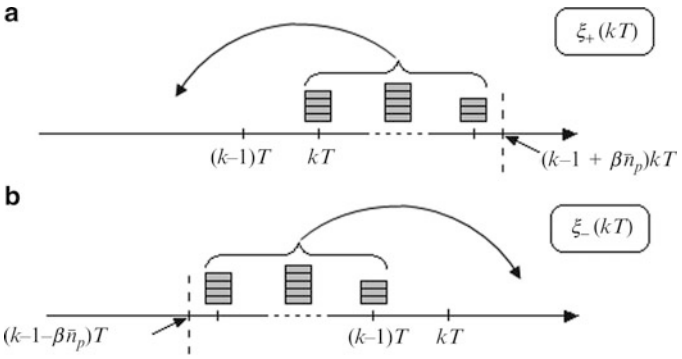


Fig. 5.45 Components of function $\xi(kT)$ (**a**) packets arrive at the node earlier than expected (**b**) packets do not arrive on time due to excessive delay

$$\xi_-(kT) = \sum_{j\in[0,\beta\bar{n}_p]:n_p(k-j)>\bar{n}_p+j} u\left[\left(k-\bar{n}_p-j\right)T\right] \tag{5.154}$$

denotes the sum of these packets which should have arrived at the node by the time *kT*, but which cannot reach the node due to the delay greater than the nominal one. Thus, $\xi_-(kT)$ accounts for all the packets which in the system with constant delay would appear at the note by *kT* but in the network with *time-varying* latency are excessively delayed and feed the queue at some time instant(s) afterwards. The components of function $\xi(\cdot)$ are illustrated in Fig. 5.45.

If one assumes that the rate is bounded by some positive constant $u_{\max}$ (which is the case in any real network), then on the basis of (5.148), the following constraint can be imposed on the values of $\xi(\cdot)$:

$$\forall_{k\geq 0} \; |\xi(kT)| \leq \xi_{\max} = u_{\max}\beta\bar{n}_p. \tag{5.155}$$

With this notation, we can rewrite formula (5.151) for the queue length at instant *kT* in the following way:

$$y(kT) = \sum_{j=0}^{k-1} u\left(jT-\bar{n}_pT\right) + \xi(kT) - \sum_{j=0}^{k-1} h(jT). \tag{5.156}$$

Before the connection is established, there are no packets in the buffer, i.e., for $k<0$, $y(kT)=0$. If we assume that the controller assigns the initial rate at the instant $kT=0$, then taking into account possible delay variations, the first packets arrive at the node no sooner than at $kT = (1-\beta)\bar{n}_p T$, and $y(kT) = 0$ for all $k \leq (1-\beta)\bar{n}_p$. Consequently,

$$\begin{aligned} y(kT) &= \sum_{j=-\bar{n}_p}^{k-\bar{n}_p-1} u(jT) + \xi(kT) - \sum_{j=0}^{k-1} h(jT) \\ &= \sum_{j=0}^{k-\bar{n}_p-1} u(jT) + \xi(kT) - \sum_{j=0}^{k-1} h(jT). \end{aligned} \tag{5.157}$$

Formula (5.157) reflects the nominal system operation (packets arriving due to the nominal delay $\overline{\mathrm{RTT}} = \bar{n}_p T$) affected by perturbation $\xi(\cdot)$ originating from delay variations. The key point to realize in the assumed modeling concept is that it suffices to analyze the influence of perturbation $\xi(\cdot)$ on the queue length only in the neighborhood of kT defined by constraint (5.148). Because the summing operation is commutative, any delay variation, as well as packet reordering, occurring in the previous intervals (earlier than $(k-1-\beta\bar{n}_p)T$) is already accounted for by the first sum in (5.157). Consequently, the packets arriving in the far past can be added as if they had actually reached the node on time, and this will not alter the current queue length.

5.2.1.1 State-Space Representation

The network model with time-varying delay can also be presented in the state space:

$$\begin{aligned} \mathbf{x}[(k+1)T] &= \mathbf{A}\mathbf{x}(kT) + \mathbf{b}u(kT) + \mathbf{v}_1 h(kT) + \mathbf{v}_2 \xi(kT), \\ y(kT) &= \mathbf{q}^{\mathrm{T}}\mathbf{x}(kT), \end{aligned} \tag{5.158}$$

where $\mathbf{x}(kT) = [x_1(kT)\ \ x_2(kT)\ \ x_3(kT), \ldots, \ x_n(kT)]^{\mathrm{T}}$ is the state vector with $x_1(kT) = y(kT)$ representing the bottleneck queue length at instant kT, and the remaining state variables $x_j(kT) = u[(k-n+j-1)T]$ for any $j = 2, 3, \ldots, n$ equal to the delayed input signal $u(\cdot)$, exactly as in the fixed delay scenario. The state matrix $\mathbf{A}_{n\times n}$, input $\mathbf{b}_{n\times 1}$, and output $\mathbf{q}_{n\times 1}$ vectors are defined exactly as in (5.8). Vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ of size $n \times 1$ are given as

$$\mathbf{v_1} = \begin{bmatrix} -1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{v}_2 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{5.159}$$

The system order $n = \bar{n}_p + 1$, and the demand state vector $\mathbf{x_d} = [y_{\mathrm{D}}\ 0\ 0 \ldots\ 0]^{\mathrm{T}}$.

In the following sections, we will present two control laws for the network with variable delay described by (5.158) and (5.159). We will also prove a number of the controller advantageous properties related to regulating the flow of data.

5.2.2 SM Controller with Saturation

In this section, we present a control algorithm which will be demonstrated to provide robustness with respect to uncertain, fluctuating delay in the considered network. However, before we state the control law, we first elaborate on the adverse effects of delay variations degrading the transmission consistency and discuss the importance of taking appropriate measures to guarantee communication system stability. These issues are addressed in Example 5.4.

Example 5.4. Let us study the rate allocation in a connection characterized by the nominal round-trip time $\bar{n}_p T = 5T$. We assume that the source transmission rate is regulated according to control law (5.67) adjusted for a dead-beat scheme, i.e., with the gain $\gamma = 1$. The demand queue length y_D is set equal to 10 packets. Consequently, in this example, we analyze the rate allocation performed according to the following equation:

$$u(kT) = y_\mathrm{D} - y(kT) - \sum_{j=k-\bar{n}_p}^{k-1} u(jT) = 10 - y(kT) - \sum_{j=k-5}^{k-1} u(jT), \quad (5.160)$$

where $u(kT)$ specifies the number of packets to be sent by the source. For the purpose of exposition, we assume that there is no bandwidth available at the node output interface in the whole transmission process ($d(\cdot) \equiv 0$). If the delay in the feedback loop is constant and perfectly known to the controller, algorithm (5.160) generates the rate

$$u(kT) = \begin{cases} 10 \text{ packets} & \text{for} \quad kT = 0, \\ 0 \text{ packets} & \text{for} \quad kT > 0. \end{cases}$$

The source sends the initial amount of packets after backward propagation delay. These packets arrive at the node at $kT = 5T$ and are stored in the buffer for all time afterwards since there is no bandwidth available to drain the queue. This means that the queue length

$$y(kT) = \begin{cases} 0 \text{ packets} & \text{for} \quad kT \le 5T, \\ 10 \text{ packets} & \text{for} \quad kT > 5T. \end{cases}$$

Let us investigate the rate assignment performed by controller (5.160), and the resultant queue length evolution, for the case of time-varying delay (5.148) with the tolerance $\beta = 0.2$. This means that the first packets sent according to the allocation at $kT = 0$ may arrive at the node:

1. At $kT = 5T$, if the instantaneous delay matches the nominal one.
2. At $kT = 4T$, if the instantaneous delay is shorter than the nominal one.
3. At $kT = 6T$, if the instantaneous delay exceeds the nominal one.

Note that since the tolerance $\beta = 0.2$, the delay range is specified as $4T \leq \mathrm{RTT}(k) \leq 6T$. As point 1 reflects the situation of a fixed and perfectly known delay considered above, the main emphasis will be placed on the networking events taking place in situations 2 and 3.

Case 1. First, we analyze the circumstances of the network latency matching the nominal delay. If packets arrive at the node at $kT = 5T$, the controller operation agrees with the actual network state. Thus, we have the rate allocation $u(0) = 10$ packets and $u(kT) = 0$ for $k > 0$. The resulting queue length $y(kT) = 0$ for $k \leq 5$ and $y(kT) = 10$ packets for $k > 5$, precisely as indicated above. The system is asymptotically stable.

Case 2. When packets arrive with delay $\mathrm{RTT}(4) = 4T$ at instant $kT = 4T$, i.e., one period earlier than expected, then from (5.149), we get

$$\begin{aligned} y\,(5T) &= y\,(4T) + u\,[4T - RTT(4)] - h\,(4T) \\ &= y\,(4T) + u(0) - h\,(4T) = 0 + 10 - 0 = 10 \text{ packets.} \end{aligned}$$

However, the rate assignment at $kT = 5T$ is not zero as in the case of transmission with constant delay. This happens because even though the error at the output $y_{\mathrm{D}} - y(kT)$ vanishes from expression (5.160), the rate history $\sum_{j=k-5}^{k-1} u\,(jT)$ still contains the packets assigned for the source at $kT = 0$. We get

$$\begin{aligned} &u(0) = 10 \text{ packets,} \\ &u\,(0 < kT < 5T) = 0, \\ &u\,(5T) = y_{\mathrm{D}} - y\,(5T) - \sum_{j=0}^{4} u\,(jT) = 10 - 10 - 10 = -10 \text{ packets.} \end{aligned}$$

Consequently, in the considered scenario, the packet arrival earlier than predicted by the controller results in a negative rate value calculated at instant $kT = 5T$. This is clearly not appropriate from the point of view of networking application since a negative rate would imply retrieving from the network the packets which have already been sent by the source towards the destination.

Case 3. We consider the circumstances of a prolonged delay, i.e., the case when packets arrive at $kT = 6T$ instead of $kT = 5T$ (one period after they were expected). We get from (5.150)

$$y\,(kT \le 6T) = 0.$$

This means that the rate assignment at $kT = 6T$ cannot be zero, as in the case of the network with constant delay. Note that the error at the output $y_{\mathrm{D}} - y(kT)$ is positive. However, the rate history $\sum_{j=1}^{6-1} u\,(jT)$ no longer encompasses the initial assignment from $kT = 0$, and the history does not compensate the output error. We have

$$u(0) = 10 \text{ packets},$$
$$u\,(0 < kT < 6T) = 0,$$
$$u\,(6T) = y_{\mathrm{D}} - y\,(6T) - \sum_{j=1}^{6-1} u\,(jT) = 10 - 0 - 0 = 10 \text{ packets}.$$

The packets reaching the node at $kT = 6T$ contribute to the queue buildup at $kT = 7T$, which leads to the rate assignment performed according to (5.160) at $kT = 7T$

$$u\,(7T) = y_{\mathrm{D}} - y\,(7T) - \sum_{j=2}^{7-1} u\,(jT) = 10 - 10 - 10 = -10 \text{ packets}.$$

Therefore, we obtain an inappropriate (negative) rate value. Moreover, a stability threat arises since there are in-flight packets in the network due to the assignment from $kT = 6T$, which will further disrupt the flow control process in the analyzed connection.

The discussion presented in this example concludes with a simulation run for the system with delay varying according to the periodic pattern shown in Fig. 5.46. The values indicated in the graph reflect the actual delay experienced by the packets sent according to the assignments generated at consecutive time instants. Consequently, the value of delay shown in Fig. 5.46 at $kT = 0$ $(4T)$ means that the packets sent due to the assignment generated by the controller at $kT = 0$ will arrive at the node with 4 period delay at instant $kT = 4T$. In the test, we assume that the traffic in the network is regulated by controller (5.160) operating on the basis of the nominal delay of $5T$. The rate assignments performed by the controller (with negative signals permitted for the purpose of analysis) are depicted in Fig. 5.47, and the buffer occupancy in Fig. 5.48. It is clear from the graphs that the controller, which guarantees very efficient regulation properties in the communication system with fixed delay, is no longer appropriate for the network with variable latency. In the analyzed scenario, the system becomes unstable.

Fig. 5.46 Delay variations

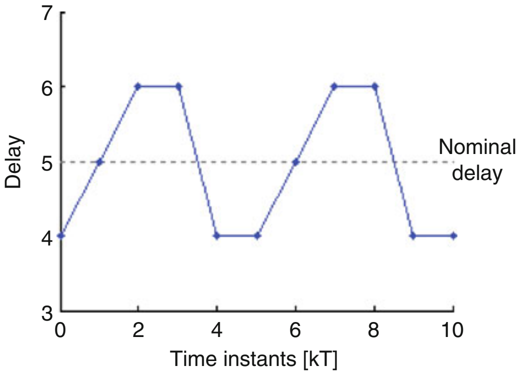


Fig. 5.47 Transmission rate

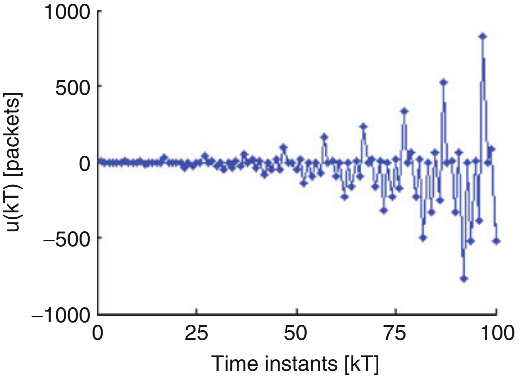
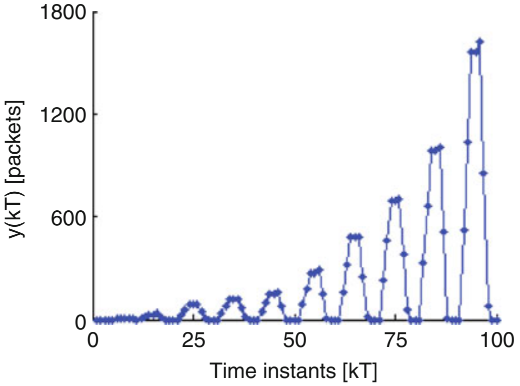


Fig. 5.48 Packet queue length



Below, we propose a nonlinear control strategy which provides the desired robustness to unknown delay variations.

5.2.2.1 Proposed Control Strategy

The controllers presented previously in this chapter were designed with the assumption of perfect knowledge of delay that stayed unchanged during the whole transmission process. However, as was analyzed in detail in Example 5.4, in the situation when the true delay differs from the value used in the calculations, the controllers developed for the nominal system may generate negative transmission rates and even drive the system into instability. In order to ensure closed-loop stability, and guarantee feasible rate assignments, we need to introduce structural modifications into the basic algorithms discussed so far. We propose here to combine the benefits of smooth response to the changing networking conditions provided by SM controller with LQ optimal sliding plane (5.67) with the robustness enhancement resulting from the application of a saturation element. Consequently, the transmission rate in the network with time-varying delay is calculated from the following algorithm:

$$u\left(kT\right)=\begin{cases}0, & \text{if} \quad \omega_{\gamma}\left(kT\right)<0,\\ \omega_{\gamma}\left(kT\right), & \text{if} \quad 0\le\omega_{\gamma}\left(kT\right)\le u_{\max},\\ u_{\max}, & \text{if} \quad \omega_{\gamma}\left(kT\right)>u_{\max},\end{cases} \tag{5.161}$$

where $u_{\max} > d_{\max}$, and function $\omega_{\gamma}(\cdot)$ is defined as

$$\omega_{\gamma}\left(kT\right)=\gamma\left[y_{\mathrm{D}}-y\left(kT\right)-\sum_{j=k-\bar{n}_p}^{k-1}u\left(jT\right)\right] \tag{5.162}$$

with the gain $\gamma \in (0, 1]$. Notice that this algorithm can be regarded as a modification of the one given by (5.117) and (5.118) with function $\omega_{\gamma}(\cdot)$ replacing $\omega(\cdot)$ (the two functions are identical for $\gamma = 1$), and additional transmission rate constraint introduced at the zero level. Thus, the rate assigned according to the robust control law (5.161) and (5.162) is always limited to the interval $[0, u_{\max}]$, which guarantees feasible transmission rate allocation.

In the next section, we formulate important properties of control law (5.161) and (5.162). The described properties are demonstrated analytically, and afterwards, the algorithm performance is verified in numerical tests.

5.2.2.2 Properties of the Proposed Strategy

The properties of nonlinear controller (5.161) and (5.162) will be specified in two theorems. The first proposition shows how to select the buffer capacity to always accommodate the entire packet queue and in this way eliminate losses originating from congestion. The second theorem states that with the appropriately chosen demand queue length, there are always some packets in the buffer, which implies that all of the available bandwidth at the bottleneck link is used for data transfer.

Theorem 5.15. *If controller (5.161) with function $\omega_\gamma(\cdot)$ defined by (5.162) is applied to system (5.158) and (5.159), then the queue length in the bottleneck node buffer is always upper-bounded, i.e., for any $k \geq 0$,*

$$y\,(kT) \leq y_{\max} = y_{\mathrm{D}} + u_{\max} + \xi_{\max}. \tag{5.163}$$

Proof. It follows from the algorithm definition and the system initial conditions that the buffer at the bottleneck node is empty for any $k \leq (1-\beta)\bar{n}_p$. Consequently, it is sufficient to show that the proposition holds for all $k > (1-\beta)\bar{n}_p$. Let us consider some integer $l > (1-\beta)\bar{n}_p$ and the value of function $\omega_\gamma(\cdot)$ at the time instant lT. Two cases ought to be analyzed: the situation when $\omega_\gamma(lT) \geq 0$ and the circumstances when $\omega_\gamma(lT) < 0$.

Case 1. We investigate the situation when $\omega_\gamma(lT) \geq 0$. Directly from the definition of function $\omega_\gamma(\cdot)$, (5.162), we get

$$\omega_\gamma\,(lT) = \gamma\left[y_{\mathrm{D}} - y\,(lT) - \sum_{j=l-\bar{n}_p}^{l-1} u\,(jT)\right] \geq 0, \tag{5.164}$$

which leads to

$$y\,(lT) \leq y_{\mathrm{D}} - \sum_{j=l-\bar{n}_p}^{l-1} u\,(jT). \tag{5.165}$$

From the algorithm definition, in turn, it follows that $u(\cdot)$ is always nonnegative, which implies $y(lT) \leq y_{\mathrm{D}}$. This ends the first part of the proof.

Case 2. In the second part of the proof, we analyze the situation when $\omega_\gamma(lT) < 0$. First, we find the last instant $l_1T < lT$ when $\omega_\gamma(\cdot)$ was nonnegative. According to (5.162), $\omega_\gamma(0) = \gamma y_{\mathrm{D}} > 0$ since both γ and y_{D} are positive, and moment l_1T indeed exists. If $\omega_\gamma(l_1T) \geq 0$, then, with analogy to (5.164) and (5.165), we have

$$\omega_\gamma\,(l_1T) = \gamma\left[y_{\mathrm{D}} - y\,(l_1T) - \sum_{j=l_1-\bar{n}_p}^{l_1-1} u\,(jT)\right] \geq 0 \tag{5.166}$$

and

$$y\,(l_1T) \leq y_{\mathrm{D}} - \sum_{j=l_1-\bar{n}_p}^{l_1-1} u\,(jT). \tag{5.167}$$

The queue length at instant lT can be expressed relative to $y(l_1T)$ as

$$y\,(lT) = y\,(l_1T) + \sum_{j=l_1-\bar{n}_p}^{l-\bar{n}_p-1} u\,(jT) + \xi\,(lT) - \sum_{j=l_1}^{l-1} h\,(jT). \tag{5.168}$$

Applying inequality (5.167), we get

$$\begin{aligned} y\,(lT) &\le y_{\mathrm{D}} - \sum_{j=l_1-\bar{n}_p}^{l_1-1} u\,(jT) + \sum_{j=l_1-\bar{n}_p}^{l-\bar{n}_p-1} u\,(jT) + \xi\,(lT) - \sum_{j=l_1}^{l-1} h\,(jT) \\ &\le y_{\mathrm{D}} + \sum_{j=l_1}^{l-\bar{n}_p-1} u\,(jT) + \xi\,(lT) - \sum_{j=l_1}^{l-1} h\,(jT). \end{aligned} \tag{5.169}$$

The algorithm sets a nonzero rate for the last time before lT at instant l_1T, and this value could be as large as $u_{\max}$. Consequently, the sum

$$\sum_{j=l_1}^{l-\bar{n}_p-1} u\,(jT) = u\,(l_1T) \le u_{\max}. \tag{5.170}$$

Since the utilized bandwidth is always nonnegative, then using the condition $\xi(lT) \le \xi_{\max}$, we obtain from (5.169) and (5.170) the following estimate of the queue length at instant lT:

$$\begin{aligned} y\,(lT) \le & y_{\mathrm{D}} + u\,(l_1T) + \xi\,(lT) - \sum_{j=l_1}^{l-1} h\,(jT) \\ \le & y_{\mathrm{D}} + u\,(l_1T) + \xi\,(lT) - 0 \le y_{\mathrm{D}} + u_{\max} + \xi_{\max}. \end{aligned} \tag{5.171}$$

This concludes the second part of the reasoning and completes the proof. □

Theorem 5.15 shows how to adjust the buffer size at the controlling node to eliminate losses originating from congestion. This property is achieved irrespective of bandwidth or delay variations, which need not be correlated with each other. In the second theorem, presented below, we indicate how the demand queue length should be selected in order to ensure full bandwidth utilization.

Theorem 5.16. *If controller (5.161) with function $\omega_\gamma(\cdot)$ defined by (5.162) is applied to system (5.158) and (5.159), and the demand queue length satisfies the following inequality:*

$$y_D > u_{\max}\left(\bar{n}_p + 1/\gamma\right) + d_{\max} + \xi_{\max}, \tag{5.172}$$

then for any $k \geq (1 + \beta)\bar{n}_p + n_{\max}$, *where* $n_{\max} = y_{\max}/(u_{\max} - d_{\max})$, *the queue length is strictly positive.*

Proof. The theorem assumption implies that we deal with time instants $kT \geq (1 + \beta)\bar{n}_p T + n_{\max} T$. Considering some integer $l \geq (1 + \beta)\bar{n}_p + n_{\max}$ and the value of signal $\omega_\gamma(\cdot)$ at instant lT, we may distinguish two cases: the situation when $\omega_\gamma(lT) < u_{\max}$ and the circumstances when $\omega_\gamma(lT) \geq u_{\max}$.

Case 1. First, we consider the situation when $\omega_\gamma(lT) < u_{\max}$. From the definition of function $\omega_\gamma(\cdot)$, we obtain

$$\omega_\gamma\,(lT) = \gamma\left[y_{\mathrm{D}} - y\,(lT) - \sum_{j=l-\bar{n}_p}^{l-1} u\,(jT)\right] < u_{\max}, \tag{5.173}$$

which after the term rearrangement results in

$$y\,(lT) > y_{\mathrm{D}} - u_{\max}/\gamma - \sum_{j=l-\bar{n}_p}^{l-1} u\,(jT). \tag{5.174}$$

The transmission rate generated according to (5.161) is always bounded by $u_{\max}$, which implies

$$y\,(lT) > y_{\mathrm{D}} - u_{\max}/\gamma - \bar{n}_p u_{\max} = y_{\mathrm{D}} - u_{\max}\left(\bar{n}_p + 1/\gamma\right). \tag{5.175}$$

Using assumption (5.172), we get $y(lT) > 0$, which concludes the first part of the proof.

Case 2. In the second part of the proof, we investigate the situation when $\omega_\gamma(lT) \geq u_{\max}$. First, we find the last moment $l_1 T < lT$ when signal $\omega_\gamma(\cdot)$ was smaller than $u_{\max}$. It comes from Theorem 5.15 that the queue length never exceeds the value of $y_{\max}$. Furthermore, from (5.2), we know that the consumed bandwidth $h(\cdot)$ is limited by $d_{\max}$. Thus, the maximum interval $n_{\max}T$ during which the controller may continuously generate the maximum transmission rate is determined as $n_{\max}T = Ty_{\max}/(u_{\max} - d_{\max})$, and instant $l_1 T$ does exist. Moreover, from the theorem assumptions, we get $l_1 T \geq (1 + \beta)\bar{n}_p T$, which means that by the time $l_1 T$ the first packets reach the node, no matter the delay variation.

The value of $\omega_\gamma(l_1 T) < u_{\max}$. Consequently, following a similar reasoning as presented in (5.173)–(5.175), we arrive at $y(l_1 T) > 0$ and

$$y\,(lT) > y_{\mathrm{D}} - \frac{u_{\max}}{\gamma} - \sum_{j=l_1-\bar{n}_p}^{l_1-1} u\,(jT) + \sum_{j=l_1-\bar{n}_p}^{l-\bar{n}_p-1} u\,(jT) + \xi\,(lT) - \sum_{j=l_1}^{l-1} h\,(jT). \tag{5.176}$$

After performing algebraic manipulations on the first two sums in (5.176), we get

$$\begin{aligned} y\left(lT\right) > & y_{\mathrm{D}} - \frac{u_{\max}}{\gamma} + \sum_{j=l_1}^{l-1} u\left(jT\right) - \sum_{j=l-\bar{n}_p}^{l-1} u\left(jT\right) + \xi\left(lT\right) - \sum_{j=l_1}^{l-1} h\left(jT\right) \\ = & y_{\mathrm{D}} - \frac{u_{\max}}{\gamma} + u\left(l_1 T\right) + \sum_{j=l_1+1}^{l-1} u\left(jT\right) \\ & - \sum_{j=l-\bar{n}_p}^{l-1} u\left(jT\right) + \xi\left(lT\right) - \sum_{j=l_1}^{l-1} h\left(jT\right). \end{aligned} \tag{5.177}$$

Recall that $l_1 T$ was the last instant before lT when the controller calculated rate smaller than $u_{\max}$. This rate could be as low as zero. Afterwards, the algorithm generates the maximum rate value, and the first sum in (5.177) reduces to $u_{\max}(l-1-l_1)$. Since for any k, $u(kT) \leq u_{\max}$, the second sum is upper-bounded by $u_{\max}\bar{n}_p$. Thus, we have

$$y\left(lT\right) > y_{\mathrm{D}} - \frac{u_{\max}}{\gamma} + 0 + u_{\max}\left(l-1-l_1\right) - u_{\max}\bar{n}_p + \xi\left(lT\right) - \sum_{j=l_1}^{l-1} h\left(jT\right). \tag{5.178}$$

Using the fact that $\xi(\cdot) \geq -\xi_{\max}$ and $h(\cdot) \leq d_{\max}$, we get a further estimate of the queue length at instant lT

$$y\left(lT\right) > y_{\mathrm{D}} - u_{\max}/\gamma + u_{\max}\left(l-1-l_1\right) - u_{\max}\bar{n}_p - \xi_{\max} - d_{\max}\left(l-l_1\right). \tag{5.179}$$

Using the theorem assumption (5.172), we obtain

$$\begin{aligned} y\left(lT\right) > & u_{\max}\left(\bar{n}_p + 1/\gamma\right) + d_{\max} + \xi_{\max} \\ & - u_{\max}/\gamma + u_{\max}\left(l-1-l_1\right) - u_{\max}\bar{n}_p - \xi_{\max} - d_{\max}\left(l-l_1\right) \\ = & u_{\max}\left(l-1-l_1\right) - d_{\max}\left(l-1-l_1\right). \end{aligned} \tag{5.180}$$

Finally, since $l > l_1$ and $u_{\max} > d_{\max}$, we arrive at

$$y\left(lT\right) > \left(u_{\max} - d_{\max}\right)\left(l-1-l_1\right) \geq 0. \tag{5.181}$$

This conclusion ends the proof. □

In the remainder of this section, we will present the simulation results demonstrating the properties of the designed robust control law (5.161) and (5.162).

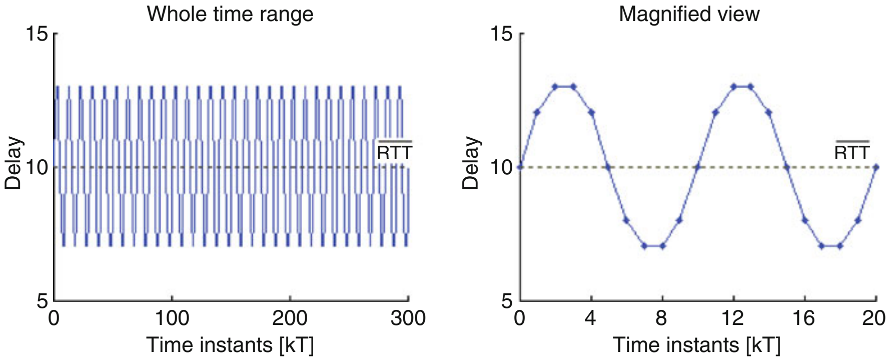


Fig. 5.49 Delay variations in Test 1

5.2.2.3 Simulation Results

We consider the flow control process in the connection characterized by the nominal round-trip time $\bar{n}_p T = 10T = 100\,\text{ms}$. The true delay varies in a way unknown to the controller in the range specified by $\beta = 0.3$. We verify the performance of controller (5.161) and (5.162) with the maximum transmission rate set as $u_{\max} = 11$ packets. This results in the estimate of the maximum perturbation $\xi_{\max}$ calculated according to (5.155) equal to $u_{\max}\beta\bar{n}_p = 33$ packets. Similarly as in the previous sections, two series of simulation tests are run: one series for the bandwidth illustrated in Fig. 5.7 and another one for the stochastic pattern depicted in Fig. 5.12. In each case, the maximum bandwidth $d_{\max} = 10$ packets.

Test 1. In the first simulation scenario, we verify the controller performance in response to the available bandwidth evolving according to the function depicted in Fig. 5.7. The delay in the feedback loop is assumed to vary according to the following equation:

$$\text{RTT}(k) = \left\lfloor \left[1 + \beta \,\sin\left(2\pi kT/\bar{n}_p\right)\right] \bar{n}_p T \right\rfloor, \tag{5.182}$$

where $\lfloor x \rfloor$ denotes the integer part of x. The delay variations are illustrated in Fig. 5.49. To facilitate the analysis, we present the delay in such a way that the value shown in the plot at a particular time instant, say kT, corresponds to the actual RTT of the assignment sent to the source by the controller at kT. Referring to the graph, the packets from the assignments performed at 0, T, $2T$, $3T, \ldots,$ actually arrive at the controlling node with delay $10T$, $12T$, $13T$, $13T, \ldots,$ at instants $10T$, $13T$, $15T$, $16T, \ldots,$ and so on. Several simulations are run for different dynamics adjusted through the choice of the gain constant γ. The demand queue length is set according to the guidelines given in Theorem 5.16 so that full bandwidth utilization is achieved. In order to ensure loss-free transmission, the buffer size is set equal to the value indicated by Theorem 5.15. The actual parameters used in the test are summarized in Table 5.6.

Table 5.6 Controller parameters

Controller gain γ	Demand queue length y_D [packets]	Buffer size $y_{\max}$ [packets]
0.259	$196 > 195$	240
0.618	$172 > 171$	216
1	$165 > 164$	209

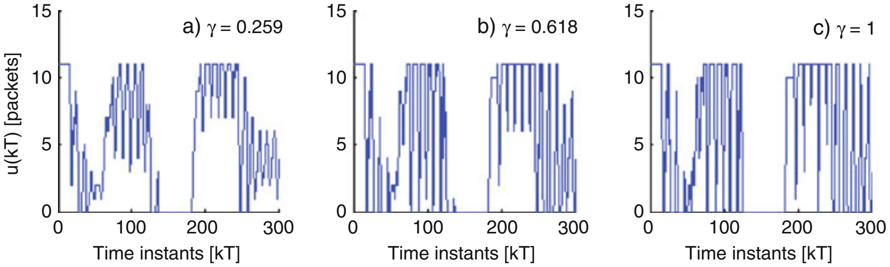


Fig. 5.50 Generated transmission rate: (**a**) $\gamma = 0.259$, (**b**) $\gamma = 0.618$, and (**c**) $\gamma = 1$

The test results are shown in Figs. 5.50–5.52: the transmission rate established by the controller in Fig. 5.50, the incoming packet rate in Fig. 5.51, and the resulting buffer queue length in Fig. 5.52. We can see from the graphs that the controller quickly responds to the sudden changes in the available bandwidth, the queue length does not increase beyond the buffer capacity, and it never drops to zero after the initial phase. In consequence, packet losses are eliminated and the maximum throughput is guaranteed. Moreover, even though the delay undergoes large fluctuations (in the range of 30% from the nominal value) which severely disrupts the packet incoming rate (see Fig. 5.51), the degree of oscillations in the output variable (the queue length) is reduced. The plots in Fig. 5.50 indicate that the system dynamics and the degree of oscillations depend on the choice of the controller gain. As γ increases, the controller reacts faster to the demand changes, and as γ is reduced, responsiveness drops. However, as can be learned from Fig. 5.50, placing more impact on the output error elimination (large γ) amplifies the oscillations in control signal resulting from delay variations. Therefore, in a majority of practical settings in networks with variable delay, γ should not exceed 0.618 which corresponds to the golden-ratio controller discussed in Sect. 5.1.2.

We can notice from Fig. 5.51 that because of time-varying delay, the packets from certain assignments arrive concurrently with other pieces sent by the source. Such situation may occur, for instance, if packets are directed along different routes when traveling towards the controlling node. As a consequence, the packet incoming rate may exceed the maximum value of the rate assigned by the controller $u_{\max} = 11$ packets, which typically leads to stability problems in the considered class of systems. In other circumstances, the packets may arrive too early (or be excessively delayed) leaving a gap in the incoming rate at the instant when they should appear at the node if transfer with nominal delay could have been ensured. This poses

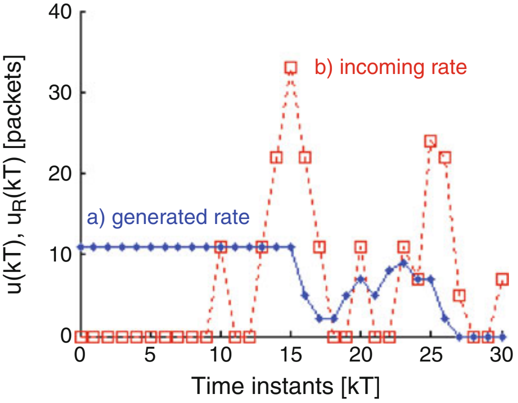


Fig. 5.51 *a* Generated transfer rate and *b* incoming rate in the initial phase for $\gamma = 0.259$

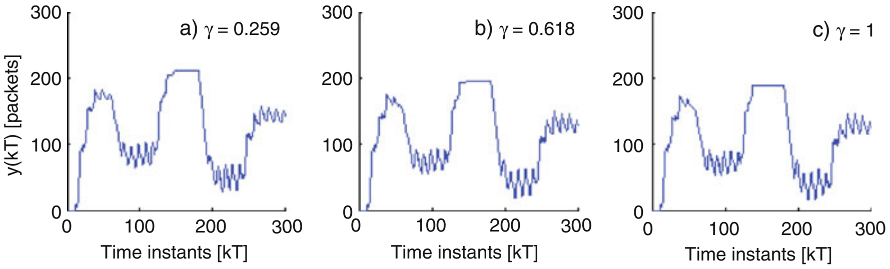


Fig. 5.52 Queue length: (**a**) $\gamma = 0.259$, (**b**) $\gamma = 0.618$, and (**c**) $\gamma = 1$

a threat of bandwidth underutilization. As shown in Theorems 5.15 and 5.16, the presented controller (5.161) and (5.162) guarantees both the system stability and full bandwidth utilization in the network with uncertain, variable delay.

Let us study the phenomena related to delay variations in detail for the dynamical setting $\gamma = 0.259$ illustrated in Fig. 5.50. It follows from the plots that in the interval $[0, 15T]$ the controller assigns the rate of 11 packets. In the network with constant delay $\bar{n}_p T = 10T$, the packets from these assignments would appear at the node at $10T$, $11T$, $12T, \ldots, 25T$. However, because in the network analyzed here the delay is assumed to vary according to the pattern shown in Fig. 5.49, the packets from the initial assignments performed at

0	T	$2T$	$3T$	$4T$	$5T$	$6T$	$7T$	$8T$	$9T$	$10T$	$11T$	$12T$	$13T$	$14T$	$15T$

arrive with delay

$10T$	$12T$	$13T$	$13T$	$12T$	$10T$	$8T$	$7T$	$7T$	$8T$	$10T$	$12T$	$13T$	$13T$	$12T$	$10T$

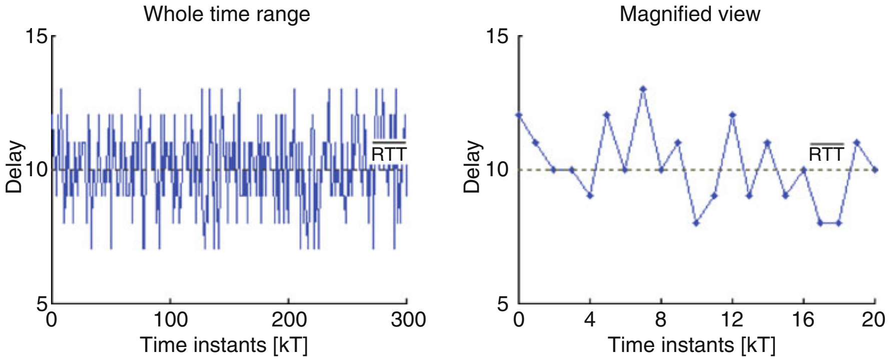


Fig. 5.53 Delay variations in Test 2

at instants

$10T$	$13T$	$15T$	$16T$	$16T$	$15T$	$14T$	$14T$	$15T$	$17T$	$20T$	$23T$	$25T$	$26T$	$26T$	$25T$.

Consequently, the packets from the assignments performed at $2T$, $5T$, and $8T$ arrive concurrently at the node at $kT = 15T$, following the initial positive ripple in delay. This results in the incoming rate of 33 packets at the instant $kT = 15T$. Similarly, the packets from assignment 3 and 4 arrive at $kT = 16T$, resulting in the incoming rate of 22 packets. Although it is predicted that 11 packets will arrive at $11T$ and $12T$, due to the fluctuations of delay, no piece of the data stream generated by the source reaches the node at these time instants. Thus, $u_R(11T) = u_R(12T) = 0$, and a gap in the incoming rate is formed, which may lead to decreased bandwidth utilization at the controlling node. The remaining events can be studied in a similar way by analyzing the plots sketched in Figs. 5.50 and 5.51.

Test 2 In the second simulation scenario, we verify the controller performance in a stochastic environment. The available bandwidth, illustrated in Fig. 5.12, is assumed to follow the normal distribution with mean $d_\mu = 5$ packets and standard deviation $d_\delta = 5$ packets, $D_{\text{norm}}(5,\ 5)$. The delay in the feedback loop changes randomly according to the normal distribution $D_{\text{norm}}(10T,\ 1.41T)$. It is depicted in Fig. 5.53, which shows RTTs of the assignments placed by the controller in subsequent time intervals. The controller parameters are set as in Test 1.

The results of the simulations are shown in Figs. 5.54–5.56: the transmission rate generated by the algorithm in Fig. 5.54, the received packet number in Fig. 5.55, and the buffer queue length in Fig. 5.56.

It follows from the graphs that the packet queue length remains finite in the entire time range. This means that the system is stable in the Lyapunov sense, and it remains robust to delay and bandwidth variations. However, the established transmission rate depicted in Fig. 5.54 indicates increased amplitude of oscillations as compared to the case of the system with constant delay illustrated in Fig. 5.13. Moreover, the degree of oscillations grows with the increase in the controller

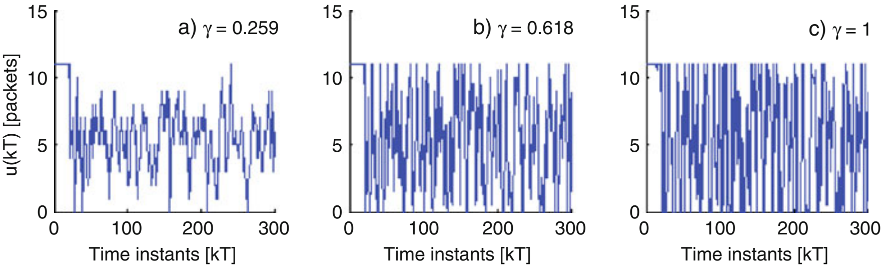


Fig. 5.54 Generated transmission rate: (**a**) $\gamma = 0.259$, (**b**) $\gamma = 0.618$, and (**c**) $\gamma = 1$

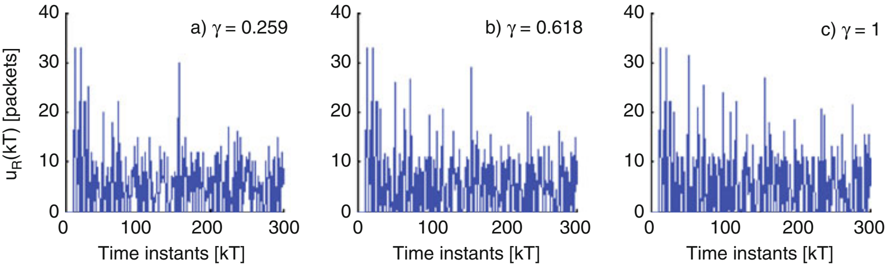


Fig. 5.55 Incoming rate: (**a**) $\gamma = 0.259$, (**b**) $\gamma = 0.618$, and (**c**) $\gamma = 1$

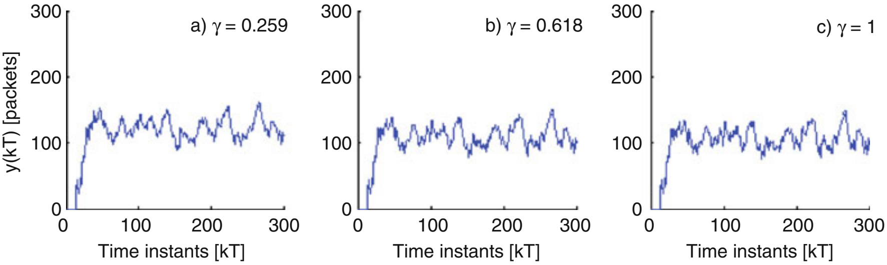


Fig. 5.56 Queue length: (**a**) $\gamma = 0.259$, (**b**) $\gamma = 0.618$, and (**c**) $\gamma = 1$

dynamics, and the assigned rate can be difficult to reproduce by slowly responding transmitters. The reason for the increased level of oscillations as compared to the constant-delay case is the perturbed input signal shown in Fig. 5.55. The actual input signal significantly differs from the function generated by the controller and used to predict the number of in-flight packets in (5.162). However, note that the queue length illustrated in Fig. 5.56 does not grow beyond the buffer capacity and remains positive after the initial phase. Consequently, even though we deal with both bandwidth and delay variations, the throughput is kept at the maximum.

The test results show that robustness to delay and bandwidth variations is achieved by controller (5.161) and (5.162). However, in order to keep the oscillations of the control signal reasonable, responsiveness to the changes of networking conditions needs to be reduced. In the next section, we will present a mechanism to compensate for the adverse effects of delay variations which allows for regenerating the control signal so that it is established as if the delay was constant and equal to the estimate used by the controller.

5.2.3 Delay Variability Compensation

The traditional dead-time compensators (DTCs), such as the popular Smith predictor (SP) [36], are vulnerable to plant modeling errors and imprecise delay estimates. In the situations when delay cannot be determined accurately, exhibits unpredictable variations, or is simply unknown and not accessible for measurement, the application of a traditional DTC guarantees neither appropriate system performance nor stability. Having noted these limitations, many researchers intended to either provide modifications to the basic structure of typical SP-like compensators (see, e.g., the overview given in [31]) or use alternative configurations and perform robustness analysis for the model and delay mismatches [38]. With better modeling techniques and application of robust control methods, inaccuracy of the plant model parameters usually poses a smaller stability threat than uncertainty in delay. Indeed, as stated in [24], in the delayed control of a stable plant, the SP guarantees the same robustness level as the plant controller does for the delay-free process even in the case of multiplicative or structured uncertainties. Therefore, it is of utmost importance to study the effects of *delay uncertainty* in assessing the controller performance and system stability. The solutions presented so far in the literature for the systems with time-varying delay typically consider linear plants, or applications where some form of delay estimation (or measurement) is available, for example, [5, 28, 37]. It is also a common practice in the systems with uncertain delay to proceed with the design for the most restrictive case of maximum expected time lag, thus sacrificing the dynamics [7, 32–34]. In the subsequent part of this section, we present an alternative method for eliminating the adverse effects of DT in systems with unknown and/or time-varying delay. The method retains the conceptual simplicity of the traditional DTCs, and it allows us to preserve good system dynamical properties. We will show that the proposed compensation mechanism does not require throttling the controller gain to dampen the oscillations of the generated transmission rate. As a result, the primary drawback of the robust control scheme presented in the previous section, (5.161) and (5.162), is eliminated.

Similarly as in the case of the traditional DTCs, the compensator proposed in this work allows for carrying out the controller design as if there was no delay in the feedback loop and the controller acted immediately on the plant. The idea presented here is based on the observation that the actual performance limitation of the traditional DTCs in the systems with unknown and/or variable delay is the mismatch between the true input exerted on the plant and the one used by the predictor. One way to solve this problem proposed in the literature [5, 28, 37], is to apply the control signal issued for the remote plant and to adapt the estimate of the true delay to minimize the latency mismatch in the compensator structure. Unfortunately, the delay measurement in many systems poses severe difficulties, leads to errors, and may be impractical due to the required additional feedback mechanism, or even impossible. A different and promising approach that could be used for DT compensation in the systems with variable delay has been recently reported in [25] (and analyzed in detail in [26] and [27]). It treats the difference between the delayed and undelayed input as a network-induced disturbance and applies the concept of a communication disturbance observer to generate the state prediction. In effect, the observer extracts the true input exerted on the plant from the output measurements and uses the filtered version of the reconstructed delayed input in the DTC. However, this interesting approach suffers from several inconveniencies (also noted by the authors of [27]): it is in principle restricted to linear plants, additional errors may be introduced while passing the output through the inverse transfer function of the plant, and performance (or even stability) may suffer from filter nonidealities. Therefore, instead of the error-prone delay estimation, or input reconstruction from the output, in this work, we propose to use a direct input measurement to improve DT compensation mechanism in the systems with uncertain, time-varying delay. This is clearly a feasible solution since, unlike the delay, the input acting on the plant is usually available for measurement, and in certain cases, it can be even used directly by the compensator without the necessity of a sensor mechanism. In data transmission networks, for instance, the true delay between issuing the control signal and experiencing the new input rate at the network node is not known in advance by the controller situated at this node, but the number of incoming packets at the node is measurable and can be used in a direct way by the DTC. As another example, we may consider logistic applications, in which the latency in order procurement tends to undergo significant, *a priori* unknown fluctuations, but the input (the shipments arriving at a goods distribution center) is recorded online by the inventory management system, thus being readily available for generating new ordering decisions.

First, we will discuss the idea behind the proposed compensation mechanism and illustrate its properties in a few examples. Then, we apply the proposed compensator in the robust controller structure presented in Sect. 5.2.2. The properties of the developed control scheme are discussed and proved analytically. Finally, the controller performance is evaluated in a simulation scenario given in the last part of Sect. 5.2.3.

5.2.3.1 Compensation Mechanism

The traditional method of DT compensation by means of the SP is illustrated in Fig. 5.57, where $r(\cdot)$ is the reference signal, $u(\cdot)$ denotes the manipulated variable, and $y(\cdot)$ is the system output. If DT is known and constant, the delayed control action executed on a remote, stable plant described by transfer function $P(s)$ can

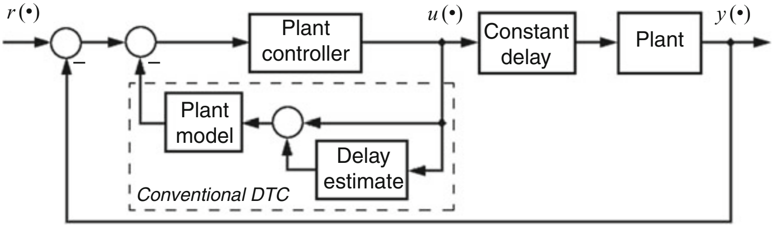


Fig. 5.57 Delay compensation by means of a conventional DTC

be simulated locally, and the predicted plant output can be used by the controller $K(s)$ as a local feedback. In this way, the latency is eliminated from the (outer) feedback loop, and the controller design can be carried out as if there was no delay between the controller and the plant. However, if the actual delay differs from the one used by the DTC (the nominal one), then the closed-loop stability is no longer ensured, and additional constraints on the controller dynamics need to be imposed [22]. It is typical in such situations to consider the worst-case scenario of the maximum possible delay and throttle the system dynamics accordingly, see, for instance, the controller design and stability analysis conducted in [10], or [32], for data transmission networks.

The new method of DT compensation based on delayed input measurement is shown in Fig. 5.58. Configuration I is used for linear plants where the DTC in the controller structure has already been designed and implemented for the case of a known, constant delay. In such circumstances, the proposed method provides the correction of the mismatch between the prediction of nominal system behavior (with constant delay) and the actual one (with time-varying delay). Configuration II is applied when there is no DTC in the plant controller structure, and/or the delay is unknown. In that case, the compensator predicts the plant output as if it was connected directly to the controller and waits for the delayed input measurement to complete the inner loop.

In addition to providing efficient controller operation in the systems with unknown, variable delay in terms of the tracking performance, the presented solution also eliminates the adverse effects of delay variability on the signal established by the controller. It is of great importance for systems where only a specified range of control signal is allowed, and in cases where oscillations originating from delay variations are not acceptable. For instance, in communication networks, a data source cannot inject packets into the network at a negative rate (or with infinite speed), so the signal issued by the controller must be nonnegative (and bounded). Therefore, it is desirable to obtain the signal generated by the controller in a system with variable delay as if this controller was operating in the system with perfectly known, constant latency. As shown in the case studies analyzed below, the alternative solution to DT compensation based on direct input measurements demonstrates such property.

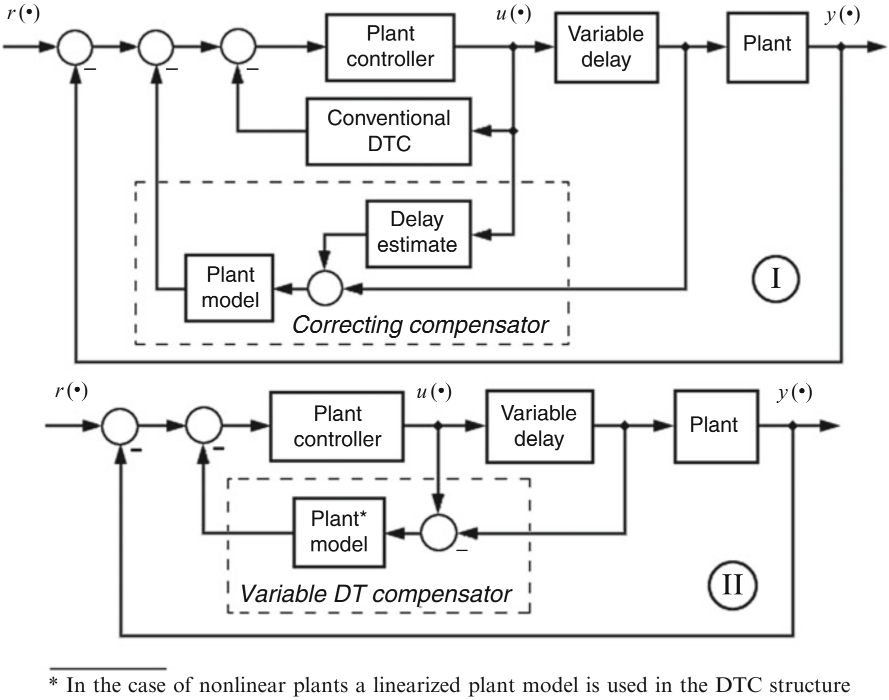


Fig. 5.58 Proposed method of DTC by input measurement: *I* correcting compensator, *II* variable DT compensator

5.2.3.2 Case Studies

In this section, we verify the efficiency of the proposed method of delay variability compensation in two case studies. For each scenario, we run three tests and display both the control signal and the plant output. To form a basis for comparison, test ① is executed for the system with constant delay and a controller employing the SP, or other method of delay compensation based on the nominal delay value. We refer to this setting as the ideal case. In test ②, we show how the system performance changes when delay varies in time in a way unpredictable to the controller. The controller in test ② uses the same DTC as in test ①. Finally, in test ③, we show how our DT compensation method improves the system performance in the presence of unknown, variable delay. The configuration for each test is illustrated in Fig. 5.59. Signal $\pi(\cdot)$ represents the disturbance acting on the input signal (in addition to the perturbation originating from delay variations), and $\tilde{u}_{\mathrm{R}}(\cdot)$ is the perturbed input. Other quantities are exactly the same as defined in Fig. 5.57; i.e., $u(\cdot)$ is the manipulated variable, and $y(\cdot)$ is the system output. Since we are primarily interested in studying the effects related to delay mismatches, we assume that the model accurately describes the controlled process. However, we include in our study

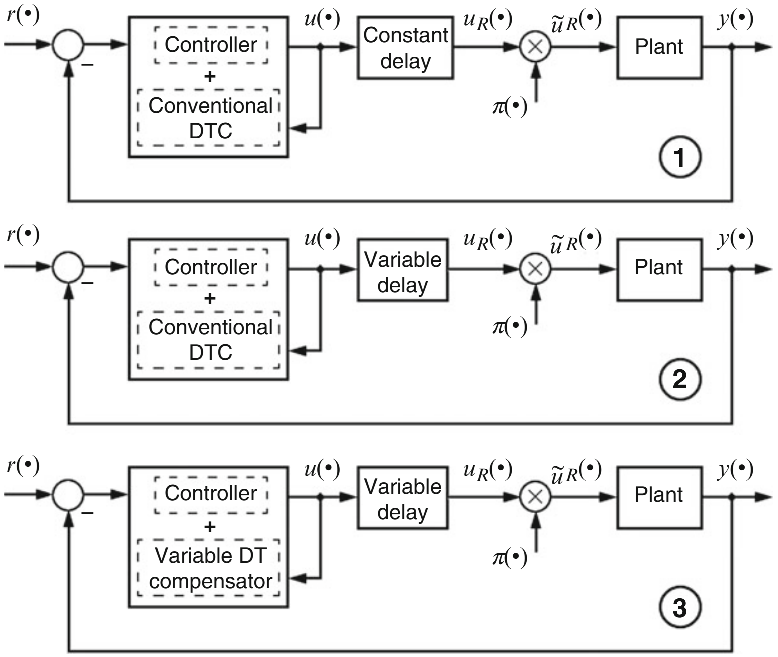


Fig. 5.59 Test configuration for case studies

the effects of external disturbance $\pi(\cdot)$ altering the control signal. The loop delay $\theta(\cdot)$ is assumed to be subject to the following type of variation:

$$\theta(t) = \theta_{\mathrm{n}}\,[1 + \theta_{\mathrm{a}} \sin\,(2\pi t\,/\theta_{\mathrm{n}})]\,, \tag{5.183}$$

where t is a continuous variable denoting time, θ_{n} is the nominal delay, and θ_{a} is a real number from the interval $(-1, 1)$ which specifies the amplitude of delay variations.

Example 5.5. In the first case study, we consider a linear third-order plant specified by the transfer function $P(s) = 1/s(s+1)(s+2)$. The remote plant is regulated by the proportional controller $K(s) = K_{\mathrm{p}}$ with the SP used for DT compensation. The purpose of the control action is to track the reference $r(t) = \mathbf{1}(t)$ (unit step). The external disturbance is absent; i.e., $\pi(\cdot) \equiv 0$.

In Fig. 5.60, we plot the signal generated by the controller for two values of the gain (a) $K_{\mathrm{p}} = 1$ and (b) $K_{\mathrm{p}} = 2$; and in Fig. 5.61, we show the system output. In the first test, the delay $\theta(t) = \theta_{\mathrm{n}} = 5$s, whereas in tests ② and ③, $\theta(t) = 5[1 + 0.6\sin(2\pi t/5)]$. We can see from the graphs how the performance of the SP degrades when the true delay differs from the nominal one and varies with time. More importantly, when the delay exhibits variations, the conventional DTC cannot prevent the system instability (for $K_{\mathrm{p}} = 2$). However, when the correction of the

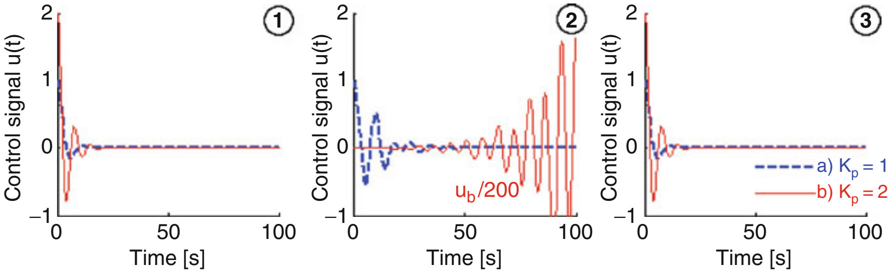


Fig. 5.60 Control signal for the plant $1/s(s+1)(s+2)$ with proportional control: *a* $K_p = 1$, *b* $K_p = 2$

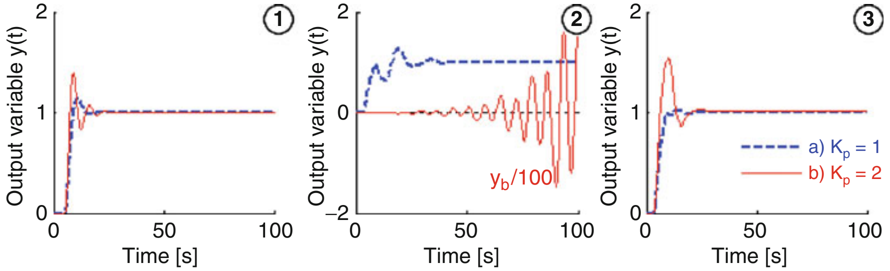


Fig. 5.61 Output variable of the plant $1/s(s+1)(s+2)$ with proportional control: *a* $K_p = 1$, *b* $K_p = 2$

input signal used by the predictor is applied ③ (configuration I from Fig. 5.58), the stability is maintained and the output follows the baseline ①. The minor differences between the two output signals $y_1(t)$ and $y_3(t)$ are due to the distortion of the delayed control signal caused by latency variations which turns out to have the largest impact on the plant behavior in the initial phase of the control process. Nevertheless, which is clear from the graphs in Fig. 5.60, the input signal generated by the controller employing the proposed DT compensation method in test ③ is nearly identical to the one in the system with constant delay and the conventional DTC ①. This is the primary reason for stability enhancement observed in the operation of the proposed method. The analysis of the root locus drawn for the system without delay in Fig. 5.62 indicates a crossing of the imaginary axis for $K_p = 6$. The evolution of the control signal sketched for $K_p = 6$ shows that the SP ensures bounded-input bounded-output (BIBO) stability for the system with constant delay ①, and our method does the same for the case of unknown, time-varying DT ③. Consequently, the proposed DT compensation technique eliminates stability threat caused by delay variations in the considered case of regulating a remote third-order plant.

Example 5.6. In the second case study, we direct our attention to the control of nonminimum phase systems, which tend to pose difficulties in providing efficient control [13, 14]. We consider the plant $P(s) = (1-s)/(s+1)(s+2)(s+3)$ regulated

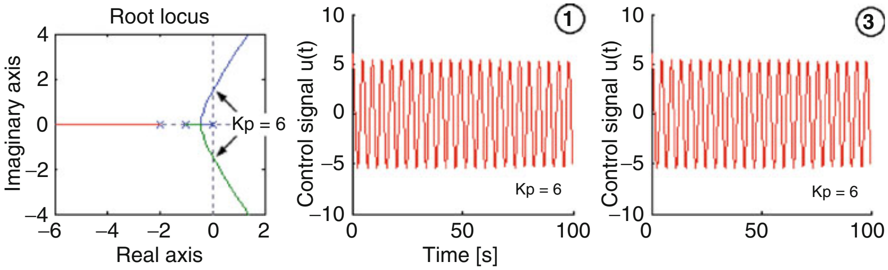


Fig. 5.62 Root locus for the plant $1/s(s + 1)(s + 2)$ and control signal for $K_p = 6$

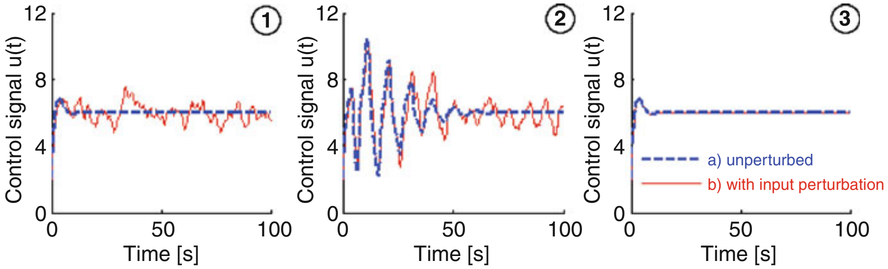


Fig. 5.63 Control signal for the plant $(1 - s)/(s + 1)(s + 2)(s + 3)$: *a* $\pi(t) \equiv 0$, *b* $\pi(t) \in [-0.5, 1.5]$

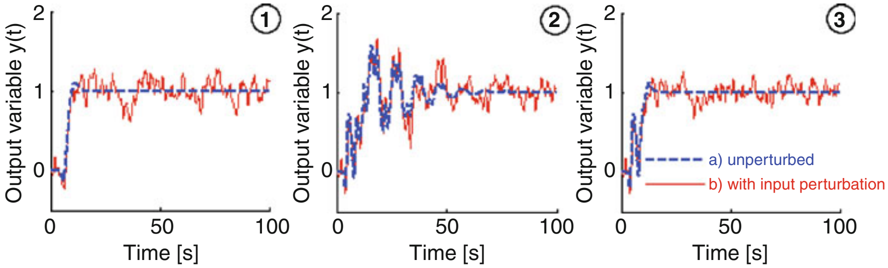


Fig. 5.64 Output of the plant $(1 - s)/(s + 1)(s + 2)(s + 3)$: *a* $\pi(t) \equiv 0$, *b* $\pi(t) \in [-0.5, 1.5]$

by the PI controller $K(s) = 2(s + 1)/s$. Similarly as in the previous scenario analyzed in Example 5.5, the SP is used to compensate for the effects of delay. We set the reference signal $r(t) = \mathbf{1}(t)$ and run two simulations: in the first one (a), it is assumed that no disturbance is present in the system, whereas in the second simulation (b), we introduce a multiplicative input perturbation $\pi(\cdot)$ being a uniformly distributed random number from the interval $[-0.5, 1.5]$. In Fig. 5.63, we plot the input signal generated by the controller, and in Fig. 5.64, we show the system output. In test ①, the delay is constant and equal to 5 s, whereas in tests ② and ③, $\theta(t) = 5[1 + 0.6\sin(2\pi t/5)]$.

It is evident from the input and output evolution in Figs. 5.63 and 5.64 that our method successfully combats the adverse effects caused by delay fluctuations. Moreover, unlike the traditional DT compensation technique, the proposed method reduces the plant sensitivity to the input signal perturbations $\pi(\cdot)$, which is a direct consequence of the obtained control signal regeneration (compare curves ①b and ③b in Figs. 5.63 and 5.64).

5.2.3.3 Proposed Control Strategy

The control strategy presented here extends the robust controller described in Sect. 5.2.2 by incorporating delay variability compensator shown in Fig. 5.58 (configuration I). The compensator, employing the measurements of the actual incoming packet rate, allows us to keep good controller dynamics without producing undesirable oscillations of the calculated transmission rate. This is in stark contrast to the earlier proposals [7, 32–34] which imposed a trade-off between good dynamics and system stability.

The transmission rate is generated by the controller according to the following equation:

$$u(kT) = \begin{cases} 0, & \text{if} \quad \omega_{\text{comp}}(kT) < 0, \\ \omega_{\text{comp}}(kT), & \text{if} \quad 0 \le \omega_{\text{comp}}(kT) \le u_{\max}, \\ u_{\max}, & \text{if} \quad \omega_{\text{comp}}(kT) > u_{\max}, \end{cases} \tag{5.184}$$

where $u_{\max} > d_{\max}$ and function $\omega_{\text{comp}}(\cdot)$ is defined as

$$\omega_{\text{comp}}(kT) = \gamma \left\{ y_{\text{D}} - y(kT) - \sum_{j=k-\bar{n}_p}^{k-1} u(jT) \right. \\ \left. + \varepsilon \sum_{j=0}^{k-1} \left[u_{\text{R}}(jT) - u\left(jT - \bar{n}_p T\right) \right] \right\}. \tag{5.185}$$

The structure of controller (5.184) and (5.185) is illustrated in Fig. 5.65. The structure consists of three elements:

1. The LQ optimal SM controller given by (5.67), which operates using the nominal delay in the feedback loop $\overline{\text{RTT}} = \bar{n}_p T$.
2. The delay variability compensator $\varepsilon \sum_{j=0}^{k-1} \left[u_{\text{R}}(jT) - u\left(jT - \bar{n}_p T\right) \right]$.
3. The saturation element described by (5.184).

The operation of the compensator for the considered integrating plant (packet buffer at the controlling node) may be interpreted as accumulating the information about differences between the number of packets that actually arrived at the node and those which were expected to arrive in each discrete-time interval. In software

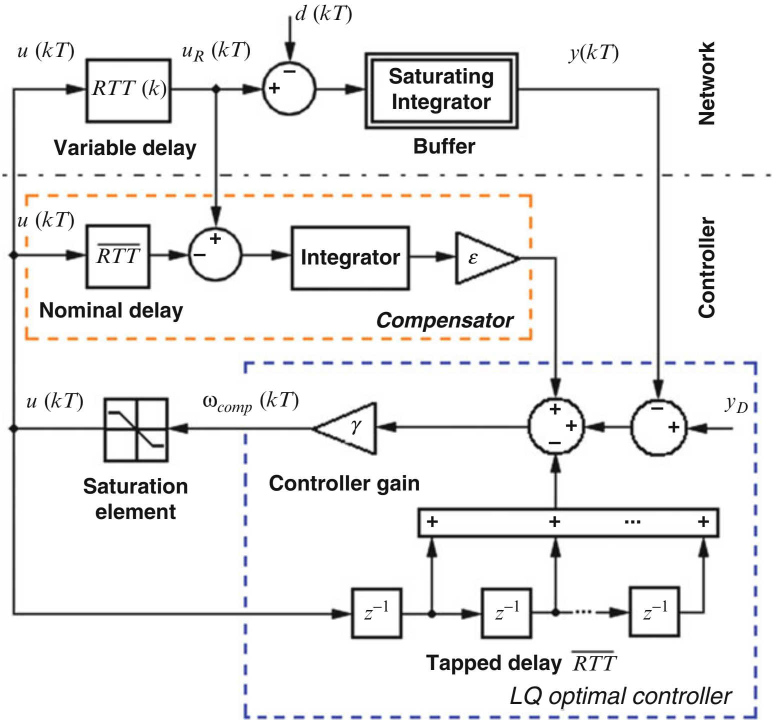


Fig. 5.65 Structure of controller (5.184) and (5.185)

realization, this information can be stored as a single value in the node memory. The influence of the compensation mechanism is adjusted through the tuning coefficient $\varepsilon \in [0, 1]$. Setting $\varepsilon = 1$ corresponds to the case of full compensation, and $\varepsilon = 0$ reflects the case of the compensation turned-off. Note that with the compensator absent ($\varepsilon = 0$), the proposed controller reduces to the robust scheme (5.161) and (5.162) analyzed in the previous section.

In the next section, we state the most important properties of the proposed control law (5.184) and (5.185), and we prove them analytically.

5.2.3.4 Properties of the Proposed Strategy

The properties of the designed strategy will be formulated as two theorems. The first one shows how to select the buffer capacity to always accommodate the entire packet queue and in this way eliminate losses. The second theorem, on the other hand, defines the minimum value of the demand queue length necessary to provide full bandwidth utilization irrespective of the pattern of delay and bandwidth variations.

Theorem 5.15. *If controller (5.184) with function $\omega_{\mathrm{comp}}(\cdot)$ defined by (5.185) is applied to system (5.158) and (5.159), then the queue length in the controlling node buffer is always upper-bounded; i.e., for any $k \geq 0$,*

$$y\,(kT) \leq y_{\max} = y_{\mathrm{D}} + u_{\max} + (1+\varepsilon)\,\xi_{\max}. \tag{5.186}$$

Proof. Using (5.151), (5.156), and the definition of function $\xi(\cdot)$, the term compensating the effects of delay variations in (5.185) can be reduced to the following form:

$$\begin{aligned}\sum_{j=0}^{k-1}\left[u_{\mathrm{R}}\,(jT) - u\left(jT - \bar{n}_p T\right)\right] &= \sum_{j=0}^{k-1}\left\{u\,[jT - RTT(j)] - u\left(jT - \bar{n}_p T\right)\right\} \\ &= \sum_{j=0}^{k-1}\xi\,(jT) = \xi\,(kT)\,.\end{aligned} \tag{5.187}$$

Hence, the correcting action of the compensator is to nullify the perturbation originating from delay variations, $\xi(kT)$, affecting the queue length $y(kT)$. In consequence, we may rewrite function $\omega_{\mathrm{comp}}(\cdot)$ as

$$\omega_{\mathrm{comp}}\,(kT) = \gamma\left[y_{\mathrm{D}} - y\,(kT) - \sum_{j=k-\bar{n}_p}^{k-1} u\,(jT) + \varepsilon\xi\,(kT)\right]. \tag{5.188}$$

It follows from the system initial conditions that the buffer at the controlling node is empty for any $k \leq (1-\beta)\bar{n}_p$. Consequently, it is sufficient to show that the proposition holds for all $k > (1-\beta)\bar{n}_p$. Let us consider some integer $l > (1-\beta)\bar{n}_p$ and the value of $\omega_{\mathrm{comp}}(\cdot)$ at instant lT. Two cases ought to be analyzed: the situation when $\omega_{\mathrm{comp}}(lT) \geq 0$ and the circumstances when $\omega_{\mathrm{comp}}(lT) < 0$.

Case 1. We address the case when $\omega_{\mathrm{comp}}(lT) \geq 0$. Directly from (5.188), we get

$$\omega_{\mathrm{comp}}\,(lT) = \gamma\left[y_{\mathrm{D}} - y\,(lT) - \sum_{j=l-\bar{n}_p}^{l-1} u\,(jT) + \varepsilon\xi\,(lT)\right] \geq 0, \tag{5.189}$$

which leads to

$$y\,(lT) \leq y_{\mathrm{D}} + \varepsilon\xi\,(lT) - \sum_{j=l-\bar{n}_p}^{l-1} u\,(jT). \tag{5.190}$$

It follows from the algorithm definition that $u(\cdot)$ is always nonnegative; hence,

$$y\,(lT) \leq y_{\mathrm{D}} + \varepsilon\xi\,(lT)\,. \tag{5.191}$$

Moreover, since $\xi(lT) \leq \xi_{\max}$, we obtain

$$y\left(lT\right) \leq y_{\mathrm{D}} + \varepsilon\xi_{\max} < y_{\max}, \tag{5.192}$$

which ends the first part of the proof.

Case 2. In the second part of the proof, we analyze the situation when $\omega_{\mathrm{comp}}(lT) < 0$. First, we find the last instant $l_1 T < lT$ when $\omega_{\mathrm{comp}}(\cdot)$ was nonnegative. According to (5.188), $\omega_{\mathrm{comp}}(0) = \gamma y_{\mathrm{D}} > 0$, so the moment $l_1 T$ indeed exists and the value of $y(l_1 T)$ satisfies an inequality similar to (5.190); i.e.,

$$y\left(l_1 T\right) \leq y_{\mathrm{D}} + \varepsilon\xi\left(l_1 T\right) - \sum_{j=l_1-\bar{n}_p}^{l_1-1} u\left(jT\right). \tag{5.193}$$

The queue length at instant lT can be expressed relative to $y(l_1 T)$ as follows:

$$y\left(lT\right) = y\left(l_1 T\right) + \sum_{j=l_1-\bar{n}_p}^{l-\bar{n}_p-1} u\left(jT\right) + \xi\left(lT\right) - \sum_{j=l_1}^{l-1} h\left(jT\right), \tag{5.194}$$

which after applying (5.193) leads to

$$\begin{aligned} y(lT) &\leq y_{\mathrm{D}} + \varepsilon\xi\left(l_1 T\right) - \sum_{j=l_1-\bar{n}_p}^{l_1-1} u\left(jT\right) + \sum_{j=l_1-\bar{n}_p}^{l-\bar{n}_p-1} u\left(jT\right) + \xi\left(lT\right) - \sum_{j=l_1}^{l-1} h(jT) \\ &\leq y_{\mathrm{D}} + \varepsilon\xi\left(l_1 T\right) + \xi\left(lT\right) + \sum_{j=l_1}^{l-\bar{n}_p-1} u\left(jT\right) - \sum_{j=l_1}^{l-1} h\left(jT\right). \end{aligned} \tag{5.195}$$

The controller sets a nonzero transmission rate for the last time before lT at instant $l_1 T$, and this value could be as large as $u_{\max}$. Consequently, the sum $\sum_{j=l_1}^{l-\bar{n}_p-1} u\left(jT\right) = u\left(l_1 T\right) \leq u_{\max}$. Since the utilized bandwidth is always nonnegative, then using the condition $\xi(lT) \leq \xi_{\max}$ we obtain from (5.195) the following estimate of the queue length at instant lT:

$$\begin{aligned} y\left(lT\right) &\leq y_{\mathrm{D}} + \varepsilon\xi\left(l_1 T\right) + \xi\left(lT\right) + u\left(l_1 T\right) - \sum_{j=l_1}^{l-1} h\left(jT\right) \\ &\leq y_{\mathrm{D}} + \varepsilon\xi\left(l_1 T\right) + \xi\left(lT\right) + u\left(l_1 T\right) \leq y_{\mathrm{D}} + \left(1+\varepsilon\right)\xi_{\max} + u_{\max} = y_{\max}. \end{aligned} \tag{5.196}$$

This concludes the second part of the reasoning and completes the proof. □

Theorem 5.17 states that the packet queue length is finite and never exceeds the level of $y_{\max}$. This means that irrespective of bandwidth and delay variations, the system output $y(\cdot)$ is bounded and the risk of losses is eliminated. The second theorem, formulated below, shows that with the appropriately selected demand queue length y_{D}, we can make the queue length always strictly positive, which guarantees full bandwidth utilization.

Theorem 5.18. *If controller (5.184) with function $\omega_{\mathrm{comp}}(\cdot)$ defined by (5.185) is applied to system (5.158) and (5.159) , and the demand queue length satisfies the following inequality:*

$$y_{\mathrm{D}} > u_{\max}\left(\bar{n}_p + 1/\gamma\right) + d_{\max} + (1+\varepsilon)\,\xi_{\max}, \tag{5.197}$$

then for any $k \geq (1+\beta)\bar{n}_p + n_{\max}$, where $n_{\max} = y_{\max}/(u_{\max} - d_{\max})$, the queue length is strictly positive.

Proof. The theorem assumption implies that we deal with time instants $kT \geq (1+\beta)\bar{n}_p T + n_{\max} T$. Taking some integer $l \geq (1+\beta)\bar{n}_p + n_{\max}$ and the value of signal $\omega_{\mathrm{comp}}(\cdot)$ at instant lT, we need to consider two cases: the situation when $\omega_{\mathrm{comp}}(lT) < u_{\max}$ and the circumstances when $\omega_{\mathrm{comp}}(lT) \geq u_{\max}$.

Case 1. First, we analyze the situation when $\omega_{\mathrm{comp}}(lT) < u_{\max}$. From (5.188), we get

$$\omega_{\mathrm{comp}}\left(lT\right) = \gamma\left[y_{\mathrm{D}} - y\left(lT\right) - \sum_{j=l-\bar{n}_p}^{l-1} u\left(jT\right) + \varepsilon\xi\left(lT\right)\right] < u_{\max}, \tag{5.198}$$

which after the term rearrangement results in

$$y\left(lT\right) > y_{\mathrm{D}} - u_{\max}/\gamma - \sum_{j=l-\bar{n}_p}^{l-1} u\left(jT\right) + \varepsilon\xi\left(lT\right). \tag{5.199}$$

The transmission rate is always bounded by $u_{\max}$, which implies

$$y\left(lT\right) > y_{\mathrm{D}} - u_{\max}/\gamma - u_{\max}\bar{n}_p + \varepsilon\xi\left(lT\right). \tag{5.200}$$

Since $\xi(\cdot) \geq -\xi_{\max}$, we get

$$y\left(lT\right) > y_{\mathrm{D}} - u_{\max}/\gamma - u_{\max}\bar{n}_p - \varepsilon\xi_{\max} = y_{\mathrm{D}} - u_{\max}\left(\bar{n}_p + 1/\gamma\right) - \varepsilon\xi_{\max}. \tag{5.201}$$

Using assumption (5.197), we get $y(lT) > 0$, which concludes the first part of the proof.

Case 2. In the second part of the proof, we investigate the situation when $\omega_{\text{comp}}(lT) \geq u_{\max}$. First, we find the last moment $l_1 T < lT$ when function $\omega_{\text{comp}}(\cdot)$ was smaller than $u_{\max}$. It comes from Theorem 5.17 that the queue length never exceeds the value of $y_{\max}$. Furthermore, from (5.2), we know that the consumed bandwidth is limited by $d_{\max}$. Thus, the maximum interval $n_{\max}T$ during which the controller may continuously generate the maximum transmission rate for the source is determined as $n_{\max}T = Ty_{\max}/(u_{\max} - d_{\max})$, and instant $l_1 T$ does exist. Moreover, from the theorem assumptions, we get $l_1 T \geq (1+\beta)\bar{n}_p T$, which means that by the time $l_1 T$, the first packets reach the node, no matter the delay variation.

The value of $\omega_{\text{comp}}(l_1 T) < u_{\max}$. Thus, following a similar reasoning as presented in (5.198)–(5.201), we arrive at $y(l_1 T) > 0$ and

$$y\left(lT\right) > y_{\text{D}} - \frac{u_{\max}}{\gamma} - \sum_{j=l_1-\bar{n}_p}^{l_1-1} u\left(jT\right) + \varepsilon\xi\left(l_1 T\right) + \sum_{j=l_1-\bar{n}_p}^{l-\bar{n}_p-1} u\left(jT\right)$$
$$+\,\xi\left(lT\right) - \sum_{j=l_1}^{l-1} h\left(jT\right). \tag{5.202}$$

After performing algebraic manipulations on the first two sums in (5.176), we get

$$y\left(lT\right) > y_{\text{D}} - \frac{u_{\max}}{\gamma} + u\left(l_1 T\right) + \sum_{j=l_1+1}^{l-1} u\left(jT\right) - \sum_{j=l-\bar{n}_p}^{l-1} u\left(jT\right)$$
$$-\sum_{j=l_1}^{l-1} h\left(jT\right) + \varepsilon\xi\left(l_1 T\right) + \xi\left(lT\right). \tag{5.203}$$

Recall that $l_1 T$ was the last instant before lT when the controller calculated rate smaller than $u_{\max}$. This rate could be as low as zero. Afterwards, the algorithm generates the maximum rate value, and the first sum in (5.177) reduces to $u_{\max}(l - 1 - l_1)$. Since for any k, $u(kT) \leq u_{\max}$, the second sum is upper-bounded by $u_{\max} = \bar{n}_p$. Thus, we have

$$y\left(lT\right) > y_{\text{D}} - \frac{u_{\max}}{\gamma} + u_{\max}\left(l-1-l_1\right) - u_{\max}\bar{n}_p + \varepsilon\xi\left(l_1 T\right) + \xi\left(lT\right) - \sum_{j=l_1}^{l-1} h\left(jT\right). \tag{5.204}$$

Using the fact that $\xi(\cdot) \geq -\xi_{\max}$ and $h(\cdot) \leq d_{\max}$, we get the following estimate of the queue length $y(lT)$:

$$y\left(lT\right) > y_{\text{D}} - u_{\max}/\gamma + u_{\max}\left(l-1-l_1\right) - u_{\max}\bar{n}_p - \varepsilon\xi_{\max} - \xi_{\max} - d_{\max}\left(l-l_1\right). \tag{5.205}$$

Table 5.7 Controller parameters

Controller gain γ	Demand queue length y_{D} [packets]	Buffer size $y_{\max}$ [packets]
0.259	$230 > 228$	273
0.618	$205 > 204$	238
1	$200 > 197$	233

Using the theorem assumption (5.197), we obtain

$$\begin{aligned} y\,(lT) &> u_{\max}\left(\bar{n}_p + 1/\gamma\right) + d_{\max} + (1+\varepsilon)\,\xi_{\max} \\ &\quad - u_{\max}/\gamma + u_{\max}\,(l-1-l_1) - u_{\max}\bar{n}_p - (1+\varepsilon)\,\xi_{\max} - d_{\max}\,(l-l_1) \\ &= u_{\max}\,(l-1-l_1) - d_{\max}\,(l-1-l_1)\,. \end{aligned} \tag{5.206}$$

Finally, since $l > l_1$ and $u_{\max} > d_{\max}$, we arrive at $y(lT) > 0$. This conclusion ends the proof. □

In the next section, we present the results of simulations demonstrating the properties of the proposed compensation mechanism implemented in controller (5.184) and (5.185), improving the quality of data flow control in the network with time-varying delay.

5.2.3.5 Simulation Results

The simulations are run for the connection characterized by the nominal delay $\overline{\mathrm{RTT}} = 10T = 100$ ms. Similarly as in Sect. 5.2.2.3, the actual delay in the feedback loop varies in a way unknown to the controller in the range specified by $\beta = 0.3$. The performance of controller (5.184) and (5.185) is verified for different dynamical settings specified by the gain chosen as either 0.259, 0.618, or 0.916. In each test, the maximum transmission rate is set as $u_{\max} = 11$ packets. We assume that full compensation of delay variations is applied; i.e., $\varepsilon = 1$ is set in (5.185). Three series of simulation tests are run: two for the bandwidth illustrated in Fig. 5.7 and the third one for the stochastic pattern depicted in Fig. 5.12. In each case, the maximum bandwidth $d_{\max} = 10$ packets.

Test 1. In the first simulation scenario, we verify the controller performance in response to the available bandwidth depicted in Fig. 5.7. The delay in the feedback loop, illustrated in Fig. 5.49, is assumed to vary according to (5.182). We run a few simulations for different gain settings of controller (5.184) and (5.185). In order to obtain full bandwidth utilization, the demand queue length is selected in each test according to (5.197), and the buffer capacity is adjusted according to (5.186) so that loss-free transmission is ensured. Parameters used in the tests are summarized in Table 5.7.

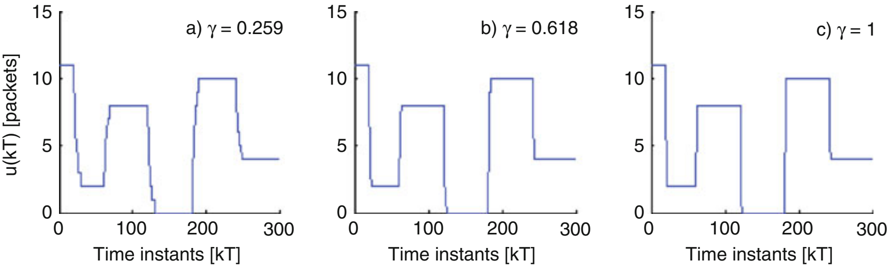


Fig. 5.66 Generated transmission rate: (**a**) $\gamma = 0.259$, (**b**) $\gamma = 0.618$, (**c**) $\gamma = 1$

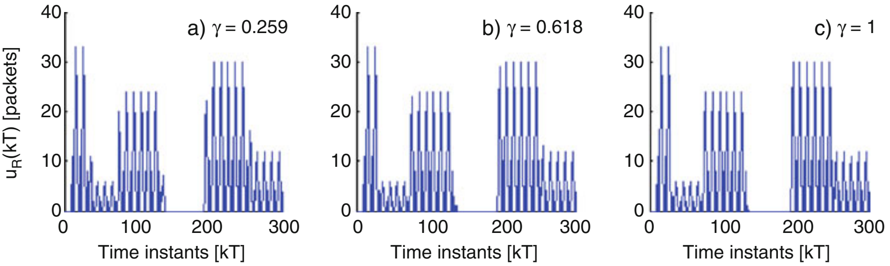


Fig. 5.67 Incoming rate: (**a**) $\gamma = 0.259$, (**b**) $\gamma = 0.618$, (**c**) $\gamma = 1$

The test results are illustrated in Figs. 5.66–5.68: the transmission rate established by the proposed controller in Fig. 5.66, the incoming stream of packets in Fig. 5.67, and the buffer queue length in Fig. 5.68. We can see from the graphs that controller (5.184) and (5.185) quickly responds to the abrupt changes of the available bandwidth, yet without undesirable oscillations disrupting the performance of robust controller (5.161) and (5.162), analyzed in the previous section for the identical network model (see Fig. 5.50). In fact, if we compare the transmission rate generated by controller (5.184) and (5.185) with the one established by the LQ optimal SM controller operating in the network with constant delay depicted in Fig. 5.8, we can notice that the curves are very similar, differing primarily in the initial phase. This clearly demonstrates the efficiency of the proposed compensation mechanism, which eliminates the adverse effects of delay variations from the rate signal established by the controller. We may say that the variable DT compensator regenerates the input signal in network with unknown, time-varying delay so that the established transmission rate evolves as if RTT was fixed.

We can see from Fig. 5.68 that the queue length does not increase beyond the buffer capacity specified in Table 5.7 and it does not fall to zero after the initial phase. In consequence, packet losses are eliminated and the maximum throughput is guaranteed. If we compare the buffer occupancy shown in Fig. 5.52, resulting from the application of controller (5.161) and (5.162), with the one obtained by

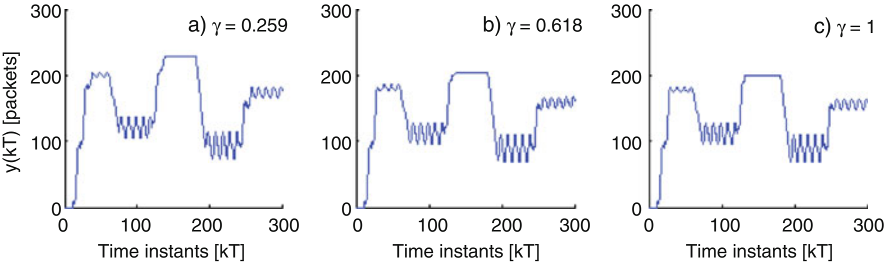


Fig. 5.68 Queue length: (**a**) $\gamma = 0.259$, (**b**) $\gamma = 0.618$, (**c**) $\gamma = 1$

using controller (5.184) and (5.185) depicted in Fig. 5.68, we can see a similar degree of oscillations which are caused by the disrupted incoming rate. The plots in Figs. 5.52 and 5.68 also show that the price to pay for a smoother transmission rate is the increased buffer capacity required to accommodate extra packets allowed for transmission by the compensation mechanism. However, from the implementation point of view, this trade-off is in favor of the compensation mechanism since with a smoother transmission rate there is a bigger chance that the source will respond correctly to the controller command.

Test 2. In the second simulation scenario, we test the controller robustness with respect to an external, multiplicative perturbation acting in the input channel, $\pi(\cdot)$, which alters the incoming packet rate in addition to delay variations. We assume that the perturbation follows the normal distribution with mean equal to 1 and variance 0.05, which results in the values of $\pi(\cdot) \in [0.4, 1.6]$. When $\pi(kT) > 1$, we have the situation of a nonconforming source (which transmits data at a rate greater than the assigned one), and when $\pi(kT) < 1$, we deal with a saturated transmitter (experiencing temporary data shortage), or losses on the forward path. We verify the performance of controller (5.184) and (5.185) for the case of the compensation turned-off (a), and with full compensation applied (b). The controller parameters are set in the following way: the gain $\gamma = 0.618$ (the golden-ratio controller), the maximum rate $u_{\max} = 11$ packets, and the demand queue length (adjusted to the value from Table 5.7 for the case of $\gamma = 0.618$) $y_D = 205$ packets. We run two series of simulations: in the first test, we assume that RTT is constant and equal to the value known to the controller, whereas in the second test, the delay varies according to $\mathrm{RTT}(k) = \lfloor 10 \cdot [1 + 0.3\sin(0.628k)]\rfloor$. The rate generated by the controller is shown in Fig. 5.69, the perturbed incoming rate in Fig. 5.70, and the packet queue length in Fig. 5.71.

As one can see from the graphs in Fig. 5.69, the proposed DT compensator not only mitigates the effects of delay variations but also eliminates the influence of extra perturbation of the incoming packet stream from affecting the generated transmission rate. The rate established by the controller in the fully compensated case is smooth and nonoscillatory. Moreover, although the compensator cannot prevent the external perturbation from altering the input signal (obviously, it cannot

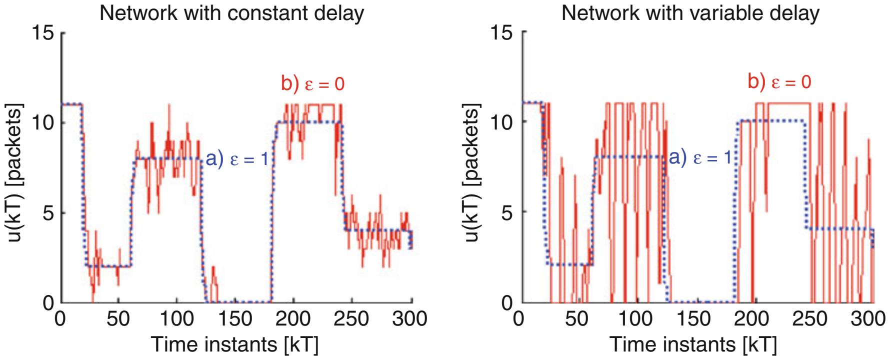


Fig. 5.69 Generated transmission rate: *a* without compensation, $\varepsilon = 0$, *b* with full compensation applied, $\varepsilon = 1$

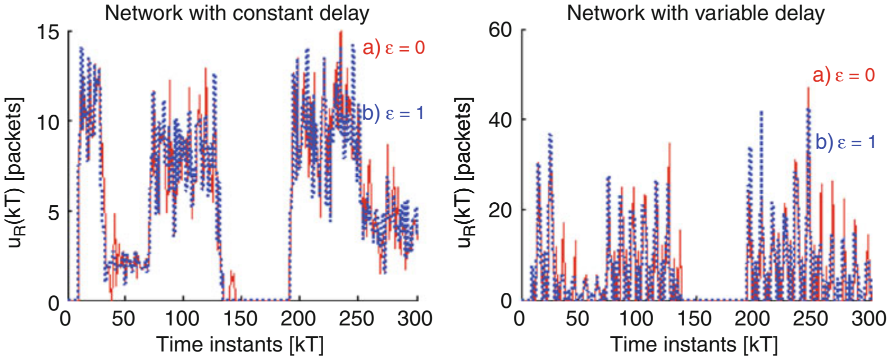


Fig. 5.70 Incoming rate: *a* without compensation, *b* with full compensation applied

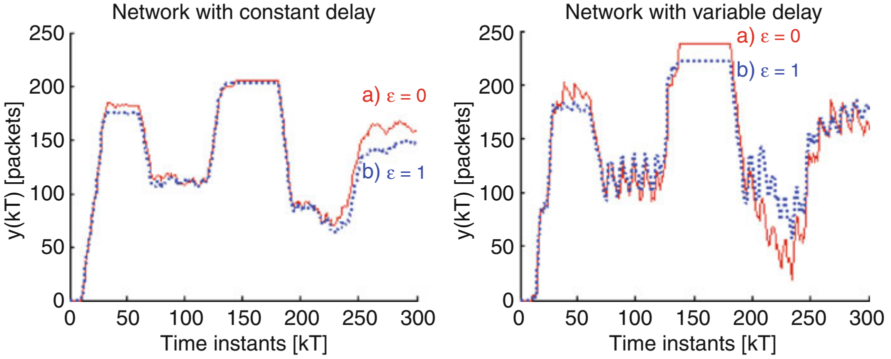


Fig. 5.71 Queue length: *a* without compensation, *b* with full compensation applied

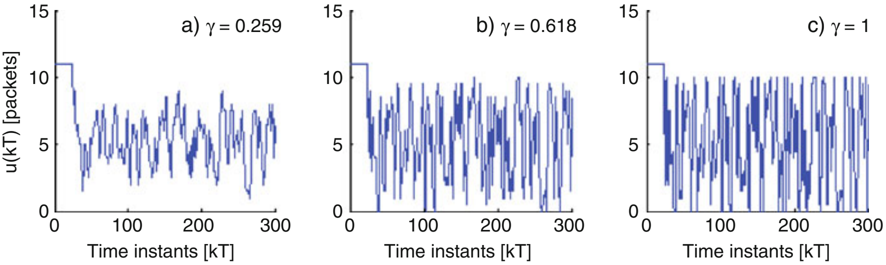


Fig. 5.72 Generated transmission rate: (**a**) $\gamma = 0.259$, (**b**) $\gamma = 0.618$, (**c**) $\gamma = 1$

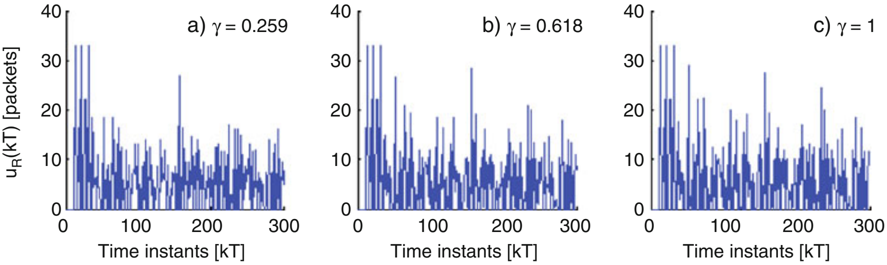


Fig. 5.73 Incoming rate: (**a**) $\gamma = 0.259$, (**b**) $\gamma = 0.618$, (**c**) $\gamma = 1$

anticipate neither the exact form of source nonideality nor packet losses occurring at some other node in the network), it improves the system robustness. It can be seen from Figs. 5.70 and 5.71 that with comparable input perturbation, the queue length in the case of the compensation applied is less oscillatory and always remains within the assigned buffer space. This is not the case of the uncompensated system, in which the packet losses occur due to buffer overflow in the network with variable delay in the interval $[137T, 181T]$.

Test 3. In the third simulation scenario, we test the controller performance in a stochastic environment with the bandwidth following the normal distribution with mean $d_\mu = 5$ packets and standard deviation $d_\delta = 5$ packets, illustrated in Fig. 5.12, and delay subject to random, normally distributed changes, $D_{\text{norm}}(10T, 1.41T)$. In the tests, we assume full delay variability compensation and set $\varepsilon = 1$. The remaining controller parameters are adjusted as in Test 1. They are listed in Table 5.7.

The results of the simulations are illustrated in Figs. 5.72–5.74: the transmission rate established by the algorithms in Fig. 5.72, the incoming packet rate in Fig. 5.73, and the buffer occupancy in Fig. 5.74. The analysis of the plots from Fig. 5.72 shows that by decreasing the gain, one can reduce the oscillations of the generated transmission rate caused by fluctuations of the available bandwidth. Moreover, the queue length stays within the assigned buffer space in the whole simulation interval,

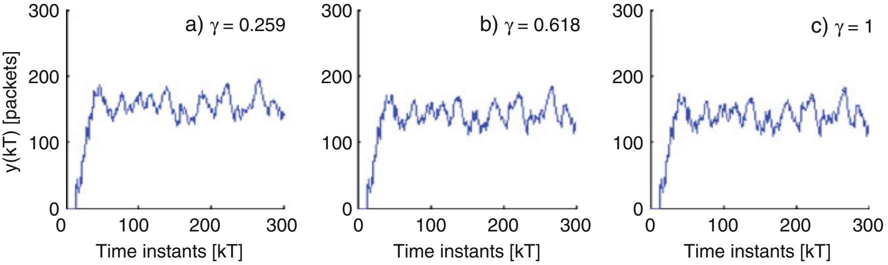


Fig. 5.74 Queue length: (**a**) $\gamma = 0.259$, (**b**) $\gamma = 0.618$, (**c**) $\gamma = 1$

which means that packet losses are eliminated. Despite the presence of both the bandwidth and delay variations, the queue length does not drop to zero, which means that all of the available bandwidth is efficiently used for data transfer.

5.3 Chapter Summary

In this chapter, we studied the flow control process in a single connection of a data transmission network in which the feedback information about the current transmission rate is provided at discrete-time instants only.

First, in Sect. 5.1, we analyzed phenomena related to handling the flow of data in the network with constant delay in the feedback loop. We presented a number of control strategies, each based on sound theoretical foundations of a robust control method – SMC. Since in SMC a crucial design step is selection of an appropriate sliding plane, we focused on choosing the plane parameters and its dynamics. First, an optimization problem with quadratic performance index was stated and solved analytically. The analytical solution of the optimization procedure allowed us to represent the control law in a closed form, which can be intuitively interpreted as adjusting the packet transmission rate in proportion to the free buffer space minus the packets from the assignments performed in the last RTT. A simple form of the obtained control law guarantees its straightforward implementation at the software and hardware level. It also ensures operational efficiency (as the rate calculation involves only a few additions and one multiplication). A possible drawback of the proposed strategy is that in order to quickly bring the communication system into the region of maximum throughput, it requires large input signal at the beginning of the control process. In order to conform to the transmitter limitations and at the same time maintain fast system dynamics, we proposed a number of solutions for the network with transmission rate constraints. Three nonlinear controllers were designed. The first one uses a time-varying sliding plane, providing an implicit limitation on the value of the generated control signal and explicit control over the duration of the initial phase. The second algorithm is based on the reaching-law approach. It gives direct control over the maximum allowed transmission rate and indirect control over the duration of the initial phase. Finally, the third controller employs a saturation element to clamp any control signal exceeding the explicit maximum, giving indirect control over the time of the initial phase. The third control law shows the most convenient form for the algorithmic, software or hardware, implementation. All the designed controllers have been demonstrated to ensure the communication system stability in the presence of unpredictable variations of the available bandwidth. It has also been shown that with appropriately chosen buffer capacity, the controllers guarantee that the bottleneck node is not overflowed with packets, yet at the same time all of the available bandwidth is used to transmit the users' data. In this way, the maximum throughput in the network is achieved.

In Sect. 5.2, we analyzed a more complex network model in which the delay in the feedback loop is no longer assumed to be constant. We considered the situation of arbitrary (bounded) delay fluctuations, both in the feedback and data channels, which are *unknown* to the controller. Two control laws were proposed. They combine the advantageous smoothness of rate transitions offered by the SM controller with LQ optimal sliding plane, and the robustness features provided by the saturation element limiting the transmission rate to the feasible interval $[0, u_{\max} > 0]$. In order to further enhance the controller robustness to delay variations (and other disturbances affecting the incoming rate, such as unanticipated packet losses on the forward path), the second controller employs a delay variability compensator. The compensator utilizes the measurement of packet incoming rate to prevent perturbations in the input channel from inciting oscillations in the control signal. As a result, the transmission rate established by the controller is smooth and evolves as if the controller operated in the network with fixed and perfectly known delay. In consequence, the data source is assigned rate in a form which is easier to reproduce. For both controllers, there was specified a set of conditions which guarantee that despite unpredictable bandwidth and delay variations, the buffer capacity at the controlling node is not exceeded and all of the available bandwidth is used for data transfer. As a result, the maximum throughput in the communication system is obtained.

References

1. Anderson BDO, Moore JB (1990) Optimal control: linear quadratic methods. Prentice-Hall, New York
2. Arnold WF, Laub AJ (1984) Generalized eigenproblem algorithms and software for algebraic Riccati equations. Proc IEEE 72:1746–1754
3. Bartoszewicz A (1996) Remarks on 'Discrete-time variable structure control systems'. IEEE Trans Ind Electron 43:235–238
4. Bartoszewicz A (1998) Discrete time quasi-sliding mode control strategies. IEEE Trans Ind Electron 45:633–637
5. Bauer P, Sichitiu M, Ernst R, Premaratne K (2001) A new class of Smith predictors for network congestion control. In: Proceedings of IEEE international conference on electronics, Circuits and Systems, vol 2, Malta, pp 685–688

6. Benner P, Byers R (1998) An exact line search method for solving generalized continuous-time algebraic Riccati equations. IEEE Trans Autom Control 43:101–107
7. Blanchini F, Lo Cigno R, Tempo R (2002) Robust rate control for integrated services packet networks. IEEE/ACM Trans Netw 10:644–652
8. Bohacek S, Hespanha JP, Lee J, Lim C, Obraczka K (2006) A new TCP for persistent packet reordering. IEEE/ACM Trans Netw 14:369–382
9. Chu D, Lin WW, Tan CER (2006) A numerical method for a generalized algebraic Riccati equation. SIAM J Control Optim 45:1222–1250
10. De Cicco L, Mascolo S, Niculescu SI (2009) Robust stability analysis of a class of Smith predictor-based congestion control algorithms for computer networks. In: Proceedings of the 8th workshop on time-delay system, Sinaia, Romania
11. Edwards C, Spurgeon SK (1998) Sliding mode control: theory and applications. Taylor & Francis, London
12. Gao W, Wang Y, Homaifa A (1995) Discrete-time variable structure control systems. IEEE Trans Ind Electron 42:117–122
13. García P, Albertos P, Hägglund T (2006) Control of unstable non-minimum-phase delayed systems. J Process Control 16:1099–1111
14. Gessing R (2008) Parallel compensator versus Smith predictor for control of the plants with delay. Bull Pol Acad Sci – Tech Sci 56:339–345
15. Golo G, Milosavljević Č (2000) Robust discrete-time chattering free sliding mode control. Syst Control Lett 41:19–28
16. Guo CH, Laub AJ (2000) On a Newton-like method for solving algebraic Riccati equations. SIAM J Matrix Anal Appl 21:694–698
17. Higham NJ, Kim HM (2001) Solving a quadratic matrix equation by Newton’s method with exact line searches. SIAM J Matrix Anal Appl 23:303–313
18. Johnson MA, Grimble MJ (1987) Recent trends in linear optimal quadratic multivariable control system design. IEE Rev 134:53–71
19. Kaczorek T (2002) Positive 1D and 2D Systems. Springer, London
20. Lewis FL, Syrmos VL (1995) Optimal control, 2nd edn. Wiley, New York
21. Long JH, Hu XY, Zhang L (2008) Improved Newton’s method with exact line searches to solve quadratic matrix equation. J Comput Appl Math 222:645–654
22. Michiels W, Niculescu S (2007) Stability and stabilization of time-delay systems: an eigenvalue-based approach. SIAM, Philadelphia
23. Milosavljević Č, Peruničić-Draženović B, Veselić B, Mitić D (2006) Sampled data quasi-sliding mode control strategies. In: Proceedings of the IEEE international conference of industrial technology, pp 2640–2645
24. Mirkin L, Raskin N (2003) Every stabilizing dead-time controller has an observer-predictor-based structure. Automatica 39:1747–1754
25. Natori K, Ohnishi K (2006) A design method of communication disturbance observer for time delay compensation. In: Proceedings of the annual conference IEEE Industrial Electronics Society, Paris, France, pp 730–735
26. Natori K, Ohnishi K (2008) A design method of communication disturbance observer for time-delay compensation, taking the dynamic property of network disturbance into account. IEEE Trans Ind Electron 55:2152–2168
27. Natori K, Oboe R, Ohnishi K (2008) Stability analysis and practical design procedure of time delayed control systems with communication disturbance observer. IEEE Trans Ind Inform 4:185–197
28. Nortcliffe A, Love J (2004) Varying time delay Smith predictor process controller. ISA Trans 43:61–71
29. Paxson V (1999) End-to-end Internet packet dynamics. IEEE/ACM Trans Netw 7:277–292
30. Ogata K (1995) Discrete-time control systems, 2nd edn. Prentice Hall, Upper Saddle River
31. Palmor ZJ (1996) Time-delay compensation – Smith predictor and its modifications. In: Levine WS (ed) The control handbook. CRC Press, Boca Raton

32. Pietrabissa A, Delli Priscoli F, Fiaschetti A, Di Paolo F (2006) A robust adaptive congestion control for communication networks with time-varying delays. In: Proceedings of 2006 IEEE international conference on control applications, Munich, Germany, pp 2093–2098
33. Quet PF, Ataşlar B, İftar A, Özbay H, Kalyanaraman S, Kang T (2002) Rate-based flow controllers for communication networks in the presence of uncertain time-varying multiple time-delays. Automatica 38:917–928
34. Ren T, Gao Z, Kong W, Jing Y, Yang M, Dimirovski GM (2008) Performance and robustness analysis of a fuzzy immune flow controller in ATM networks with time-varying multiple time delays. J Control Theory Appl 6:253–258
35. Sichitiu ML, Bauer PH, Premaratne K (2003) The effect of uncertain time-variant delays in ATM networks with explicit rate feedback: a control theoretic approach. IEEE/ACM Trans Netw 11:628–637
36. Smith OJM (1959) A controller to overcome dead time. ISA J 6:28–33
37. Vatanski N, Georges JP, Aubrun Ch, Rondeau E, Jämsä-Jounela SL (2009) Networked control with delay measurement and estimation. Control Eng Pract 17:231–244
38. Zhong QC (2006) Robust control of time-delay systems. Springer, London

Chapter 6
Flow Control in a Multisource Discrete-Time System

In Chap. 5, we analyzed the basic networking phenomena related to controlling the flow of data in the network in which the feedback information about the network state is accessible for input rate adaptation at discrete-time instants only. We considered the problem of regulating the transmission rate in a single connection. In this chapter, we study a more complex setting in which the controller placed at the bottleneck node regulates the flow of data in multiple connections originating at various sources. It is assumed that the controlled connections can be characterized by different round-trip times, which is a typical situation in real networks. Consequently, the model developed in Sect. 5.1 for the case of a single flow needs to be extended to cover the case of simultaneous transmission rate control in multiple connections.

The model for the multisource setting considered in this chapter is introduced in Sect. 6.1. Then, we proceed with the controller design for different networking scenarios. Similarly as in Chap. 5, the robust control method – SMC – is applied, and we concentrate on the selection of an appropriate sliding plane to meet various design objectives. We begin with the plane selection through dynamical optimization employing a quadratic performance index. Since in the multisource network the optimal solution may generate negative data flow rates, which is clearly inappropriate as it would require retrieving from the network the data already sent by the sources, a modified suboptimal controller, always generating nonnegative rates, is designed. Similarly as in Sect. 5.1.2, the key point in the optimization procedure is the analytical solution of matrix Riccati equation for an nth-order discrete-time system. Asymptotic stability of the system with the designed controller implemented, together with full bottleneck link utilization and data loss elimination in the network, is demonstrated. In consequence, the proposed suboptimal control law allows for the maximum throughput in the investigated communication system. Moreover, since the transfer rates calculated for the sources are always nonnegative and limited, the described strategy can be feasibly implemented in real networks. Next, we analyze the methods of constraining the input signal. Three methods are considered: the application of a time-varying sliding plane, the design employing

P. Ignaciuk and A. Bartoszewicz, *Congestion Control in Data Transmission Networks: Sliding Mode and Other Designs*, Communications and Control Engineering,
DOI 10.1007/978-1-4471-4147-1_6, © Springer-Verlag London 2013

a reaching law, and the use of a saturation element. All three nonlinear control algorithms are shown to maintain the favorable properties of the LQ suboptimal SM controller related to packet loss elimination and full bandwidth usage. In addition, the transmission rate generated by the controllers is guaranteed to meet the input constraints without throttling the system dynamics.

In Sect. 6.2, we release the assumption about the fixed latency in packet and control unit delivery and design new algorithms for the system with uncertain, time-varying delay. We propose two robust controllers that can provide the maximum throughput in the network with unknown pattern of delay and bandwidth fluctuations. The external disturbances need not be correlated with each other. For the designed controllers, the most important properties related to handling the flow of data are defined and strictly proved. Following the analytical discussion, the described features are illustrated with numerous simulation examples presented for convenience at the end of each section.

6.1 Flow Control in a Network with Constant Delay

We consider the networking scenario in which multiple connections pass through the bottleneck node and contend for the bandwidth available at its output connection. We assume that the sources are persistent, meaning that they always have enough data to send at the rate determined by the controller placed at the node. The node at which the controller operates is assumed the bottleneck for the entire pool of flows passing through its outgoing link. The flows served at the output link which are not subject to the controller command (the uncontrolled flows) are treated as high-priority traffic with variable intensity that constitutes an external disturbance in the system. In this section, we assume that round-trip time of each regulated flow is constant and known to the controller. For instance, it can be measured or estimated in the connection setup phase.

The section is organized in the following way. First, the model of the wide-area network introduced in Sect. 5.1.1 is extended for the multisource traffic scenario considered in this chapter in Sect. 6.1.1. Afterwards, the system state-space description is modified according to the changes in the network model, and new SM control laws are designed based on formal control-theoretic methods. The properties of each controller are proved analytically and substantiated with the results of numerical tests.

6.1.1 Network Model

We analyze the situation of m data flows passing through the bottleneck node and its output connection. The principles of feedback information exchange are identical to

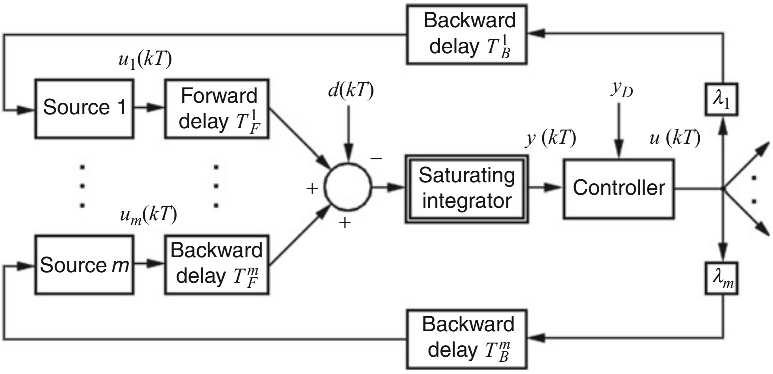


Fig. 6.1 Network model – multiple virtual circuits

the ones described in Sect. 5.1.1. This means that each source periodically emits control units, which travel through the network gathering the feedback information from the nodes. It is assumed that control units are served on a priority basis, i.e., they are not subject to packet queuing delays. Once the control units in each connection reach the respective destination, they are turned back and appear at their origin round-trip time after they were generated. As soon as control units arrive back at the sources, the feedback information is extracted and the source transmission rate is adjusted accordingly.

The model of the considered multisource network is illustrated in Fig. 6.1. The sources send packets at discrete-time instants kT (T is the discretization period and $k = 0, 1, 2, \ldots$) in the amounts determined by the controller placed at the bottleneck node. After forward delayT_{F}^{p}, packets from source p ($p = 1, 2, \ldots, m$) reach the bottleneck node and are served according to the bandwidth availability at the output link $d(kT)$. The remaining data accumulates in the buffer. The packet queue length in the buffer $y(kT)$ and its demand value y_{D} are used to calculate the current amount of data $u(kT)$ to be sent by the sources. The total amount $u(kT)$ is distributed among the connections according to the assumed resource allocation policy, for instance, using the customer priority or fairness criteria. Consequently, at each time instant kT, λ_p of the total rate is assigned to source p, where λ_p is a real number from the interval [0, 1] such that $\sum_{p=1}^{m} \lambda_p = 1$. In the limit case, when $\lambda_p = 1$, the entire rate is allocated to connection$p$, whereas $\lambda_p = 0$ implies no share of the total rate for this flow. As a result, the functions of flow and fairness control are decoupled and can be incorporated independently in the overall traffic regulation scheme (see, e.g., [7] for the discussion of numerous benefits of such approach). Once the control units from source p appear at the end system, they are turned back to arrive at their origin with backward propagation delay T_{B}^{p} after being processed by the bottleneck node. The source adapts its rate to the value read from the received control unit and transmits

$$u_p\,(kT) = \lambda_p u\left(kT - T_{\mathrm{B}}^{p}\right) \tag{6.1}$$

packets in period $[kT, (k+1)T)$. Since control units are not subject to queuing delays, $\mathrm{RTT}_p = T_{\mathrm{F}}^p + T_{\mathrm{B}}^p = n_p T$, where n_p is a positive integer, remains constant for the duration of data transfer in connection p. Without losing generality, we may order the flows according to the value of their RTT as in (4.22), which in the considered discrete-time system with uniform sampling is equivalent to the condition

$$n_1 \le n_2 \le \cdots \le n_{m-1} \le n_m. \tag{6.2}$$

The case of variable discretization period will be covered in Chap. 7.

The available bandwidth at the node output interface $d(kT)$ and the utilized bandwidth $h(kT)$ satisfy inequalities (5.2). The rate of change of the queue length is related to the overall incoming rate and the utilized bandwidth by the following equation

$$y\left[(k+1)T\right] = y(kT) + \sum_{p=1}^{m} u_p\left(kT - T_{\mathrm{F}}^p\right) - h(kT). \tag{6.3}$$

Using identity (6.1), we get

$$u_p\left(kT - T_{\mathrm{F}}^p\right) = \lambda_p u\left(kT - T_{\mathrm{F}}^p - T_{\mathrm{B}}^p\right) = \lambda_p u\left(kT - \mathrm{RTT}_p\right)$$

and

$$y\left[(k+1)T\right] = y(kT) + \sum_{p=1}^{m} \lambda_p u\left(kT - \mathrm{RTT}_p\right) - h(kT). \tag{6.4}$$

We assume that initially the buffer is empty, i.e., $y(0) = 0$. Then, the packet queue length at the node for any $kT \ge 0$ may be expressed as

$$y(kT) = \sum_{p=1}^{m} \sum_{j=0}^{k-1} \lambda_p u\left(jT - \mathrm{RTT}_p\right) - \sum_{j=0}^{k-1} h(jT). \tag{6.5}$$

Applying the definition $\mathrm{RTT}_p = n_p T$, we can rewrite (6.5) as

$$\begin{aligned} y(kT) &= \sum_{p=1}^{m} \sum_{j=0}^{k-1} \lambda_p u\left(jT - n_p T\right) - \sum_{j=0}^{k-1} h(jT) \\ &= \sum_{p=1}^{m} \sum_{j=-n_p}^{k-n_p-1} \lambda_p u(jT) - \sum_{j=0}^{k-1} h(jT). \end{aligned} \tag{6.6}$$

Therefore, if the control process commences at the time instant $kT = 0$ (the first rate is assigned to the sources at $kT = 0$), the initial packets belonging to connection p arrive at the node no sooner than at $kT = n_pT$. In consequence, we may represent (6.6) as

$$y(kT) = \sum_{p=1}^{m} \sum_{j=0}^{k-n_p-1} \lambda_p u(jT) - \sum_{j=0}^{k-1} h(jT). \tag{6.7}$$

This formula will be applied in the analysis of the controller properties conducted in a latter part of this chapter (Sect. 6.1.2).

6.1.1.1 State-Space Representation

Before we proceed with the formal controller design, it is convenient to represent the network model (6.1)–(6.7) in the state space. Choosing state space (5.7), we get

$$\begin{aligned} \mathbf{x}[(k+1)T] &= \mathbf{A}\mathbf{x}(kT) + \mathbf{b}u(kT) + \mathbf{v}h(kT), \\ y(kT) &= \mathbf{q}^{\mathrm{T}}\mathbf{x}(kT), \end{aligned}$$

where $\mathbf{b}$, $\mathbf{v}$, and $\mathbf{q}$ are defined by (5.8) and the system order $n = n_m + 1 = (\mathrm{RTT}_m/T) + 1$ depends on the discretization period T and the longest round-trip time in the pool of connections passing through the node. In order to take into account the traffic coming from multiple sources, the state matrix given in (5.8) for the case of a single flow needs to be modified in the following way:

$$\mathbf{A} = \begin{bmatrix} 1 & a_{n-1} & a_{n-2} & \dots & a_1 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}. \tag{6.8}$$

The elements in the first row of $\mathbf{A}$,

$$a_j = \sum_{p:\, \mathrm{RTT}_p = jT} \lambda_p, \tag{6.9}$$

denote the share of connections with delay jT ($j = 1, 2, \dots, n_m$) in the total transmission rate allocated by the controller. Obviously, if no connection is characterized by delay jT, then the corresponding share a_j in the total rate equals zero. Taking into account the condition $\sum_{p=1}^{m} \lambda_p = 1$, we have

$$\sum_{j=1}^{n-1} a_j = 1. \tag{6.10}$$

With the state matrix given by (6.8), the system dynamics (5.7) can be presented in the alternative form:

$$\begin{cases} x_1\left[(k+1)\,T\right] &= x_1\left(kT\right) + a_{n-1}x_2\left(kT\right) + \cdots + a_1 x_n\left(kT\right) - h\left(kT\right), \\ x_2\left[(k+1)\,T\right] &= x_3\left(kT\right), \\ x_3\left[(k+1)\,T\right] &= x_4\left(kT\right), \\ \quad\vdots & \\ x_{n-1}\left[(k+1)\,T\right] &= x_n\left(kT\right), \\ x_n\left[(k+1)\,T\right] &= u\left(kT\right). \end{cases} \tag{6.11}$$

The desired system state is defined as $\mathbf{x_d} = [y_D\ 0\ 0 \ldots\ 0]^T$, which means that the control objective is to stabilize the queue length $x_1(kT) = y(kT)$ at the level y_D. For this to occur, all the remaining state variables $x_1, x_2, \ldots, x_n$, which represent the incoming rate, should be equal to zero once $x_1(kT) = y_D$. It is evident from (6.11) that vector $\mathbf{x}(kT)$ together with the interaction among the state variables specified by matrix (6.8) provides a complete description of the network with multiple sources specified by Eqs. (6.1)–(6.7). The way matrix $\mathbf{A}$ is constructed for different network configurations is illustrated in Example 6.1.

Example 6.1. In order to see how a particular network configuration relates to the state-space description, we need to analyze the structure of matrix $\mathbf{A}$ that defines the interaction among the state variables. Recall that $x_1(kT)$ represents the queue length at instant kT and $x_j(kT) = u[(k-n+j-1)T]$ for any $j = 2, \ldots, n$ equals the delayed input signal $u(\cdot)$.

Scenario 1. Let us first consider the situation in which two sources ($m = 2$) compete for the bandwidth at the node output interface. The connections are characterized by the delays $\text{RTT}_1 = 20$ ms and $\text{RTT}_2 = 60$ ms. With the discretization period $T = 20$ ms, we get $\text{RTT}_1 = n_1 T = T$, $\text{RTT}_2 = n_2 T = 3\,T$, and the system order related to the flow with the longest delay $n = n_2 + 1 = 4$. If the equal rate partitioning $\lambda_1 = \lambda_2 = 1/2$ is applied, we get $a_1 = a_3 = 1/2$, $a_2 = 0$, and matrix $\mathbf{A}$,

$$\mathbf{A} = \begin{bmatrix} 1 & a_3 & a_2 & a_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 & 0 & 0.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

In the situation when the discretization period is reduced to $T = 10$ ms, we get $\text{RTT}_1 = n_1 T = 2\,T$, $\text{RTT}_2 = n_2 T = 6\,T$, and the system order $n = n_2 + 1 = 7$. With the equal rate partitioning $\lambda_1 = \lambda_2 = 1/2$, we get $a_2 = a_6 = 1/2$, $a_1 = a_3 = a_4 = a_5 = 0$, and matrix $\mathbf{A}$,

$$\mathbf{A} = \begin{bmatrix} 1 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Scenario 2. In the second scenario, we consider the network with $m = 7$ connections passing through the bottleneck node. The connections are characterized by the delays: $\mathrm{RTT}_1 = 20$ ms, $\mathrm{RTT}_2 = 30$ ms, $\mathrm{RTT}_3 = \mathrm{RTT}_4 = \mathrm{RTT}_5 = 50$ ms, and $\mathrm{RTT}_6 = \mathrm{RTT}_7 = 70$ ms. Assuming the discretization period $T = 10$ ms, we get $n_1 = 2$, $n_2 = 3$, $n_3 = n_4 = n_5 = 5$, $n_6 = n_7 = 7$, and the system order $n = n_7 + 1 = 8$. If the equal rate partitioning $\lambda_p = 1/m = 1/7$ is applied ($p = 1, 2, \ldots, 7$), we get $a_2 = a_3 = 1/7$, $a_5 = 3/7$, $a_7 = 2/7$, $a_1 = a_4 = a_6 = 0$, and the first row of $\mathbf{A}$,

$$\left[a_{1j}\right] = \left[1 \quad 2/7 \quad 0 \quad 3/7 \quad 0 \quad 1/7 \quad 1/7 \quad 0\right].$$

However, if a different rate allocation policy is applied, for example, the one given by $\lambda_1 = \lambda_3 = 0.25$, $\lambda_2 = \lambda_4 = \lambda_5 = \lambda_6 = \lambda_7 = 0.1$, we get from (6.9)

$$\begin{aligned}
a_1 &= \sum\nolimits_{p:\mathrm{RTT}_p = T} \lambda_p = 0, \\
a_2 &= \sum\nolimits_{p:\mathrm{RTT}_p = 2T} \lambda_p = \lambda_1 = 0.25, \\
a_3 &= \sum\nolimits_{p:\ \mathrm{RTT}_p = 3T} \lambda_p = \lambda_2 = 0.1, \\
a_4 &= \sum\nolimits_{p:\mathrm{RTT}_p = 4T} \lambda_p = 0, \\
a_5 &= \sum\nolimits_{p:\ \mathrm{RTT}_p = 5T} \lambda_p = \lambda_3 + \lambda_4 + \lambda_5 = 0.45, \\
a_6 &= \sum\nolimits_{p:\ \mathrm{RTT}_p = 6T} \lambda_p = 0, \\
a_7 &= \sum\nolimits_{p:\mathrm{RTT}_p = 7T} \lambda_p = \lambda_6 + \lambda_7 = 0.2
\end{aligned}$$

and the first row of $\mathbf{A}$,

$$\left[a_{1j}\right] = \left[1 \quad 0.2 \quad 0 \quad 0.45 \quad 0 \quad 0.1 \quad 0.25 \quad 0\right].$$

6.1.2 SM Controller with LQ Suboptimal Sliding Plane

In this section, we design a discrete-time SM controller for the multisource communication network described above. The principles of LQ optimal control are applied to get parameters of the sliding plane. Since the exact solution of the LQ problem in the case of the network serving multiple connections with different RTTs results in a controller generating negative transmission rates, we propose a modified optimization procedure to get a suboptimal controller free of this drawback. The closed-loop stability of the multisource system with the designed suboptimal controller implemented is demonstrated, and the essential properties of the proposed strategy are discussed and strictly proved.

6.1.2.1 Controller Design

The control law derivation presented in this section concentrates on the selection of parameters of an appropriate sliding plane. Substituting (5.7) into equation $\mathbf{c}^{\mathrm{T}}\mathbf{e}[(k+1)T] = 0$, and performing algebraic manipulations (5.12), we get

$$u(kT) = \left(\mathbf{c}^{\mathrm{T}}\mathbf{b}\right)^{-1}\mathbf{c}^{\mathrm{T}}\left[\mathbf{x_d} - \mathbf{A}\mathbf{x}(kT)\right].$$

Substituting vector quantities (5.8) and matrix $\mathbf{A}$ given by (6.8) into this equation, we obtain the following feedback control law:

$$u(kT) = c_n{}^{-1}\left\{c_1\left[y_{\mathrm{D}} - x_1(kT) - a_{n-1}x_2(kT)\right] - \sum_{j=3}^{n}\left(c_1 a_{n-j+1} + c_{j-1}\right)x_j(kT)\right\}. \tag{6.12}$$

Similarly as in Sect. 5.1.2, we intend to find such parameters of the sliding plane (described by vector $\mathbf{c}$) that will allow for the minimum control effort in reducing the system error to zero. For this purpose, we apply LQ optimization with tunable quality criterion (5.41) and solve the Riccati equation stated in (5.43).

However, careful investigation of the elements of matrix $\mathbf{K}$ and vector $\mathbf{g}$ obtained for the multisource scenario, confirmed by numerical computations, reveals a serious drawback of the exact solution to the LQ optimal control problem, namely, the controller generates negative transmission rates. To eliminate this deficiency and make the scheme applicable for real networks, we appropriately modify the optimization procedure introduced in Sect. 5.1.2 for the single-source case. Notice that each coefficient $a_1, a_2, \ldots, a_{n-1}$ is smaller than or equal to one.

Moreover, according to (6.10), if any a_j equals one, then the remaining coefficients are zero. Hence, in typical multisource networks, the majority of the products $a_i a_j \ll 1$, $i, j \in \{1, 2, \ldots, n-1\}$ and can be neglected in the calculations of the elements of $\mathbf{K}$ performed according to (5.43). Applying this modification in the solution procedure, instead of **K,** we will obtain a close approximation – $\hat{\mathbf{K}}$. In the further part of this section, it will be demonstrated that the proposed approximation actually allows us to formulate an enhanced control law, which indeed guarantees that the flow rates are always nonnegative and the packet queue length converges to steady state without overshoots or oscillations.

Similarly as in the case of the single-flow control, we apply an iterative, analytical procedure to solve the Riccati equation (5.43) and determine the elements of matrix $\hat{\mathbf{K}}$. We begin solving (5.43) with the most general form of $\hat{\mathbf{K}}$,

$$\hat{\mathbf{K}}_0 = \begin{bmatrix} \hat{k}_{11} & \hat{k}_{12} & \ldots & \hat{k}_{1n} \\ \hat{k}_{12} & \hat{k}_{22} & \ldots & \hat{k}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{k}_{1n} & \hat{k}_{2n} & \ldots & \hat{k}_{nn} \end{bmatrix}, \tag{6.13}$$

which is placed directly in (5.43). After substituting matrix $\mathbf{A}$ given by (6.8) and vector $\mathbf{b}$ as defined by (5.8), we look for similarities between the elements $\hat{k}_{11}, \hat{k}_{12}, \ldots, \hat{k}_{nn}$ on either side of the equality sign of the resulting equation. In this way, we find the relations among the first four elements in the upper left corner of $\hat{\mathbf{K}} : \hat{k}_{12} = a_{n-1}(\hat{k}_{11} - w)$ and $\hat{k}_{22} = a_{n-1}^2(\hat{k}_{11} - w)$. Consequently, after the first analytical iteration, we obtain the following form of $\hat{\mathbf{K}}$:

$$\hat{\mathbf{K}}_1 = \begin{bmatrix} \hat{k}_{11} & a_{n-1}\left(\hat{k}_{11} - w\right) & \hat{k}_{13} & \ldots & \hat{k}_{1n} \\ a_{n-1}\left(\hat{k}_{11} - w\right) & a_{n-1}^2\left(\hat{k}_{11} - w\right) & \hat{k}_{23} & \ldots & \hat{k}_{2n} \\ \hat{k}_{13} & \hat{k}_{23} & \hat{k}_{33} & \ldots & \hat{k}_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{k}_{1n} & \hat{k}_{2n} & \hat{k}_{3n} & \ldots & \hat{k}_{nn} \end{bmatrix}. \tag{6.14}$$

Now we substitute $\hat{\mathbf{K}}_1$ given by (6.14) into the expression on the right-hand side of (5.43) and compare with its left-hand side, which allows us to represent $\hat{k}_{i3}$ ($i = 1$, 2, 3) in terms of $\hat{k}_{11} : \hat{k}_{13} = \sum_{j=1}^{2} a_{n-j}\left[\hat{k}_{11} - (3-j)\,w\right]$, $\hat{k}_{23} = a_{n-1}\hat{k}_{13}$, and $\hat{k}_{33} = (a_{n-1} + a_{n-2})\,\hat{k}_{13}$. This results in (for the sake of clarity, we present only the upper part of the symmetric matrix $\hat{\mathbf{K}}$)

$$\hat{\mathbf{K}}_2 = \begin{bmatrix} \hat{k}_{11} & a_{n-1}\left(\hat{k}_{11} - w\right) & \sum\limits_{j=1}^{2} a_{n-j}\left[\hat{k}_{11} - (3-j)\,w\right] & \hat{k}_{14} & \ldots & \hat{k}_{1n} \\ \hat{k}_{12} & a_{n-1}^2\left(\hat{k}_{11} - w\right) & a_{n-1}\sum\limits_{j=1}^{2} a_{n-j}\left[\hat{k}_{11} - (3-j)\,w\right] & \hat{k}_{24} & \ldots & \hat{k}_{2n} \\ \hat{k}_{13} & \hat{k}_{23} & \sum\limits_{j=1}^{2} a_{n-j} \cdot \sum\limits_{j=1}^{2} a_{n-j}\left[\hat{k}_{11} - (3-j)\,w\right] & \hat{k}_{34} & \ldots & \hat{k}_{3n} \\ \hat{k}_{14} & \hat{k}_{24} & \hat{k}_{34} & \hat{k}_{44} & \ldots & \hat{k}_{4n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{k}_{1n} & \hat{k}_{2n} & \hat{k}_{3n} & \hat{k}_{4n} & \ldots & \hat{k}_{nn} \end{bmatrix}. \tag{6.15}$$

We continue the iteration until all the elements of $\hat{\mathbf{K}}$ can be expressed as functions of $\hat{k}_{11}$ and the system order n. The final form of $\hat{\mathbf{K}}$, given in terms of its first element $\hat{k}_{11}$ and the system order, is determined as

$$\begin{bmatrix} \hat{k}_{11} & a_{n-1}\left(\hat{k}_{11} - w\right) & \sum\limits_{j=1}^{2} a_{n-j}\left[\hat{k}_{11} - (3-j)\,w\right] & \ldots & \sum\limits_{j=1}^{n-1} a_{n-j}\left[\hat{k}_{11} - (n-j)\,w\right] \\ \hat{k}_{12} & a_{n-1}{}^2\left(\hat{k}_{11} - w\right) & a_{n-1}\sum\limits_{j=1}^{2} a_{n-j}\left[\hat{k}_{11} - (3-j)\,w\right] & \ldots & a_{n-1}\sum\limits_{j=1}^{n-1} a_{n-j}\left[\hat{k}_{11} - (n-j)\,w\right] \\ \hat{k}_{13} & \hat{k}_{23} & \sum\limits_{j=1}^{2} a_{n-j} \cdot \sum\limits_{j=1}^{2} a_{n-j}\left[\hat{k}_{11} - (3-j)\,w\right] & \ldots & \sum\limits_{j=1}^{2} a_{n-j} \cdot \sum\limits_{j=1}^{n-1} a_{n-j}\left[\hat{k}_{11} - (n-j)\,w\right] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{k}_{1n} & \hat{k}_{2n} & \hat{k}_{3n} & \ldots & \sum\limits_{j=1}^{n-1} a_{n-j} \cdot \sum\limits_{j=1}^{n-1} a_{n-j}\left[\hat{k}_{11} - (n-j)\,w\right] \end{bmatrix}. \tag{6.16}$$

Notice that according to (6.10), $\sum_{j=1}^{n-1} a_{n-j} = 1$, and $\hat{k}_{nn}$ reduces to $\sum_{j=1}^{n-1} a_{n-j}\left[\hat{k}_{11} - (n-j)\,w\right]$. In order to complete the solution of the Riccati equation, we need yet to determine $\hat{k}_{11}$. For this purpose, we substitute (6.16) for $\mathbf{K}$ in the expression on the right-hand side of (5.43) and compare the first element in the upper left corner of the obtained matrices. This comparison yields the following equation:

$$\hat{k}_{11} = 1 + w\left(\sum_{j=1}^{n-1} j a_j + 1\right) - \left(\hat{k}_{11} - w\sum_{j=1}^{n-1} j a_j + 1\right)^{-1}. \tag{6.17}$$

Solving for $\hat{k}_{11}$, we arrive at

$$\hat{k}_{11}^{\pm} = \left[w\left(2\sum_{j=1}^{n-1} j a_j + 1\right) \pm \sqrt{w\,(w+4)}\right] \Big/ 2. \tag{6.18}$$

Out of the obtained roots, only $\hat{k}_{11}^{+}$ guarantees that all the leading principal minors and the determinant of $\hat{\mathbf{K}}$ are positive, and $\hat{\mathbf{K}}$ is positive definite, thus giving the desired solution of (5.43).

Having found $\hat{\mathbf{K}}$, we may determine $\hat{\mathbf{g}} = \mathbf{b}^{\mathrm{T}}\hat{\mathbf{K}}(\mathbf{I}_n + \mathbf{b}\mathbf{b}^{\mathrm{T}}\hat{\mathbf{K}})^{-1}\mathbf{A}$. Substituting (6.16) into the first equation in set (5.18) with matrix $\mathbf{A}$ given by (6.8), we obtain

$$\hat{\mathbf{g}} = \left[1 \; a_{n-1} \; (a_{n-1} + a_{n-2}) \; \ldots \; \sum_{j=1}^{n-1} a_j \right] \left[1 - \left(\hat{k}_{11} - w \sum_{j=1}^{n-1} j a_j + 1 \right)^{-1} \right]. \tag{6.19}$$

Substituting $\hat{k}_{11}^{+}$ and using the fact that $\sum_{j=1}^{n-1} a_{n-j} = 1$, we get

$$1 - \left(\hat{k}_{11} - w \sum_{j=1}^{n-1} j a_j + 1 \right)^{-1} = \frac{\left(\sqrt{w\,(w+4)} - w \right)}{2} \tag{6.20}$$

and

$$\hat{\mathbf{g}} = \left[1 \; a_{n-1} \; (a_{n-1} + a_{n-2}) \; \ldots \; (a_{n-1} + \cdots + a_2) \; 1 \right] \frac{\left(\sqrt{w\,(w+4)} - w \right)}{2}. \tag{6.21}$$

Substituting (6.21) into (5.17), we arrive at

$$\hat{u}_{\mathrm{opt}}\,(kT) = -\hat{\mathbf{g}}\mathbf{x}\,(kT) + \hat{r} = -\gamma_3 \left[x_1\,(kT) + \sum_{j=2}^{n} \left(\sum_{i=1}^{j-1} a_{n-i} \right) x_j\,(kT) \right] + \hat{r}, \tag{6.22}$$

where

$$\gamma_3 = \frac{\left(\sqrt{w\,(w+4)} - w \right)}{2}. \tag{6.23}$$

Comparing (6.22) with (6.12), we get $\hat{r} = y_{\mathrm{D}} c_1 / c_2$ and the elements of vector $\mathbf{c}$,

$$\mathbf{c}^{\mathrm{T}} = \left[\gamma_3 \; \gamma_3 a_{n-1} \; \gamma_3\,(a_{n-1} + a_{n-2}) \; \ldots \; \gamma_3\,(a_{n-1} + \cdots + a_2) \; 1 \right] c_n. \tag{6.24}$$

Consequently, the control law

$$u\,(kT) = \gamma_3 \left[y_{\mathrm{D}} - x_1\,(kT) - \sum_{j=2}^{n} \left(\sum_{i=1}^{j-1} a_{n-i} \right) x_j\,(kT) \right]. \tag{6.25}$$

This concludes the solution of the optimization problem. Notice that the gain constant of the obtained controller is identical to the gain of controller (5.60). Comparing (6.25) with (5.60), we conclude that the suboptimal control law designed for the general case of a multisource network is equivalent to the optimal one when applied to the system with a single connection. Further in the text, we will use γ instead of γ_3 to denote the gain of controller (6.25).

Similarly as in (5.65), we substitute $x_1(kT) = y(kT)$ and the other state variables expressed in terms of the control signal generated at the previous $n-1$ samples, $x_j(kT) = u[(k-n+j-1)T]$ for $j = 2, 3, \ldots, n$, and obtain

$$u\left(kT\right) = \gamma \left\{ y_{\mathrm{D}} - y\left(kT\right) - \sum_{j=2}^{n} \left(\sum_{i=1}^{j-1} a_{n-i} \right) u\left[\left(k-n+j-1\right)T\right] \right\}. \tag{6.26}$$

Taking into account the relation $a_j = \sum_{p:\mathrm{RTT}_p = jT} \lambda_p$, the discrete-time control law (6.26) can be more conveniently presented in the following form:

$$u\left(kT\right) = \gamma \left[y_{\mathrm{D}} - y\left(kT\right) - \sum_{p=1}^{m} \lambda_p \sum_{j=k-n_p}^{k-1} u\left(jT\right) \right]. \tag{6.27}$$

This completes the design of the flow control algorithm for the multisource traffic scenario.

Stability Analysis

The system is asymptotically stable, if all the roots of the characteristic polynomial of the closed-loop system state matrix $\mathbf{A}_{\mathbf{cl}} = [\mathbf{I}_n - \mathbf{b}(\mathbf{c}^{\mathrm{T}}\mathbf{b})^{-1}\mathbf{c}^{\mathrm{T}}]\mathbf{A}$ are located within the unit circle. The roots of the polynomial

$$\begin{aligned}\det\left(z\mathbf{I}_n - \mathbf{A}_{\mathbf{cl}}\right) &= z^n + \left(\frac{a_1 c_1}{c_n} + \frac{c_{n-1} - c_n}{c_n}\right) z^{n-1} + \left(\frac{a_2 c_1}{c_n} + \frac{c_{n-2} - c_{n-1}}{c_n}\right) \\ &\quad \times z^{n-2} + \cdots + \left(\frac{a_{n-1} c_1}{c_n} - \frac{c_2}{c_n}\right) z \\ &= z^n + \left(\gamma - 1\right) z^{n-1} = z^{n-1}\left[z - \left(1 - \gamma\right)\right]\end{aligned} \tag{6.28}$$

are located inside the unit circle, if $0 < \gamma < 2$. Since for every w, $\gamma = (\sqrt{w(w+4)} - w)/2$ is positive and not bigger than one, the system is stable and no oscillations appear at the output. Indeed, if we change w from 0 to ∞, the nonzero pole moves from the point (1, 0i) towards the origin of the z-plane, yet it never leaves the nonnegative part of the real axis implying oscillation-free performance. Moreover, an increase of w results in faster convergence to the target state. In the limit case when $w = \infty$, all the closed-loop poles are at the origin ensuring the fastest achievable response of a linear controller in discrete-time system – a dead-beat scheme.

6.1.2.2 Properties of the Proposed Controller

The properties of the flow control algorithm (6.27) designed for the multisource network will be formulated as three theorems. The first theorem defines the buffer capacity at the bottleneck node, which is required to ensure loss-free transmission irrespective of the actual bandwidth at the bottleneck node output link. The second proposition indicates how the demand queue length should be selected to ensure the total bandwidth usage. Finally, the third theorem states that the transmission rate established by the controller is always guaranteed to be nonnegative and bounded.

Theorem 6.1. *If controller (6.27) is applied to system (5.7) with matrix* **A** *defined by (6.8), then the queue length is always upper-bounded by its demand value y_D.*

Proof. Due to the flow ordering $\mathrm{RTT}_1 \le \mathrm{RTT}_2 \le \cdots \le \mathrm{RTT}_m$, the first packets may reach the node at $kT = \mathrm{RTT}_1$. Consequently, it follows from the assumed initial conditions, $y(0) = 0$ and $u(kT) = 0$ for $k < 0$, that the bottleneck node buffer is empty for any $kT \le \mathrm{RTT}_1 = n_1 T$. Hence, it suffices to show that the proposition is satisfied for any $k \ge n_1 + 1$. Let us assume that for some integer $l \ge n_1 + 1$, $y(lT) \le y_\mathrm{D}$. We will demonstrate that the theorem is also true for $l + 1$. Substituting (6.7) into (6.27), we get

$$u(lT) = \gamma \left\{ y_\mathrm{D} - \sum_{p=1}^{m} \lambda_p \sum_{j=0}^{l-n_p-1} u(jT) + \sum_{j=0}^{l-1} h(jT) - \sum_{p=1}^{m} \lambda_p \sum_{j=l-n_p}^{l-1} u(jT) \right\}$$

$$= \gamma \left[y_\mathrm{D} - \sum_{p=1}^{m} \lambda_p \sum_{j=0}^{l-1} u(jT) + \sum_{j=0}^{l-1} h(jT) \right]. \tag{6.29}$$

Since $\sum_{p=1}^{m} \lambda_p = 1$, we can simplify (6.29) in the following way:

$$u(lT) = \gamma \left[y_\mathrm{D} - \sum_{j=0}^{l-1} u(jT) + \sum_{j=0}^{l-1} h(jT) \right]. \tag{6.30}$$

Using (6.4), the queue length at the $(l+1)T$ time instant can be expressed as

$$y\left[(l+1)T\right] = y\left(lT\right) + \sum_{p=1}^{m} \lambda_p u\left(lT - n_p T\right) - h\left(lT\right). \tag{6.31}$$

Substituting (6.30) for $u(lT - n_p T)$, we get

$$\begin{aligned} y\left[(l+1)T\right] &= y\left(lT\right) + \sum_{p=1}^{m} \lambda_p \gamma \left[y_{\mathrm{D}} - \sum_{j=0}^{l-n_p-1} u\left(jT\right) + \sum_{j=0}^{l-n_p-1} h\left(jT\right) \right] - h\left(lT\right) \\ &= y\left(lT\right) + \gamma y_{\mathrm{D}} \sum_{p=1}^{m} \lambda_p + \gamma \sum_{p=1}^{m} \lambda_p \left[- \sum_{j=0}^{l-n_p-1} u\left(jT\right) + \sum_{j=0}^{l-n_p-1} h\left(jT\right) \right] - h\left(lT\right). \end{aligned} \tag{6.32}$$

Once again, since $\sum_{p=1}^{m} \lambda_p = 1$, we can rewrite (6.32) as

$$\begin{aligned} y\left[(l+1)T\right] = y\left(lT\right) + \gamma y_{\mathrm{D}} - \gamma &\left[\sum_{p=1}^{m} \lambda_p \sum_{j=0}^{l-n_p-1} u\left(jT\right) - \sum_{j=0}^{l-1} h\left(jT\right) \right] \\ &- \gamma \sum_{p=1}^{m} \lambda_p \sum_{j=l-n_p}^{l-1} h\left(jT\right) - h\left(lT\right). \end{aligned} \tag{6.33}$$

It follows from (6.7) that the term in the square brackets in (6.32) actually equals $y(lT)$. Consequently,

$$\begin{aligned} y\left[(l+1)T\right] &= y\left(lT\right) + \gamma y_{\mathrm{D}} - \gamma y\left(lT\right) - \gamma \sum_{p=1}^{m} \lambda_p \sum_{j=l-n_p}^{l-1} h\left(jT\right) - h\left(lT\right) \\ &= y_{\mathrm{D}} - y_{\mathrm{D}} + \gamma y_{\mathrm{D}} + y\left(lT\right) - \gamma y\left(lT\right) - \gamma \sum_{p=1}^{m} \lambda_p \sum_{j=l-n_p}^{l-1} h\left(jT\right) - h\left(lT\right) \\ &= y_{\mathrm{D}} - (1-\gamma)\left[y_{\mathrm{D}} - y\left(lT\right)\right] - \gamma \sum_{p=1}^{m} \lambda_p \sum_{j=l-n_p}^{l-1} h\left(jT\right) - h\left(lT\right). \end{aligned} \tag{6.34}$$

Since $0 < \gamma \leq 1$ and $h(\cdot)$ is always nonnegative, $y[(l+1)T] \leq y_{\mathrm{D}}$. Therefore, using the principle of the mathematical induction, we conclude that the theorem is true for any time instant $kT \geq 0$. This ends the proof. □

Theorem 6.1 states that the buffer of size y_{D} suffices to ensure loss elimination at the bottleneck node no matter the evolution of the available bandwidth. The proposition formulated below indicates how y_{D} should be selected in order to ensure full bandwidth usage in the considered communication system.

Theorem 6.2. *If controller (6.27) is applied to system (5.7) with matrix* **A** *defined by (6.8), and the demand queue length satisfies*

$$y_{\mathrm{D}} > d_{\max}\left(\sum_{p=1}^{m}\lambda_p n_p + \frac{1}{\gamma}\right), \tag{6.35}$$

then for any $k \geq n_m + 1$, the queue length is strictly positive.

Proof. It follows from (6.7), (6.34), and (6.35) that $y(k = n_m + 1) > 0$. Let us assume that for some integer $l > n_m + 1$, the queue length is positive. We shall demonstrate that $y[(l+1)T]$ is also greater than zero. Since $\gamma \in (0, 1]$, then from (6.34), we get

$$\begin{aligned} y\left[(l+1)T\right] &= (1-\gamma)\,y\,(lT) + \gamma y_{\mathrm{D}} - \gamma\sum_{p=1}^{m}\lambda_p \sum_{j=l-n_p}^{l-1} h\,(jT) - h\,(lT) \\ &\geq \gamma y_{\mathrm{D}} - \gamma\sum_{p=1}^{m}\lambda_p \sum_{j=l-n_p}^{l-1} h\,(jT) - h\,(lT)\,. \end{aligned} \tag{6.36}$$

According to (5.2), the maximum buffer depletion rate equals $d_{\max}$. Therefore,

$$\begin{aligned} y\left[(l+1)T\right] &\geq \gamma y_{\mathrm{D}} - \gamma\sum_{p=1}^{m}\lambda_p \sum_{j=l-n_p}^{l-1} d_{\max} - d_{\max} \\ &= \gamma\left[y_{\mathrm{D}} - d_{\max}\left(\sum_{p=1}^{m}\lambda_p n_p + \frac{1}{\gamma}\right)\right]. \end{aligned} \tag{6.37}$$

Consequently, from the theorem assumption (6.35), we get $y[(l+1)T] > 0$, which completes the induction proof. □

For any flow control strategy to be feasibly applicable in real networks, the generated transmission rate must be always nonnegative and limited. This property of the proposed discrete-time control law is demonstrated in the next theorem.

Theorem 6.3. *If controller (6.27) is applied to system (5.7) with matrix* **A** *defined by (6.8), then the transmission rate generated by the controller satisfies inequalities (5.77).*

Proof. For $k = 0$, we have $u(0) = \gamma y_{\mathrm{D}}$, and the theorem is true at the initial time. For any integer $l \geq 0$, the transmission rate may be calculated from (6.30), which is independent of the number of connections. Consequently, by repeating reasoning (5.78), we get $0 \leq u(lT) \leq \max(\gamma y_{\mathrm{D}}, d_{\max})$. This ends the proof. □

Remark 6.1. It follows from Theorems 6.1 and 6.2 that the buffer capacity which needs to be reserved at the bottleneck node to guarantee loss-free and maximally efficient data transfer depends on the average propagation delay of the controlled connections and not their number. Therefore, since a common buffer is used for the entire pool of flows, the proposed flow control algorithm ensures a scalable memory allocation policy at the node even for a large number of connections served by the network.

Remark 6.2. In the considered network, the amount of data generated by the controller is allocated to the sources according to the choice of λ_p. Consequently, the amount of data to be sent by each source is determined as $\lambda_p u(\cdot)$ which may constitute a noninteger multiple of the packet length. In order to overcome this problem, one can introduce $v_p(kT) = \lfloor \lambda_p u(kT) \rfloor$, where $\lfloor x \rfloor$ denotes the greatest integer not exceeding x. Then, function (6.27) describing the controller operation is modified in the following way:

$$u(kT) = \gamma \left[y_{\mathrm{D}} - y(kT) - \sum_{p=1}^{m} \sum_{j=k-n_p}^{k-1} v_p(jT) \right]. \tag{6.38}$$

This modified strategy ensures that the amount of data sent by the sources is always an integer multiple of the packet length. Moreover, if the lower limit of the demand queue length y_{D}, given by (6.35), is increased by m packets, then the strategy still guarantees that following the initial period, there will always be enough packets in the bottleneck node buffer to exploit all of the available bandwidth. Consequently, if the buffer is assigned according to the guidelines of Theorem 6.1, the maximum throughput will be achieved.

In the next section, the properties of the proposed control strategy, governing the flow of data in multiple connections competing for the bandwidth at the bottleneck link, will be illustrated in numerical tests.

6.1.2.3 Simulation Results

In order to verify the operation of the designed flow control algorithm (6.27), we run several simulation tests for different networking scenarios and controller parameter settings.

Test 1. In the first simulation example, we verify the performance of the designed control law when the available bandwidth at the bottleneck link evolves as shown in Fig. 6.2. One can notice that the bandwidth follows a similar pattern as in the case of the single-source flow control considered in Chap. 5 (and illustrated Fig. 5.7) except for an increased amplitude of variations. Consequently, in Test 1, we study the algorithm behavior in the response to abrupt bandwidth changes, including the link on-off transitions (occurring at instants $120T$ and $180T$). The maximum available bandwidth $d_{\max}$ is set as 100 packets in Test 1 and all the other tests discussed in this section.

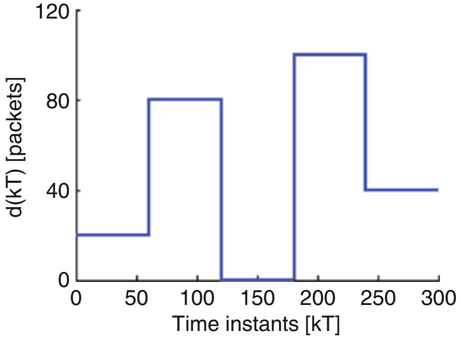


Fig. 6.2 Available bandwidth

Table 6.1 Controller parameters in Test 1

Controller	Weighting factor w	Controller gain γ	Demand queue length y_D [packets]
Linear controller (6.27)	0.09	0.259	$1{,}140 > 1{,}136$
	1	0.618	$915 > 912$
Nonlinear controller (4.29)	–	∞	$830 > 825$

We assume that four connections ($m = 4$) participate in the flow regulation process. They are characterized by the following delays: $\mathrm{RTT}_1 = 3T$, $\mathrm{RTT}_2 = 7T$, $\mathrm{RTT}_3 = 8T$, and $\mathrm{RTT}_4 = 12T$, where $T = 10$ ms represents the discretization period. Hence, the system order $n = n_4 + 1 = 13$. With equal rate allocation, $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1/m = 1/4$, we get the first row of the state matrix:

$$\left[a_{1j}\right] = \left[1 \quad 0.25 \quad 0 \quad 0 \quad 0 \quad 0.25 \quad 0.25 \quad 0 \quad 0 \quad 0 \quad 0.25 \quad 0 \quad 0\right].$$

Consequently, vector (6.24) describing the sliding plane used by controller (6.27) is determined as

$$\mathbf{c}^{\mathrm{T}} = \left[\gamma \ 0.25\gamma \ 0.25\gamma \ 0.25\gamma \ 0.25\gamma \ 0.5\gamma \ 0.75\gamma \ 0.75\gamma \ 0.75\gamma \ 0.75\gamma \ \gamma \ \gamma \ 1\right] c_n.$$

A number of simulations are run, each for a different value of the gain constant γ. The obtained results are compared with the ones obtained for the on-off controller (4.29) adapted for the discrete-time system considered here. The demand queue length y_D for the linear controller (6.27) is adjusted according to the inequality specified in Theorem 6.2. For controller (4.29), we assume $u_{\max} = 110$ packets and set y_D according to Theorem 4.4 as 830 packets. The gain and y_D settings are grouped in Table 6.1 (note that the switching action of the on-off controller can be perceived as the transmission rate adjustment with an infinite gain). The transmission rate generated by the controllers is shown in Figs. 6.3 and 6.4 (initial phase), the buffer occupancy in Fig. 6.5, and the sliding variable in Fig. 6.6.

The plots in Fig. 6.3 clearly demonstrate that the transmission rate established by controller (6.27) is always nonnegative and upper-bounded, exactly as stated in Theorem 6.3. Moreover, the packet queue length in the bottleneck node buffer

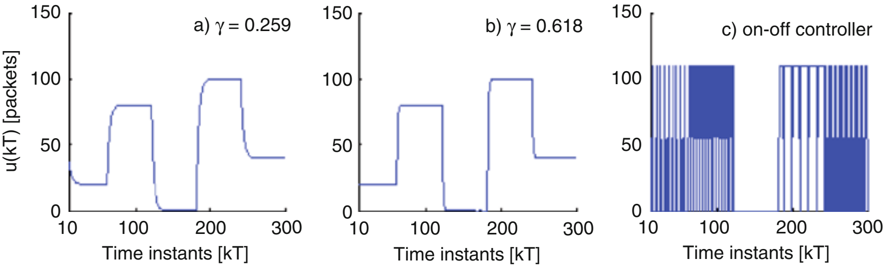
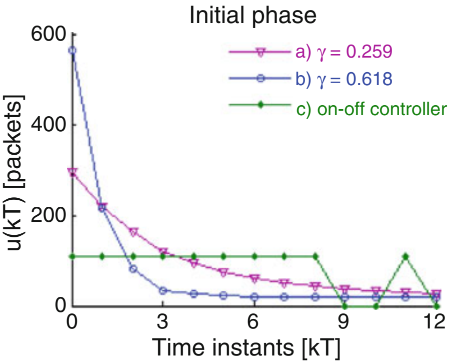


Fig. 6.3 Transmission rate: linear controller *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, and *c* nonlinear controller

Fig. 6.4 Initial transfer rate: linear controller *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, and *c* nonlinear controller



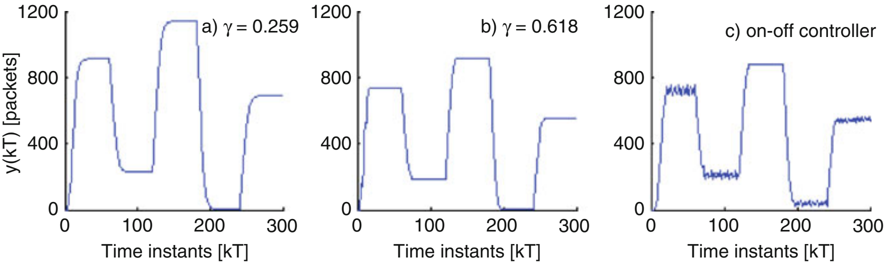


Fig. 6.5 Queue length: linear controller *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, and *c* nonlinear controller

depicted in Fig. 6.5 does not increase beyond the demand value and never drops to zero (for $k \geq n + 1 = 14$), which ensures loss-free and maximally efficient data transfer. We can also see from the plots in Fig. 6.3 that the choice of the weighting coefficient influences the system dynamics in a similar way as discussed in Sect. 5.1.2. As w increases, the controller reacts faster to the fluctuations of the available bandwidth, and as w is reduced, responsiveness to the changes drops. A bigger value of w reduces the memory requirements (smaller buffer capacity suffices to maintain the controller properties stated in the theorems), but at the same time, it results in larger values of the control signal (bigger transmission rate) in the initial phase,

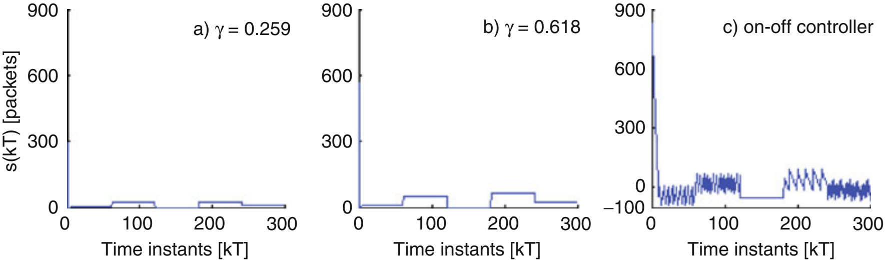


Fig. 6.6 Sliding variable: linear *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, and *c* nonlinear controller

which may be difficult to follow by low-rate transmitters. Therefore, similarly as in the case of a single flow, the weighting factor $w = 1$, which corresponds to the gain $\gamma = 0.618$, offers a fair trade-off between good system dynamics and smoothness of moderate transfer speeds when multiple connections are controlled by the bottleneck node simultaneously. Similarly as before, we will refer to this setting as the golden-ratio controller. The nonlinear controller offers the fastest response to the changes of networking conditions but produces highly oscillatory rate signal which can be difficult to reproduce by the sources.

If we compare the output variable (the packet queue length) resulting from the operation of the on-off controller in the single-source case illustrated in Fig. 5.10 with the one obtained for the multisource scenario analyzed here (Fig. 6.5), we can notice a reduced degree of oscillations. The less oscillatory response in the multisource network is the effect of smaller fluctuations in the cumulative packet incoming rate. The amplitude of oscillations is reduced due to the discrepancy in the connection RTTs. Consequently, if the packets delivered according to the $u_{\max}$ assignment distributed among the flows arrive at the node at different time instants due to nonequal RTTs, the resulting incoming rate rarely exhibits the large on-off transitions between 0 and 110 packets. However, even though the amplitude of the incoming rate variations is decreased, their frequency increases, which is the cause of small oscillations of the queue length even in the circumstances of constant steady-state bandwidth.

Figure 6.6 shows the evolution of the sliding variable. In the case of the SM controller with LQ optimal sliding plane applied (graphs a and b), the sliding variable immediately decreases from its original value $s(0) = \gamma y_{\mathrm{D}}$ to a relatively narrowband $s(\cdot) \in [0, \gamma c_n d_{\max} = \gamma d_{\max})$ and always remains in this band, which constitutes a clear evidence of a properly established quasi-sliding motion [1, 2, 5] in a discrete-time system. In the case of the on-off controller (4.29), the reaching phase is extended over several periods. However, once the system representative point approaches the vicinity of the sliding plane $s(kT) = 0$, it stays in the band around the plane for all subsequent time. Comparing the plot in Fig. 6.6c with the one representing the sliding variable for the continuous-time system shown in Fig. 4.18, we can notice oscillatory behavior caused by sampling with finite frequency in discrete-time case analyzed here.

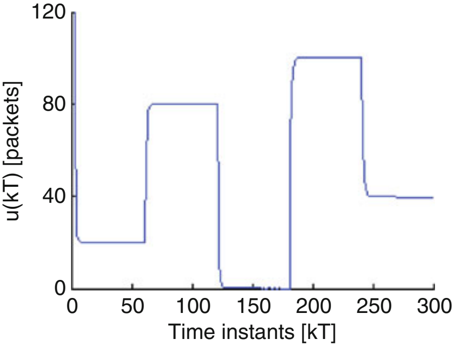


Fig. 6.7 Overall transmission rate established by the controller

Test 2. In the second simulation scenario, we investigate the performance of controller (6.27) in the case of nonequal resource allocation. We use the same network parameters as in Test 1 and apply the golden-ratio setting $\gamma = 0.618$. The following rate allocation policy is assumed: $\lambda_1 = 0.35$, $\lambda_2 = 0.2$, $\lambda_3 = 0.3$, and $\lambda_4 = 0.15$, which results in $a_3 = 0.35$, $a_7 = 0.2$, $a_8 = 0.3$, $a_{12} = 0.15$, $a_1 = a_2 = a_4 = a_5 = a_6 = a_9 = a_{10} = a_{11} = 0$, and the first row of matrix $\mathbf{A}$,

$$\left[a_{1j}\right] = \left[1 \quad 0.15 \quad 0 \quad 0 \quad 0 \quad 0.3 \quad 0.2 \quad 0 \quad 0 \quad 0 \quad 0.35 \quad 0 \quad 0\right].$$

Hence, the vector describing parameters of the sliding plane is determined as

$$\mathbf{c}^{\mathrm{T}} = \left[\gamma \; 0.15\gamma \; 0.15\gamma \; 0.15\gamma \; 0.15\gamma \; 0.45\gamma \; 0.65\gamma \; 0.65\gamma \; 0.65\gamma \; 0.65\gamma \; \gamma \; \gamma \; 1\right] c_n.$$

With such rate partitioning, the demand queue length calculated according to Theorem 6.2 is set as $830 > 827$ packets.

The test results are shown in Figs. 6.7–6.9: the transmission rate established by the controller in Fig. 6.7, the rate allocated for the sources in Fig. 6.8, and the packet queue length in the bottleneck node buffer in Fig. 6.9. For the sake of legibility, we display the magnified view of the transmission rate curves with the values limited to the range [0, 120 packets] in the case of the overall transmission rate established by the controller and [0, 40 packets] in the case of individual rate allocations. The limited value range concerns only the initial phase of transmission.

We can see from the plots that the overall transmission rate generated by the controller is nonnegative and smoothly follows the bandwidth changes. The individual flows receive the assumed fraction of the total rate, precisely as stated above. The queue length does not exceed the demand value and remains positive for $kT \geq 14T$. In consequence, the network operates at the maximum throughput.

Test 3. In the third simulation test, we compare the operation of the proposed SM controller (6.27), obtained in a modified optimization procedure, with the exact solution determined numerically by means of the Matlab built-in function for discrete-time optimization *dlqr*. Parameters of the optimal controller are determined as $r = 0.77y_{\mathrm{D}}$ and

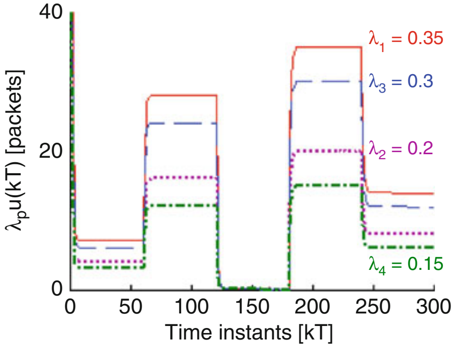


Fig. 6.8 Transmission rates allocated for the sources

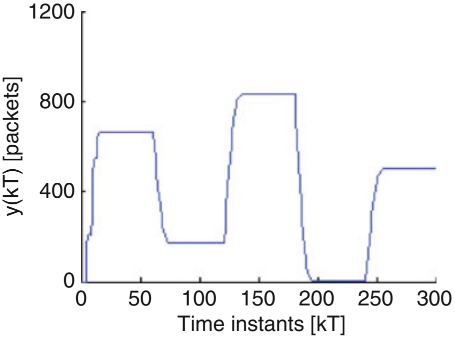


Fig. 6.9 Packet queue length

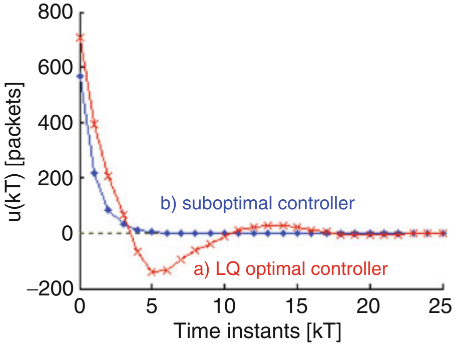


Fig. 6.10 Transmission rate comparison: *a* LQ optimal controller and *b* suboptimal controller

$$\mathbf{g} = \begin{bmatrix} 0.77 & 0.19 & 0.19 & 0.19 & 0.19 & 0.35 & 0.52 & 0.51 & 0.51 & 0.43 & 0.52 & 0.46 & 0.44 \end{bmatrix}.$$

We use the same model as in Test 1 and assume equal rate partitioning $\lambda_p = 1/4$ for $p \in \{1, 2, 3, 4\}$. The gain of the suboptimal controller (6.27) is set as 0.618 and the demand queue length read from Table 6.1 as 915 packets. For the sake of clarity, we assume $d(kT) \equiv 0$. The transmission rate established by the two controllers is shown in Fig. 6.10 and the resulting queue length in Fig. 6.11.

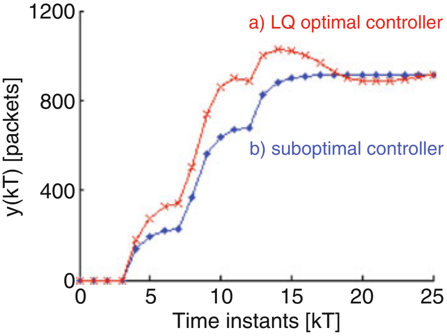


Fig. 6.11 Queue length comparison: *a* LQ optimal controller and *b* suboptimal controller

It can be seen from the graphs that the controller obtained as the exact solution to the LQ optimal control problem (curve a in Fig. 6.10) exhibits overshoots and oscillations. It is not suitable for application in the considered system since it generates a negative transmission rate. Function $u(\cdot)$ generated by the suboptimal strategy (curve b), in turn, although close to the exact solution, never falls below zero. Moreover, the value of the quadratic quality criterion (5.41) obtained for the suboptimal controller $J_{\text{subopt}} = 3.21 \cdot 10^6$ packets^2 exceeds only by 3% the optimal one $J_{\text{opt}} = 3.11 \cdot 10^6$ packets^2. Furthermore, when the proposed scheme is applied, both the rate and the queue length curves exponentially converge to the steady-state values without oscillations or overshoots. This is highly beneficial for enhancing the quality of service in data transmission systems which favor smooth transfer rates and small buffers over bursty traffic and large memory requirements.

Test 4. In this simulation, we evaluate the overall performance of the designed SM controller (6.27) in a network supporting numerous flows. The purpose of the test is to verify how the controller operates when the number of regulated connections increases. Thus, we intend to test the scalability of the proposed flow control strategy in multisource network implementation. We consider the situation when the bottleneck node allocates the transmission rate for 20 connections, whose RTTs are uniformly distributed between 50 and 240 ms. With $\gamma = 0.618$, the demand queue length necessary to guarantee full bandwidth utilization is adjusted to $1{,}615 > 1{,}612$ packets. The transfer rate established by the controller is shown in Fig. 6.12 and the buffer queue length in Fig. 6.13. For clarity, we limit the value range on the transmission rate graph to the interval [0, 120 packets].

We can notice small perturbations in the transmission rate graph as compared to the results obtained in Tests 1–3, which is attributed to the small additional delay caused by finite processing time of numerous control units arriving at the bottleneck node at the same moments of time (at multiples of the discretization period). Despite the increased input load (20 data streams) and signaling overhead (during the simulation interval of 3 s the node served 5,950 control units), the investigated algorithm maintains its properties stated in Theorems 6.1–6.3. Indeed,

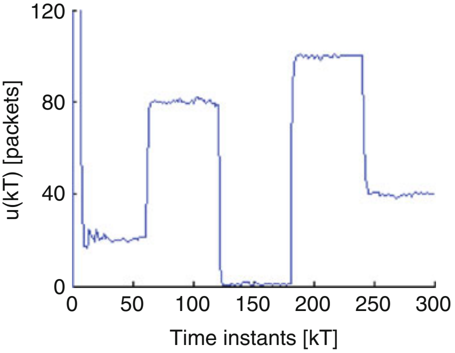


Fig. 6.12 Transmission rate – 20 connections

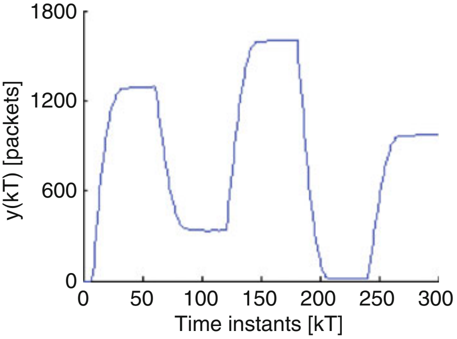


Fig. 6.13 Queue length – 20 connections

the transmission rate generated by the algorithm is nonnegative and bounded, and the queue length remains within the assigned buffer space and does not drop to zero, thus ensuring the maximum throughput.

Test 5. In the last simulation example considered in this section, we verify performance of the designed controller (6.27) when the available bandwidth at the bottleneck link exhibits high-frequency switching according to the stochastic pattern shown in Fig. 6.14. Function $d(\cdot)$ used in the simulations follows the normal distribution with mean $d_\mu = 50$ packets and standard deviation $d_\delta = 35$ packets, $D_{\text{norm}}(50, 35)$. We assume four flows and equal rate partitioning exactly as in Test 1. The controller gain is selected as $\gamma = 0.618$ (the golden-ratio setting) and two tests are run for different y_{D} levels. In the first simulation (curve a in the graphs), y_{D} is set exactly as specified in Table 6.1 – 915 packets – to make sure that all of the available bandwidth is used for the data transfer. Since the mean available bandwidth is much lower than the maximum one $d_{\max} = 100$ packets, in the second simulation (curve b in the graphs), the demand queue length is decreased to 460 packets (which corresponds to the value calculated according to (6.35) with $d_{\max}$ replaced by $d_\mu = 50$ packets).

The principal system variables obtained in the test are illustrated in Figs. 6.15–6.17: the transmission rate generated by the controller in Fig. 6.15, the buffer

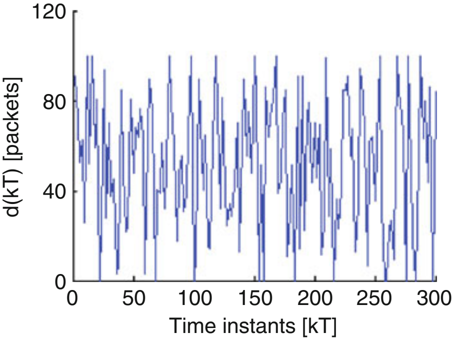


Fig. 6.14 Available bandwidth following normal distribution $D_{\mathrm{norm}}(5, 5)$

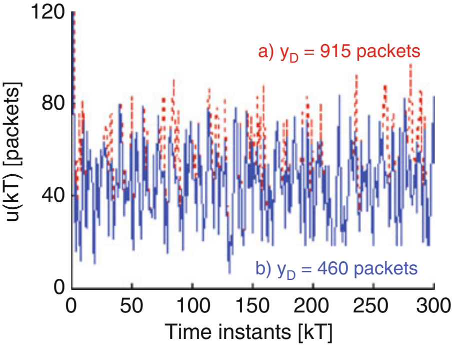


Fig. 6.15 Transmission rate: *a* $y_{\mathrm{D}} = 915$ packets and *b* $y_{\mathrm{D}} = 460$ packets

occupancy in Fig. 6.15, and the sliding variable in Fig. 6.17. We can see from the plots sketched in Fig. 6.15 (for better readability, the value range was limited to [0, 120 packets]) that irrespective of the demand queue length setting, the established transmission rate is nonnegative and upper-bounded. We can also notice that bandwidth variations are attenuated (compare with Fig. 6.14), which is highly favorable for improving the quality of service. Indeed, less oscillatory transfer rate is favored by the network users, as it allows for avoiding the annoying short-term pauses in the transmission, which may occur, for instance, during web page content retrieval. In addition, a smoother control signal propagating to the sources has a better chance of being reproduced exactly by the transmitters, which leads to better control of the network behavior.

The queue length evolution depicted in Fig. 6.16 shows that y never exceeds the demand value (in both cases a and b), which implies that the congestion does not occur and no packets need to be dropped. If y_{D} is chosen according to the indications given in Theorem 6.2 – 915 packets – then following the initial phase, y remains positive, which implies full usage of the available bandwidth. This is no longer the case, when the demand queue length is reduced to 460 packets. In that situation, certain opportunities for data transfer are wasted, and the overall bandwidth utilization in the 3 s simulation interval decreases to approximately 91%. However, the average and the maximum queue length in scenario b are much

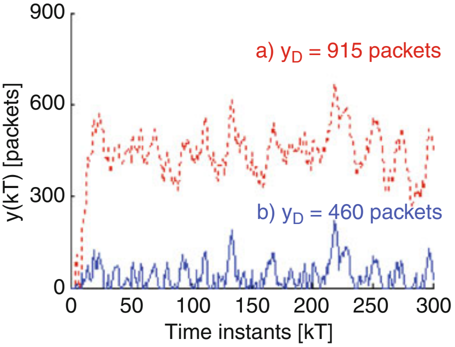


Fig. 6.16 Queue length: *a* $y_{\mathrm{D}} = 915$ packets and *b* $y_{\mathrm{D}} = 460$ packets

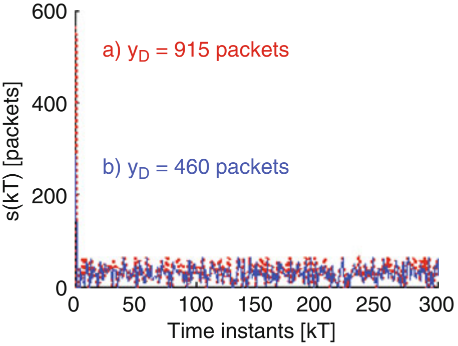


Fig. 6.17 Sliding variable: *a* $y_{\mathrm{D}} = 915$ packets and *b* $y_{\mathrm{D}} = 460$ packets

lower than in scenario a. As a result, decreasing y_{D} in the circumstances then d_{μ} is much smaller than $d_{\max}$ allows for significant buffer capacity savings (due to smaller maximum queue length) and reduction of mean transfer delay (due to smaller average queue length).

We can learn from Fig. 6.17 that the sliding variable decreases in one step from its original value $s(0) = \gamma y_{\mathrm{D}}$ to the band ([0, $\gamma c_n d_{\max} = \gamma d_{\max} < 100$ packets) and always remains in this band despite the presence of mismatched disturbance d. Hence, stability of the sliding motion is ensured during the whole transmission.

6.1.3 Methods for Constraining Excessive Initial Flow Rates

We have shown so far in this work that by applying dynamical optimization (with quadratic performance index) to obtain parameters of the sliding plane, we can provide fast reaction to the changes of networking conditions with smoothly varying transmission rates. However, the price that needs to be paid for good responsiveness and high link utilization is large value of transmission speed in the initial phase of the control process. Although in the case of controlling the rate of multiple sources one can modify the allocation so that bigger rate values are assigned to faster transmitters (by appropriate selection of λ_p coefficients), the limitation for the entire pool of connections may still be violated. Indeed, in the situation when the overall source rate capabilities are close to the maximum available bandwidth, the LQ optimal controller itself cannot guarantee both fast system dynamics and bounded transmission rates in the initial phase. In order to throttle excessive transfer speed and keep fast responsiveness, it is necessary to provide additional mechanisms. In this section, we investigate the use of three different methods which allow us to satisfy these partly contradictory objectives. The first method employs a *time-varying sliding plane* rather than the fixed one considered in Sect. 6.1.2. By appropriately adjusting the plane movement in the initial phase of the control process, the transmission rate can be enforced to remain within the bounds imposed by the system transfer capabilities. Afterwards, when the plane stops, the system dynamics is determined by the choice of the plane parameters, selected, for example, for LQ optimal or dead-beat scheme. The second investigated method employs a *reaching law*, which governs the way the system representative point approaches the (fixed) sliding plane. The reaching law and its parameters are chosen in such a way that the transmission rate constraint is never violated. The method using time-varying plane gives explicit control over the duration of the initial phase and implicit one over the maximum value of the obtained transmission rate. The reaching-law approach, in turn, allows one to impose a direct transmission speed constraint and regulate the duration of the initial phase implicitly. The third approach explored in this section uses a *saturation element* in the form of a transmission rate limiter to enforce direct bound on the maximum transfer speed generated by the controller. The third method, which gives implicit control over the duration of the initial period, can be considered the one most closely related to the actual practices of telecommunication software and hardware industry.

In the following sections, we describe the design procedures resulting in three nonlinear control laws. The most important properties of the obtained controllers relevant to controlling the flow of data in the network are then formulated and strictly proved. The controller performance is compared in a detailed numerical study presented after the analytical derivations.

6.1.3.1 Application of a Time-Varying Hyperplane

It has been shown in the previous section that by applying SM controller with LQ suboptimal sliding plane in the network serving multiple flows, one can eliminate packet losses and achieve full bandwidth utilization at the bottleneck node. However, the proposed controller generates large transmission rate in the initial phase of the control process which is required to quickly feed the bottleneck node buffer with enough packets to keep the node outgoing link fully occupied with sending the users' data. From the perspective of SM controller design, the high transmission rate at the beginning of the control process is the effect of enforcing the system representative point to cover large initial distance from the sliding plane.

Afterwards, in the sliding phase, the system representative point may leave the plane due to the external mismatched disturbance $d(\cdot)$, but the control signal necessary to bring it back onto the plane does not exceed the maximum available bandwidth $d_{\max}$ (which is typically much smaller than the initial rate value). One way to cope with high initial transfer rates is to apply time-varying plane instead of a fixed one which will move together with the system representative point in each step towards the final position. As the distance to be covered by the system representative point is reduced, the value of the control signal can be decreased to the limits allowed by the network transfer capabilities. In order to guarantee that the error is eliminated in finite time, the sliding plane should be chosen for dead-beat response, and in its final position, the plane should pass through the origin of the error state space. The rest of this section is devoted to the plane parameter selection and the choice of movement pattern so that desired system performance is achieved.

Controller Design

In order to reduce the excessive values of the controller command in the initial phase of the control process, we introduce a time-varying sliding hyperplane instead of the typically considered in the literature fixed one. The plane moves from the initial position, selected so that the system representative point belongs to the plane at $kT = 0$, towards the origin of the error state space with constant inclination. It stops moving after a predetermined interval $k_{\mathrm{VP}}T$ and remains fixed for the rest of data transfer. The design procedure consists of two phases. First, the controller action is chosen so that the system representative point is maintained on the plane as it moves through the error state space. Consequently, based on the *a priori* known plane displacement, at each instant kT, the controller generates a command which brings the representative point onto the plane at instant $(k+1)T$. In the second phase of the controller design, we select the plane parameters so that appropriately fast reaction to the changes of networking conditions is guaranteed.

Step 1. In this first part of the design, we decide on the sliding-plane dynamics. For any integer $k \geq 0$, the time-varying plane moving with constant inclination towards the final position $\mathbf{c}^{\mathrm{T}}\mathbf{e}(kT) = 0$ can be described by Equation (5.79). In order to ensure monotonic plane advancement towards the final position, *a priori* known function $f(\cdot)$ should satisfy conditions (5.80)–(5.82). These conditions can be summarized below:

- Choose the starting value of function $f(\cdot)$ such that initially the system representative point belongs to the plane, i.e., $f(0) = -\mathbf{c}^{\mathrm{T}}\mathbf{e}(0)$.
- Select the dynamics so that $f(\cdot)$ is strictly monotonic in the interval $[0, k_{\mathrm{VP}}T]$.
- Keep $f(kT) = 0$ for any $k \geq k_{\mathrm{VP}}$, which implies that the final plane position is maintained in the remaining part of the control process.

The examples of function $f(\cdot)$ given by (5.83), (5.84), and (5.85) are illustrated in Fig. 5.17, and their influence on the transmission efficiency is discussed in Example 5.2. It follows from the discussion given in that example that the plane displacement with constant velocity represented by (5.83) offers the highest transfer efficiency out of the choices (5.83), (5.84), and (5.85).

The decision about the duration of the plane movement determined by constant k_{VP} has a major impact on the system dynamics in the initial phase of the control process. With larger k_{VP} one obtains longer duration of the initial phase, which means that the plane will advance towards the final position in smaller steps. Consequently, by choosing larger k_{VP}, the control signal used to steer the system representative point onto the plane can be reduced. As a result, by appropriately selecting the duration of the plane movement, the input constraint can be met without degrading fast response in the latter phase of the transmission. This concludes the first step of the design.

In the second step of the design procedure, we will show how to choose parameters of the sliding plane so that dead-beat response to the changes of networking conditions is achieved.

Step 2. If we apply the system equation (5.7) in formula (5.79) describing the sliding plane at instant $(k+1)T$, we obtain discrete-time SM control law (5.88). With matrix $\mathbf{A}$ given by (6.8), the closed-loop system state matrix with this control applied, $\mathbf{A_{cl}} = [\mathbf{I}_n - \mathbf{b}(\mathbf{c}^{\mathrm{T}}\mathbf{b})^{-1}\mathbf{c}^{\mathrm{T}}]\mathbf{A}$, is determined as

$$\begin{bmatrix} 1 & a_{n-1} & a_{n-2} & a_{n-3} & \cdots & a_1 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ -\dfrac{c_1}{c_n} & -\dfrac{a_{n-1}c_1}{c_n} & -\dfrac{a_{n-2}c_1}{c_n} - \dfrac{c_2}{c_n} & -\dfrac{a_{n-3}c_1}{c_n} - \dfrac{c_3}{c_n} & \cdots & -\dfrac{a_1 c_1}{c_n} - \dfrac{c_{n-1}}{c_n} \end{bmatrix} \tag{6.39}$$

and its characteristic polynomial $\det(z\mathbf{I}_n - \mathbf{A_{cl}})$ as

$$z^n + \left(\frac{a_1 c_1}{c_n} + \frac{c_{n-1} - c_n}{c_n}\right) z^{n-1} + \cdots + \left(\frac{a_{n-2}c_1}{c_n} + \frac{c_2 - c_3}{c_n}\right) z^2 + \left(\frac{a_{n-1}c_1}{c_n} - \frac{c_2}{c_n}\right) z. \tag{6.40}$$

In order to obtain dead-beat controller, all the closed-loop poles should be at the origin of the error state space. Using (6.40), the roots of the characteristic equation $\det(z\mathbf{I}_n - \mathbf{A_{cl}}) = 0$ (the closed-loop poles) are all zero, if the following set of equations is satisfied:

$$\begin{aligned}
\frac{a_1 c_1}{c_n} + \frac{c_{n-1} - c_n}{c_n} &= 0, \\
\frac{a_2 c_1}{c_n} + \frac{c_{n-2} - c_{n-1}}{c_n} &= 0, \\
&\vdots \\
\frac{a_{n-2} c_1}{c_n} + \frac{c_2 - c_3}{c_n} &= 0, \\
\frac{a_{n-1} c_1}{c_n} - \frac{c_2}{c_n} &= 0.
\end{aligned} \tag{6.41}$$

This set of equations is solved recursively. First, we determine c_2 from the last identity in (6.41):

$$c_2 = a_{n-1} c_1. \tag{6.42}$$

Next, we substitute (6.42) into the equation $a_{n-2}c_1/c_n + (c_2 - c_3)/c_n = 0$ to get c_3 expressed in terms of the system parameters a_j and the first element of vector $\mathbf{c}$ as

$$c_3 = (a_{n-2} + a_{n-1})\, c_1. \tag{6.43}$$

Having determined c_3, we solve $a_{n-2}c_1/c_n + (c_3 - c_4)/c_n = 0$ for c_4, obtaining

$$c_4 = (a_{n-3} + a_{n-2} + a_{n-1})\, c_1. \tag{6.44}$$

If we continue the substitutions, we get in the last step $c_n = c_1$. Consequently, parameters $c_1, c_2, \ldots, c_n$ are determined as

$$\begin{aligned}
c_1 &= c_n, \\
c_2 &= a_{n-1} c_n, \\
c_j &= \big(a_{n-1} + a_{n-2} + \cdots + a_{n-j+1}\big)\, c_n \quad \text{for} \quad j = 3, 4, \ldots, n.
\end{aligned} \tag{6.45}$$

Finally, since $\sum_{j=1}^{n-1} a_j = 1$, we can represent the vector describing the sliding plane of a dead-beat controller in the following form:

$$\mathbf{c}^{\mathrm{T}} = \big[\, 1 \;\; a_{n-1} \;\; (a_{n-1} + a_{n-2}) \; \ldots \; (a_{n-1} + a_{n-2} + \cdots + a_2) \;\; 1 \,\big]\, c_n. \tag{6.46}$$

Substituting (6.8) and (6.46) into (5.88), we get the control law

$$u\,(kT) = y_{\mathrm{D}} - x_1\,(kT) - \sum_{j=2}^{n} \left(\sum_{i=1}^{j-1} a_{n-i} \right) x_j\,(kT) + \frac{1}{c_n} f\,[(k+1)\,T]. \tag{6.47}$$

Similarly as in (6.26), we substitute $x_1(kT) = y(kT)$ and the other state variables expressed in terms of the control signal generated in the previous $n - 1$ periods, $x_j(kT) = u[(k - n + j - 1)T]$ for $j = 2, 3, \ldots, n$, thus obtaining

$$u(kT) = y_{\mathrm{D}} - y(kT) - \sum_{j=2}^{n} \left(\sum_{i=1}^{j-1} a_{n-i} \right) u[(k - n + j - 1)T] + \frac{1}{c_n} f[(k+1)T]. \tag{6.48}$$

Finally, using the relation $a_j = \sum_{p:\mathrm{RTT}_p = jT} \lambda_p$, this control law can be rewritten as

$$u(kT) = y_{\mathrm{D}} - y(kT) - \sum_{p=1}^{m} \lambda_p \sum_{j=k-n_p}^{k-1} u(jT) + \frac{1}{c_n} f[(k+1)T]. \tag{6.49}$$

This concludes the design of SM controller with time-varying sliding plane for the considered multisource network. In the next section, we discuss several important properties of the obtained control law and substantiate our findings with formal proofs.

Properties of the Proposed Controller

The properties of the designed controller (6.49) will be given in a lemma and three theorems. The lemma and the first theorem show that the proposed nonlinear controller establishes the transmission rate which is always nonnegative and upper-bounded. Thus, the proposed scheme can be feasibly implemented in the network. The second theorem shows that the queue length in the bottleneck node buffer never exceeds the demand value. This means that if the buffer of capacity equal to at least y_{D} is assigned at that node, then packet losses originating from congestion will be eliminated. Finally, the third proposition specifies the minimum demand queue length value which is necessary to guarantee full bandwidth utilization at the bottleneck link.

We may notice that the transmission rate generated at the initial time $kT = 0$ equals

$$u(0) = y_{\mathrm{D}} + f(T)/c_n.$$

Afterwards, for $k > 0$, the control signal generated by the controller satisfies the relation given in the following lemma.

Lemma 6.4. *If controller (6.49) is applied to system (5.7) with matrix* $\mathbf{A}$ *defined by (6.8), and function* $f(\cdot)$ *satisfies conditions (5.80)–(5.82), then for any* $k > 0$,

$$u(kT) = h[(k-1)T] + \{f[(k+1)T] - f(kT)\}/c_n. \tag{6.50}$$

Proof. Substituting (6.7) into (6.49), we get

$$\begin{aligned}u\left(kT\right) &= y_{\mathrm{D}} - y\left(kT\right) - \sum_{p=1}^{m} \lambda_p \sum_{j=k-n_p}^{k-1} u\left(jT\right) + f\left[\left(k+1\right)T\right]/c_n \\ &= y_{\mathrm{D}} - \sum_{p=1}^{m} \sum_{j=0}^{k-n_p-1} \lambda_p u\left(jT\right) + \sum_{j=0}^{k-1} h\left(jT\right) \\ &\quad - \sum_{p=1}^{m} \lambda_p \sum_{j=k-n_p}^{k-1} u\left(jT\right) + f\left[\left(k+1\right)T\right]/c_n \\ &= y_{\mathrm{D}} - \sum_{p=1}^{m} \lambda_p \sum_{j=0}^{k-1} u\left(jT\right) + \sum_{i=0}^{k-1} h\left(jT\right) + f\left[\left(k+1\right)T\right]/c_n\,. \end{aligned} \tag{6.51}$$

Since the allocation policy satisfies the condition $\sum_{p=1}^{m} \lambda_p = 1$, formula (6.51) may be reduced to the following expression:

$$u\left(kT\right) = y_{\mathrm{D}} - \sum_{j=0}^{k-1} u\left(jT\right) + \sum_{i=0}^{k-1} h\left(jT\right) + f\left[\left(k+1\right)T\right]/c_n\,. \tag{6.52}$$

Consequently, for $k = 1$ we get $u(T) = h(0) + [f(2T) - f(T)]/c_n$. Taking similar steps as in (5.99) the transmission rate generated at instant $(l+1)T$ is shown to satisfy (6.50). This completes the induction proof. □

Theorem 6.5. *If controller (6.49) is applied to system (5.7) with matrix* **A** *defined by (6.8), and function f(·) satisfies conditions (5.80)–(5.82), then the transmission rate is always nonnegative and upper-bounded by* $y_D + d_{max}$.

Proof. At the initial time, the system representative point belongs to the moving sliding plane, and, according to (5.80), $f(0) = -\mathbf{c}^{\mathrm{T}}\mathbf{e}(0) = -c_n y_{\mathrm{D}}$. Since function $f(\cdot)$ is assumed to be strictly monotonic reaching zero at $kT = k_{\mathrm{VP}}T$, the identity $f(0) = -c_n y_{\mathrm{D}}$ implies that for any $k \in [0; k_{\mathrm{VP}})$, $f(kT)$ and c_n have opposite signs. Consequently, similarly as in the proof of Theorem 5.5, we can specify inequalities for the multisource network considered here identical to (5.101) and (5.102). On the other hand, it follows from the algorithm definition that $u(0) = y_{\mathrm{D}} + f(T)/c_n$. Hence, using (5.101), we get $u(0) \geq 0$ and $u(0) \leq y_{\mathrm{D}} \leq y_{\mathrm{D}} + d_{\max}$, which means that the theorem is satisfied for $k = 0$. For any $k > 0$, in turn, $u(kT)$ satisfies the relation given in Lemma 6.4. Repeating the algebraic manipulations given in the proof of Theorem 5.5, we get from Lemma 6.4

$$0 \leq u\left(kT\right) = h\left[\left(k-1\right)T\right] + \left\{f\left[\left(k+1\right)T\right] - f\left(kT\right)\right\}/c_n \leq d_{\max} + y_{\mathrm{D}}.$$

This conclusion ends the proof of Theorem 6.5. □

Theorem 6.6. *If controller (6.49) is applied to system (5.7) with matrix* **A** *defined by (6.8), and function $f(\cdot)$ satisfies conditions (5.80)–(5.82), then the queue length is always upper-bounded by its demand value y_D.*

Proof. It follows from the system initial conditions that the bottleneck node buffer is empty at $kT = 0$. Since the first nonzero rate is issued at $kT = 0$, then because of the feedback delay (ordered according to inequalities (6.2)), the first packets arrive at the bottleneck node at $kT = n_1 T$. This implies that $y(kT) = 0$ for $k \leq n_1$, and it suffices to demonstrate that the theorem holds for all $k > n_1$.

Using the fact that $u(0) = y_{\mathrm{D}} + f(T)/c_n$ and $\sum_{p=1}^{m} \lambda_p = 1$, we may present queue length (6.7) in the following form:

$$\begin{aligned} y(kT) &= \sum_{p=1}^{m} \lambda_p \sum_{j=0}^{k-n_p-1} u(jT) - \sum_{j=0}^{k-1} h(jT). \\ &= u(0) + \sum_{p=1}^{m} \lambda_p \sum_{j=1}^{k-n_p-1} u(jT) - \sum_{j=0}^{k-1} h(jT) \\ &= y_{\mathrm{D}} + \frac{1}{c_n} f(T) + \sum_{p=1}^{m} \lambda_p \sum_{j=1}^{k-n_p-1} u(jT) - \sum_{j=0}^{k-1} h(jT). \end{aligned} \tag{6.53}$$

On the other hand, we get from Lemma 6.4 that

$$\begin{aligned} y(kT) = y_{\mathrm{D}} &+ \frac{1}{c_n} f(T) + \sum_{p=1}^{m} \lambda_p \sum_{j=1}^{k-n_p-1} \left\{ h[(j-1)T] + \frac{f[(j+1)T] - f(jT)}{c_n} \right\} \\ &- \sum_{j=0}^{k-1} h(jT) = y_{\mathrm{D}} + \sum_{p=1}^{m} \lambda_p \sum_{j=1}^{k-n_p-1} h[(j-1)T] - \sum_{j=0}^{k-1} h(jT) \\ &+ \frac{1}{c_n} \left\{ f(T) + \sum_{p=1}^{m} \lambda_p \sum_{j=1}^{k-n_p-1} \{ f[(j+1)T] - f(jT) \} \right\} \\ &= y_{\mathrm{D}} + \sum_{p=1}^{m} \lambda_p \sum_{j=0}^{k-n_p-2} h(jT) - \sum_{p=1}^{m} \lambda_p \sum_{j=0}^{k-1} h(jT) + \frac{1}{c_n} \sum_{p=1}^{m} \lambda_p f\left[\left(k-n_p\right)T\right] \\ &= y_{\mathrm{D}} - \sum_{p=1}^{m} \lambda_p \sum_{j=k-n_p-1}^{k-1} h(jT) + \frac{1}{c_n} \sum_{p=1}^{m} \lambda_p f\left[\left(k-n_p\right)T\right]. \end{aligned} \tag{6.54}$$

Since the utilized bandwidth $h(\cdot)$ is always nonnegative, and $\forall k$, $f(kT)$, and c_n have opposite signs, we may conclude that the queue length given by (6.54) never exceeds the demand value y_{D}. This ends the proof. □

Theorem 6.7. *If controller (6.49) is applied to system (5.7) with matrix* **A** *defined by (6.8), and function f(·) satisfies conditions (5.80)–(5.82), and the demand queue length is selected according to the following inequality:*

$$y_{\mathrm{D}} > d_{\max}\left(\sum_{p=1}^{m}\lambda_p n_p + 1\right), \tag{6.55}$$

then for any $k \geq k_{VP} + n_m + 1$, the queue length is strictly positive.

Proof. Since we consider $k \geq k_{\mathrm{VP}} + n_m + 1$, for all n_p, we have $f[(k-n_p)T] = 0$. Consequently, since the utilized bandwidth represented by function $h(\cdot)$ is nonnegative and upper-bounded by $d_{\max}$, we get from (6.54) the following estimate of the queue length for $k \geq k_{\mathrm{VP}} + n_m + 1$:

$$\begin{aligned} y(kT) &= y_{\mathrm{D}} - \sum_{p=1}^{m}\lambda_p \sum_{j=k-n_p-1}^{k-1} h(jT) \geq y_{\mathrm{D}} - d_{\max}\sum_{p=1}^{m}\lambda_p\left(n_p+1\right) \\ &= y_{\mathrm{D}} - d_{\max}\left(\sum_{p=1}^{m}\lambda_p n_p + 1\right). \end{aligned} \tag{6.56}$$

Therefore, using assumption (6.55), we get $y(kT) > 0$ for all $k \geq k_{\mathrm{VP}} + n_m + 1$. This completes the proof. □

The fundamental properties of the presented strategy stated in the theorems have been verified in a simulation scenario described in Sect. 6.1.3.4.

6.1.3.2 Application of a Reaching Law

In order to constrain high initial transmission rates, we may modify the procedure of SM controller design to shape the way the system representative point approaches a (fixed) sliding plane. Therefore, instead of influencing the plane dynamics, as it was done in the previous section, the plane may be selected time invariant, yet the reaching phase is extended over several periods. As the representative point is required to cover a smaller distance from the plane in successive time intervals, the control signal magnitude (and as a result the transmission rate value) is reduced. In this way, we may ensure that the explicit input constraint (5.108), $0 \leq u(kT) \leq u_{\max}$ where $u_{\max} > d_{\max}$, is satisfied in the considered transmission system.

The way the system representative point approaches the sliding plane is governed by reaching law. Following the discussion presented in Sect. 5.1.3.1, in order to achieve high efficiency in transferring the users' data, it is convenient to apply the reaching law proposed by Golo and Milosavljević [6]. Further in the text, we will show that with properly adjusted reaching-law parameters, the resulting nonlinear SM controller can ensure zero loss rate and full bandwidth utilization in the multisource network without violating the explicit input constraint (5.108).

Controller Design

The reaching law proposed by Golo and Milosavljević [6] can be described by means of Eqs. (5.109) and (5.110), repeated for convenience below in a synthetic form:

$$s\left[(k+1)T\right] - s\left(kT\right) = -\min\left(\left|s\left(kT\right)\right|, \delta\right)\operatorname{sgn}\left[s\left(kT\right)\right]. \tag{6.57}$$

Function $\operatorname{sgn}(\cdot)$ in (6.57) is defined as previously, i.e., $\operatorname{sgn}(x) = -1$, if $x \le 0$, and $\operatorname{sgn}(x) = 1$, for $x > 0$. With law (6.57) applied, the system representative point is guaranteed to reach the hyperplane $s(kT) = \mathbf{c}^{\mathrm{T}}\mathbf{e}(kT) = 0$ monotonically in a finite number of steps in a way determined by the choice of parameter $\delta > 0$. We will show later in this section how this parameter should be selected to obtain high transmission efficiency in the analyzed multisource data transmission network.

Similarly as it was done in Sect. 5.1.3.1, we can present (6.57) in alternative way (5.111), where strictly monotonic function $f_{\mathrm{RL}}(\cdot)$ is defined as

$$\begin{cases} f_{\mathrm{RL}}\left[(k+1)T\right] = f_{\mathrm{RL}}\left(kT\right) + \delta\left(kT\right)\operatorname{sgn}\left[s\left(kT\right)\right] & \text{for} \quad k < k_{\mathrm{RL}}, k_{\mathrm{RL}} \in C_{+}, \\ f_{\mathrm{RL}}\left[(k+1)T\right] = 0 & \text{for} \quad k \ge k_{\mathrm{RL}}. \end{cases} \tag{6.58}$$

Note that in contrast to the single-flow scenario analyzed in Sect. 5.1.3.1, here we choose the reaching-law parameter $\delta(\cdot)$ to be an explicit function of time rather than a constant.

The controller design procedure is divided into two phases. First, the sliding-plane parameters are selected for a dead-beat scheme, and afterwards, the reaching-law parameter $\delta(\cdot)$ is chosen to satisfy the input constraint $0 \le u(kT) \le u_{\max}$.

Step 1. In the first part of the design, we choose the elements of vector $\mathbf{c}$ describing the inclination of the sliding-plane $s(kT) = \mathbf{c}^{\mathrm{T}}\mathbf{e}(kT) = 0$ passing through the origin of the error state space. If we opt for high responsiveness offered by a dead-beat scheme, then the closed-loop characteristic polynomial $\det(z\mathbf{I}_n - \mathbf{A}_{\mathbf{cl}})$ in the considered nth-order discrete-time system should be equal to z^n. Following similar steps as presented in (6.39)–(6.45), we arrive at vector $\mathbf{c}$ defined by (6.46) and the control law

$$u\left(kT\right) = y_{\mathrm{D}} - y\left(kT\right) - \sum_{p=1}^{m}\lambda_p \sum_{j=k-n_p}^{k-1} u\left(jT\right) + \frac{1}{c_n} f_{\mathrm{RL}}\left[(k+1)T\right]. \tag{6.59}$$

Step 2. In the second part of the design, we elaborate on the choice of function $\delta(\cdot)$ such that the resulting control signal never exceeds $u_{\max}$. We assume that $f_{\mathrm{RL}}(0) = -\mathbf{c}^{\mathrm{T}}\mathbf{e}(0) = -c_n y_{\mathrm{D}}$. Since $f_{\mathrm{RL}}(\cdot)$ is strictly monotonic, this assumption also implies that for any $k \in [0; k_{\mathrm{RL}}), f_{\mathrm{RL}}(\cdot)$ and c_n have opposite signs.

First, notice that since we assumed zero initial conditions, $u(0) = y_{\mathrm{D}} + f_{\mathrm{RL}}(T)/c_n$. Afterwards, for $kT \ge T$, the control signal satisfies the relation defined in the following lemma.

Lemma 6.8. *If controller (6.59) is applied to system (5.7) with matrix* **A** *defined by (6.8), and function $f_{RL}(\cdot)$ is selected according to (6.58), then for any $k \geq 1$, the generated transmission rate satisfies relation (5.114).*

Proof. Since $f_{\mathrm{RL}}(\cdot)$ is strictly monotonic, and it satisfies conditions (5.80)–(5.82), the lemma is true as a direct consequence of the reasoning presented in the proof of Lemma 6.4. This completes the proof of Lemma 6.8. □

Using (6.58), we can represent (5.114) in the following form:

$$\begin{cases} u\,(kT) = h\,[(k-1)\,T] + \delta\,(kT)\,\mathrm{sgn}[s\,(kT)]\,/c_n & \text{for} \quad k < k_{\mathrm{RL}}, \\ u\,(kT) = h\,[(k-1)\,T] & \text{for} \quad k \geq k_{\mathrm{RL}}. \end{cases} \tag{6.60}$$

It follows from (5.2) that for any k, the utilized bandwidth $0 \leq h(kT) \leq d_{\max}$. Therefore, the control signal specified by (6.60) is nonnegative and bounded by $d_{\max} < u_{\max}$ for any $k \geq k_{\mathrm{RL}}$. Since the buffer at the bottleneck node is initially empty, then, due to the delay, the first packets may be transferred at the node outgoing link once they arrive at $n_1 T$, and $h(k < n_1) = 0$. Afterwards, the utilized bandwidth $h(\cdot)$ changes according to the value of the available bandwidth $d(\cdot)$ and the packet arrival occurring in the order specified by the flow RTTs given by (6.2). Consequently,

$$\begin{aligned} &h\,(k < n_1) = 0, \\ &h\,(n_1 \leq k < n_2) \leq \min\left\{u_{\max} \sum_{j \leq n_1} a_j\,, d_{\max}\right\}, \\ &h\,(n_2 \leq k < n_3) \leq \min\left\{u_{\max} \sum_{j \leq n_2} a_j\,, d_{\max}\right\}, \ldots \end{aligned} \tag{6.61}$$

or more succinctly $h(kT) = \min\left\{u_{\max} \sum_{j \leq k} a_j\,, d_{\max}\right\}$ and $h[(k-1)T] \leq \min\left\{u_{\max} \sum_{j < k} a_j\,, d_{\max}\right\}$. Therefore, in order to ensure that the condition $0 \leq u(kT) \leq u_{\max}$ is satisfied for all $k < k_{\mathrm{RL}}$, on the basis of (6.60), we conclude that function $\delta(\cdot)$ should obey the following constraint:

$$0 < \delta\,(k < k_{\mathrm{RL}}) \leq |c_n| \left[u_{\max} - \min\left(u_{\max} \sum_{j<k} a_j\,, d_{\max}\right)\right], \tag{6.62}$$

which is calculated off-line using the information about the connections participating in the control process. This ends the design of the reaching law. The obtained controller calculates the transmission rate from (6.59) with function $f_{\mathrm{RL}}(\cdot)$ defined by (6.58) subject to constraint (6.62).

One may notice structural similarities between the controller designed using reaching-law approach (6.59) and the one employing time-varying sliding plane (6.49). The principal difference lies in the choice of function $f(\cdot)$ that governs the transmission rate allocation in the initial phase of the control process. According to the analysis presented in Example 5.3 (which can be directly extended to the multisource traffic scenario), selecting $f(kT)$ as defined in (6.58) with $\delta(kT) = |c_n| \left[u_{\max} - \min(u_{\max} \sum_{j<k} a_j, d_{\max})\right]$ will typically result in a larger number of packets sent in the initial phase of the control process than in the case of function $f(\cdot)$ given by any of (5.83), (5.84), and (5.85).

Properties of the Proposed Controller

It follows directly from Lemma 6.8 that the feasibility constraint of nonnegative transmission rates in the communication system regulated by controller (6.59) is fulfilled. The other properties of the proposed nonlinear controller related to the data transfer efficiency in the network will be formulated as two theorems. The first proposition shows that buffer overflow never occurs and no packet needs to be dropped due to possible bandwidth shortage at the outgoing link of the bottleneck node. In the second theorem, we specify the minimum value of the demand queue length which should be set in (6.59) to guarantee that the available bandwidth is entirely used for data transfer.

Theorem 6.9. *If controller (6.59) is applied to system (5.7) with matrix* $\mathbf{A}$ *defined by (6.8), and function* $f_{\mathrm{RL}}(\cdot)$ *is selected according to (6.58), then the queue length is always upper-bounded by its demand value* y_D.

Proof. The buffer at the bottleneck node is empty for any $kT \le \mathrm{RTT}_1 = n_1 T$. Hence, it suffices to show that the proposition is satisfied for any $k > n_1$. Using Lemma 6.8, the queue length in the bottleneck node buffer (6.7)

$$\begin{aligned} y\,(kT) &= u(0) + \sum_{p=1}^{m} \lambda_p \sum_{j=1}^{k-n_p-1} u\,(jT) - \sum_{j=0}^{k-1} h\,(jT) \\ &= u(0) + \sum_{p=1}^{m} \lambda_p \sum_{j=1}^{k-n_p-1} h\,[(j-1)\,T] - \sum_{j=0}^{k-1} h\,(jT) \\ &\quad + \sum_{p=1}^{m} \lambda_p \sum_{j=1}^{k-n_p-1} \{f_{\mathrm{RL}}\,[(j+1)\,T] - f_{\mathrm{RL}}\,(jT)\} \Big/ c_n. \end{aligned} \tag{6.63}$$

Since $u(0) = y_{\mathrm{D}} + f_{\mathrm{RL}}(T)/c_n$, we have

$$y\,(kT) = y_{\mathrm{D}} - \sum_{p=1}^{m} \lambda_p \sum_{j=k-n_p-1}^{k-1} h\,(jT) + \sum_{p=1}^{m} \lambda_p f_{\mathrm{RL}}\left[\left(k - n_p\right) T\right] \Big/ c_n. \tag{6.64}$$

Moreover, since the utilized bandwidth $h(\cdot)$ is always nonnegative, and for each k function $f(kT)$ and parameter c_n have opposite signs, $y(kT)$ given by (6.64) never exceeds the demand value y_{D}. This ends the proof. □

Theorem 6.10. *If controller (6.59) is applied to system (5.7) with matrix* $\mathbf{A}$ *defined by (6.8), and function* $f_{RL}(\cdot)$ *is selected according to (6.58), and the demand queue length according to (6.55), then for any* $k \geq k_{RL} + n_m + 1$*, the queue length is strictly positive.*

Proof. It follows from (6.58) that for $k > k_{\mathrm{RL}}$ function $f_{\mathrm{RL}}(kT) = 0$. Consequently, for $k \geq k_{\mathrm{RL}}$, controller (6.59) becomes equivalent to control law (6.49). Since both controllers incorporate the rate history in exactly the same way, then taking into account relation (6.55), the proposition is valid as a direct consequence of Theorem 6.7. This completes the proof. □

If we compare the properties of controller (6.59) stated in Theorems 6.9 and 6.10 with the ones demonstrated for controller (6.49) given in Theorems 6.6 and 6.7, we can notice a similar set of conditions required to guarantee the maximum throughput in the analyzed communication system. However, the described flow control strategies differ in the way the input constraint is satisfied. In the case of the controller with time-varying plane, this constraint is ensured by appropriately adjusting the duration of the plane movement. Thus, the input constraint is satisfied indirectly by manipulating the value of constant k_{VP}. On the other hand, the controller designed using the reaching-law approach allows for placing a direct limitation on the maximum transmission rate value allowed by the network. In the case of the reaching-law-based controller, the duration of the reaching phase is controlled indirectly.

It should be noted that in order to ensure that the generated transmission rate never exceeds the maximum allowed value $u_{\max}$, one can apply a direct rate limiter in the form of a saturation element. This third, direct method, which can be applied to limit excessive transmission rates without the need to throttle the controller dynamics, is given a detailed consideration in the next section.

6.1.3.3 Application of a Saturation Element

One of the methods of constraining transmission rate to a predefined range, favored by practitioners, is the application of a direct rate limiter. In software (or hardware) implementation of this method, before the information about the current transmission rate is incorporated in control units, the determined value is truncated to a given interval. In this way, direct control is exerted over the maximum transmission rate the sources will be permitted to use. From the control-theoretic perspective, the described method introduces a saturation nonlinearity into the controller operation.

Proposed Control Strategy

The transmission rate established by the algorithm at any time kT is determined according to the following equation:

$$u\left(kT\right) = \min\left\{\omega\left(kT\right), u_{\max}\right\}, \tag{6.65}$$

where function $\omega(kT)$ is defined as

$$\omega\left(kT\right) = y_{\mathrm{D}} - y\left(kT\right) - \sum_{p=1}^{m} \lambda_p \sum_{j=k-n_p}^{k-1} u\left(jT\right) \tag{6.66}$$

and $u_{\max} > d_{\max}$ is a positive constant denoting the maximum overall rate that can be assigned for the sources contending for the bandwidth at the bottleneck node output link.

Properties of the Proposed Strategy

In order to implement a congestion control algorithm in a data transmission network, one must ensure that the established transmission rate is always nonnegative and bounded so that the sources are not requested to inject packets into the network at negative or infinite rate. It follows directly from (6.65) that the transfer speed $u(\cdot)$ generated by the considered algorithm is upper-bounded by $u_{\max} > 0$. Definition (6.65) also implies that for any time instant $kT \geq 0$, we have $u(kT) \leq \omega(kT)$. Consequently, for feasible network implementation we need to guarantee that $u(kT) \geq 0$ is satisfied for any $kT \geq 0$. This is demonstrated in the following lemma.

Lemma 6.11. *If controller (6.65) with function $\omega(\cdot)$ defined by (6.66) is applied to system (5.7) with matrix $\mathbf{A}$ defined by (6.8), then for any $kT \geq 0$, the generated transmission rate is nonnegative.*

Proof. It follows from the assumed initial conditions

$$\begin{aligned} u\left(kT\right) &= 0 \;\text{ for }\; kT < 0, \\ y(0) &= 0, \end{aligned}$$

that $\omega(0) = y_{\mathrm{D}}$. Therefore, since $u_{\max} > 0$ and $y_{\mathrm{D}} > 0$, then using (6.65), we may estimate the flow rate value at the initial time as $u(0) = \min\{y_{\mathrm{D}}, u_{\max}\} > 0$. The condition $u_{\max} > 0$ also implies that if at any time instant $kT > 0$ the maximum transmission rate is established for the sources, then $u(kT) > 0$. Hence, what remains to be shown is that $u(kT) \geq 0$ for any $kT > 0$ in the circumstances when $u(kT) = \omega(kT)$. Making use of (6.4), we can represent the queue length dynamics in the following way:

$$y\,(kT) = y\,[(k-1)\,T] + \sum_{p=1}^{m} \lambda_p u\left[\left(k - n_p - 1\right) T\right] - h\,[(k-1)\,T]. \tag{6.67}$$

We analyze the situation when $u(kT) = \omega(kT)$. Then, using (6.67), we can write

$$\begin{aligned}
u\,(kT) = \omega\,(kT) &= y_{\mathrm{D}} - y\,(kT) - \sum_{p=1}^{m} \lambda_p \sum_{j=k-n_p}^{k-1} u\,(jT) \\
&= y_{\mathrm{D}} - y\,[(k-1)\,T] - \sum_{p=1}^{m} \lambda_p u\left[\left(k-n_p-1\right) T\right] \\
&\quad + h\,[(k-1)\,T] - \sum_{p=1}^{m} \lambda_p \sum_{j=k-n_p}^{k-1} u\,(jT) \\
&= y_{\mathrm{D}} - y\,[(k-1)\,T] - \sum_{p=1}^{m} \lambda_p \sum_{j=k-n_p-1}^{k-1} u\,(jT) + h\,[(k-1)\,T] \\
&= y_{\mathrm{D}} - y\,[(k-1)\,T] - \sum_{p=1}^{m} \lambda_p \sum_{j=k-n_p-1}^{k-2} u\,(jT) \\
&\quad - \sum_{p=1}^{m} \lambda_p u\,[(k-1)\,T] + h\,[(k-1)\,T].
\end{aligned}$$

Note that the first three terms in the last line of the preceding equation combine into $\omega[(k-1)T]$. On the other hand, since the rate partitioning strategy satisfies the condition $\sum_{p=1}^{m} \lambda_p = 1$, we have

$$\sum_{p=1}^{m} \lambda_p u\,[(k-1)\,T] = u\,[(k-1)\,T] \tag{6.68}$$

and

$$u\,(kT) = \omega\,[(k-1)\,T] - u\,[(k-1)\,T] + h\,[(k-1)\,T]. \tag{6.69}$$

Taking into account the fact that for any time instant kT, $u(kT) \leq \omega(kT)$, we get $\omega[(k-1)T] - u[(k-1)T] \geq 0$. Consequently, since the utilized bandwidth is always nonnegative, we obtain $u(kT) \geq h[(k-1)T] \geq 0$, which shows that $u(kT) \geq 0$ at any time instant $kT > 0$ when $u(kT) = \omega(kT)$. This conclusion ends the proof. □

The preceding discussion shows that by applying the control strategy (6.65), and (6.66), the explicit input constraint $0 \leq u(kT) \leq u_{\max}$ is always satisfied. In the further part of this section, we state properties of the described flow control algorithm and prove them analytically. First, we show that the packet queue length is upper-bounded, which means that if the corresponding buffer capacity is provided at the bottleneck node, then packet losses related to congestion are eliminated. Secondly, we specify the value of the demand queue length so that entire available bandwidth at the output link of the bottleneck node is used for data transfer and maximum throughput is achieved. Finally, we formulate a relation between the utilized bandwidth $h(\cdot)$ and the established control signal $u(\cdot)$.

Theorem 6.12. *If controller (6.65) with function $\omega(\cdot)$ defined by (6.66) is applied to system (5.7) with matrix* **A** *defined by (6.8), then the queue length in the bottleneck node buffer is always upper-bounded by its demand value y_{D}.*

Proof. It follows from Lemma 6.11 that the data transmission rate generated by controller (6.65), and (6.66) is always nonnegative. We know from *(6.65)* that $u(kT)$ never exceeds $\omega(kT)$. Therefore, for any time $kT \geq 0$, we can write

$$y_{\mathrm{D}} - y\,(kT) - \sum_{p=1}^{m} \lambda_p \sum_{j=k-n_p}^{k-1} u\,(jT) = \omega\,(kT) \geq u\,(kT) \geq 0. \tag{6.70}$$

Consequently, the queue length at instant kT is subject to the following constraint:

$$y\,(kT) \leq y_{\mathrm{D}} - \sum_{p=1}^{m} \lambda_p \sum_{j=k-n_p}^{k-1} u\,(jT). \tag{6.71}$$

Taking into account the inequality $u(kT) \geq 0$, we may conclude that the queue length never exceeds y_{D}. This ends the proof. □

Theorem 6.12 states that the packet queue length will not grow beyond its demand value y_{D}, which means that arbitrarily small buffer (but equal at least to y_{D}) suffices to store all the intermediate packets before they are forwarded by the bottleneck node towards destination. In this way, the packet losses due to lack of bandwidth and the resulting buffer overflow are eliminated. However, by assigning small values of the demand queue length, one may fail to ensure that the available bandwidth is efficiently used for data transfer. Therefore, it is desirable to specify conditions under which full bottleneck link bandwidth utilization is achieved. It turns out that in the considered system, it suffices to manipulate a single controller parameter – the demand queue length – to guarantee efficient bandwidth usage. The minimum value of the demand queue length required to meet this important objective is specified in the next theorem.

Theorem 6.13. *If controller (6.65) with function $\omega(\cdot)$ defined by (6.66) is applied to system (5.7) with matrix* **A** *defined by (6.8), the maximum rate $u_{\max} > d_{\max}$, and the demand value of the queue length y_D satisfies the following inequality:*

$$y_{\mathrm{D}} > u_{\max}\left(\sum_{p=1}^{m}\lambda_p n_p + 1\right), \tag{6.72}$$

then there exists a time instant k_0T,

$$k_0 < \frac{y_{\mathrm{D}} - u_{\max}}{u_{\max} - d_{\max}} + 1, \tag{6.73}$$

such that for any $kT > k_0T$, the queue length in the bottleneck node buffer is strictly positive.

Proof. Let us introduce an auxiliary function $\varphi(\cdot)$ defined as

$$\varphi(kT) = y(kT) + \sum_{p=1}^{m}\lambda_p \sum_{j=k-n_p}^{k-1} u(jT). \tag{6.74}$$

Substituting (6.7) into (6.74), we can express function $\varphi(kT)$ as a sum of input signals

$$\begin{aligned}\varphi(kT) &= y(kT) + \sum_{p=1}^{m}\lambda_p \sum_{j=k-n_p}^{k-1} u(jT)\\ &= \sum_{p=1}^{m}\sum_{j=0}^{k-n_p-1}\lambda_p u(jT) - \sum_{j=0}^{k-1} h(jT) + \sum_{p=1}^{m}\lambda_p \sum_{j=k-n_p}^{k-1} u(jT)\\ &= \sum_{p=1}^{m}\lambda_p \sum_{j=0}^{k-1} u(jT) - \sum_{j=0}^{k-1} h(jT).\end{aligned} \tag{6.75}$$

Since $\sum_{p=1}^{m}\lambda_p = 1$, we may simplify (6.75) in the following way:

$$\varphi(kT) = \sum_{j=0}^{k-1} u(jT) - \sum_{j=0}^{k-1} h(jT). \tag{6.76}$$

Considering assumption (6.72), we get

$$\varphi(0) = 0 < y_{\mathrm{D}} - u_{\max}\left(\sum_{p=1}^{m}\lambda_p n_p + 1\right) < y_{\mathrm{D}} - u_{\max}. \tag{6.77}$$

Moreover, if for some k the following inequality $\varphi(kT) < y_D - u_{\max}$ is satisfied, then

$$\omega(kT) = y_D - y(kT) - \sum_{p=1}^{m} \lambda_p \sum_{j=k-n_p}^{k-1} u(jT) = y_D - \varphi(kT) > u_{\max}, \quad (6.78)$$

which implies $u(kT) = u_{\max}$. Consequently, since $u_{\max} > d_{\max}$, we conclude that if $\varphi(kT) < y_D - u_{\max}$, then function $\varphi(\cdot)$ increases at least at the rate $u_{\max} - d_{\max}$. Thus, there exists such a finite time instant k_0T, when the following condition

$$\varphi(kT) \geq y_D - u_{\max} \quad (6.79)$$

is satisfied for the first time. Note that according to (6.76), the value of $\varphi(\cdot)$ does not depend on the number of sources. Consequently, the search for the latest time instant when inequality (6.79) can be satisfied for the first time and that it is satisfied for any $kT > k_0T$ proceeds as in the proof of Theorem 5.14. Using (6.79), one can see from relation (6.74) and the theorem assumption (6.72) that for any time $kT > k_0T$,

$$y(kT) \geq y_D - u_{\max} - \sum_{p=1}^{m} \lambda_p n_p u_{\max} = y_D - u_{\max}\left(\sum_{p=1}^{m} \lambda_p n_p + 1\right) > 0. \quad (6.80)$$

This completes the proof. □

Theorems 6.12 and 6.13 specify the conditions for achieving the maximum throughput in the considered communication network, in which the data flow rate of the sources is regulated by controller (6.65), and (6.66). In the next theorem, a relation between the generated transmission rate and the utilized bandwidth is formulated and strictly proved.

Theorem 6.14. *If controller (6.65) with function $\omega(\cdot)$ defined by (6.66) is applied to system (5.7) with matrix* **A** *defined by (6.8), the demand queue length $y_D > u_{max}$ and the maximum flow rate $u_{\max} > d_{\max}$, then there exists a nonnegative integer k_0 satisfying inequality (6.73) such that for any $k > k_0$, the following relation holds*

$$u(kT) = h[(k-1)T]. \quad (6.81)$$

Furthermore, when $y_D \leq u_{max}$, relation (6.81) is satisfied for any $k \geq 1$.

Proof. First, we will consider the situation when inequality $y_D \leq u_{\max}$ holds. We will demonstrate that in this case for any $k \geq 0$, we have $\omega(kT) \leq u_{\max}$, which using definition (6.65) implies $u(kT) = \omega(kT)$.

In order to prove that $\omega(kT) \leq u_{\max}$ for any $k \geq 0$, we apply the principle of the mathematical induction. At the beginning of the control process, $\omega(0) = y_D \leq u_{\max}$. Therefore, the considered inequality holds for $k = 0$. Let us assume that it is true for some integer $l > 0$, and we will show that it is also satisfied for $l+1$. Using (6.66) and (6.67), and taking into account that $u(kT) = \omega(kT)$, we get

$$\omega\left[(l+1)\,T\right] = \omega\left(lT\right) - u\left(lT\right) + h\left(lT\right) = h\left(lT\right) \leq d_{\max} < u_{\max}, \tag{6.82}$$

which ends the induction proof. Since in the analyzed situation $u(kT) = \omega(kT)$ for any $kT \geq 0$, we get from (6.69) $u(kT) = h[(k-1)T]$. This clearly shows that, if inequality $y_{\text{D}} \leq u_{\max}$ is satisfied, then (6.81) indeed holds for any positive integer k.

Now, let us consider the situation when $y_{\text{D}} > u_{\max}$. We know that if for some k the inequality $\omega(kT) < y_{\text{D}} - u_{\max}$ is satisfied, then it follows from (6.74) and the assumption $u_{\max} > d_{\max}$ that function $\omega(\cdot)$ increases at least at the rate $u_{\max} - d_{\max}$. Thus, there exists such a finite time instant k_0T, when condition (6.79) is satisfied for the first time and, as it has been proved, integer k_0 satisfies inequality (6.73). Condition (6.79) implies that for any $kT > k_0T$, $\omega(kT) = y_{\text{D}} - \varphi(kT) \leq u_{\max}$ and then $u(kT) = \omega(kT)$. Therefore, again, it immediately follows from (6.69) that relation (6.81) is indeed satisfied for any $k > k_0$. This conclusion ends the proof. □

We analyzed three methods of constraining high control signals: the application of a time-varying sliding plane, the use of a reaching law, and incorporating saturation nonlinearity. In the remainder of this section, the properties of the designed nonlinear controllers operating in the considered multisource communication network are compared in a series of simulation tests.

6.1.3.4 Simulation Results

To illustrate the controller properties, we apply a similar discrete-time network model as described in Sect. 6.1.2.3 with the discretization period set as $T = 10$ ms. Four connections ($m = 4$) are assumed to contend for the bandwidth at the output link of the bottleneck node. They are characterized by the delays: $\text{RTT}_1 = 3T$, $\text{RTT}_2 = 7T$, $\text{RTT}_3 = 8T$, and $\text{RTT}_4 = 12T$. The controller is assumed to treat the connections equally in the rate allocation. Therefore, the weights take the values: $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0.25$, which leads to matrix $\mathbf{A}$ given by (6.8) with the first row of the following form:

$$\begin{aligned}\left[a_{1j}\right] &= \left[a_{13} \quad a_{12} \quad a_{11} \quad a_{10} \quad a_9 \quad a_8 \quad a_7 \quad a_6 \quad a_5 \quad a_4 \quad a_3 \quad a_2 \quad a_1\right] \\ &= \left[1 \quad 0.25 \quad 0 \quad 0 \quad 0 \quad 0.25 \quad 0.25 \quad 0 \quad 0 \quad 0 \quad 0.25 \quad 0 \quad 0\right].\end{aligned}$$

The available bandwidth at the bottleneck link is limited by $d_{\max} = 100$ packets per period. It is also assumed that the biggest rate the controller may establish for the connections cannot exceed $u_{\max} = 150$ packets.

We compare the three methods of constraining the transmission rate to the interval $[0, u_{\max}]$ presented in Sect. 6.1.3: the application of time-varying sliding plane (controller (6.49)), reaching-law-based design (controller (6.59)), and the use of direct transmission rate limiter (controller (6.65)). In order to provide a baseline for comparison, we also run the tests for the (linear) LQ suboptimal controller (6.27) with the gain adjusted such that the rate does not exceed $u_{\max}$, i.e., the gain needs to fulfill the inequality $\gamma \leq u_{\max}/y_{\mathrm{D}}$. We run two series of simulations: one for the piecewise constant bandwidth pattern with abrupt transitions illustrated in Fig. 6.2 and another series for the stochastic signal depicted in Fig. 6.14.

Test 1. We verify the controller performance in the case when the available bandwidth at the bottleneck link evolves as shown in Fig. 6.2. The parameters of each controller are chosen so that the overall transmission rate is limited by $u_{\max} = 150$ packets. At the same time, it is desired to drive the network into the state of maximum throughput that would be ensured irrespective of the actual bandwidth variations (*a priori* unknown to the controllers). Consequently, the gain of the LQ suboptimal controller (curve a in the graphs) is selected as $\gamma = 0.066$, and the demand queue length for this controller is adjusted according to Theorem 6.2 as $2{,}255 > 2{,}250$ packets. The gain selection $\gamma = 0.066$ corresponds to the output weighting factor in the quadratic cost functional $w = 0.0047$ (refer to (6.23)). Parameters of time-invariant LQ suboptimal plane (with $c_n = 1$) are as follows:

$$\mathbf{c}^{\mathrm{T}} = \begin{bmatrix} \gamma & 0.25\gamma & 0.25\gamma & 0.25\gamma & 0.25\gamma & 0.5\gamma & 0.75\gamma & 0.75\gamma & 0.75\gamma & 0.75\gamma & \gamma & \gamma & 1 \end{bmatrix}.$$

In the case of controller (6.49) (curve b in the graphs), which employs a time-varying sliding plane, we need to decide on both the plane dynamics and inclination (determined by the parameters $c_1, \ldots, c_n$). Parameters of the plane are selected for a dead-beat scheme as

$$\mathbf{c}^{\mathrm{T}} = \begin{bmatrix} 1 & 0.25 & 0.25 & 0.25 & 0.25 & 0.5 & 0.75 & 0.75 & 0.75 & 0.75 & 1 & 1 & 1 \end{bmatrix}. \tag{6.83}$$

It is assumed that the plane moves with constant inclination towards the final position $\mathbf{c}^{\mathrm{T}}\mathbf{e}(kT) = 0$. Note that in its final position, the plane passes through the origin of the error state space and the controller is capable of reducing the error to zero in finite time. The plane movement is described by a linear function $f_{\mathrm{VP}}(\cdot)$ defined in the following way:

$$f_{\mathrm{VP}}(kT) = \begin{cases} -\frac{k-k_{\mathrm{VP}}}{k_{\mathrm{VP}}}\mathbf{c}^{\mathrm{T}}\mathbf{e}(0) = -\frac{k-k_{\mathrm{VP}}}{k_{\mathrm{VP}}}y_{\mathrm{D}} & \text{for} \quad k = 0, 1, \ldots, k_{\mathrm{VP}}; \\ 0 & \text{for} \quad k > k_{\mathrm{VP}}; \end{cases}$$

where the terminal condition is selected to ensure $u(kT) \leq u_{\max}$ as $k_{\mathrm{VP}} = 8$. The demand queue length is chosen according to Theorem 6.7 such that full bandwidth usage is ensured. We set $y_{\mathrm{D}} = 855 > 850$ packets.

In the case of controller (6.59) designed using reaching-law approach, the key point is an appropriate selection of parameter $\delta(\cdot)$ that governs the way the system representative point approaches the fixed sliding plane. To quickly bring the system

Table 6.2 Demand queue length selection

Controller	y_D [packets]
Linear with LQ suboptimal plane	2,255 > 2,250
Nonlinear with time-varying plane	855 > 850
Nonlinear employing reaching law	855 > 850
Nonlinear with saturation element	1,280 > 1,275

into the region of maximum throughput, we opt for the fastest point movement towards the plane, and, once the plane is reached, we expect rapid error convergence to zero. Therefore, we choose the plane parameters as in (6.83) and $\delta(\cdot)$ as the biggest value permitted by the input constraint at the right margin of inequality (6.62). We select $\delta(\cdot)$ according to

$$\delta\left(k < k_{\mathrm{RL}}\right) = u_{\max} - \min\left(u_{\max}\sum_{j<k} a_j, d_{\max}\right)$$

as

$$\begin{aligned}
\delta\left(0 \le k < 4\right) &= u_{\max} - \min\left(0, d_{\max}\right) = 150 - 0 = 150 \text{ packets,}\\
\delta\left(4 \le k < 8\right) &= u_{\max} - \min\left(u_{\max}a_3, d_{\max}\right) = 150 - 150 \cdot 0.25 = 112.5 \text{ packets,}\\
\delta(8) &= u_{\max} - \min\left[u_{\max}\left(a_3 + a_7\right), d_{\max}\right] = 150 - 150\left(0.25 + 0.25\right)\\
&= 75 \text{ packets,}\\
\delta(9) &= u_{\max} - \min\left[u_{\max}\left(a_3 + a_7 + a_8\right), d_{\max}\right]\\
&= 150 - \min\left(112.5, 100\right) = 50 \text{ packets.}\\
\delta\left(9 < k < k_{\mathrm{RL}}\right) &= 50 \text{ packets.}
\end{aligned}$$

The demand queue length for the reaching-law-based controller is chosen as suggested by Theorem 6.10 at the level exceeding 850 packets. We select 855 packets.

The saturation limit of controller (6.65) is set directly as $u_{\max} = 150$ packets. Then, the demand queue length resulting in the maximally efficient bandwidth usage is chosen according to the guidelines of Theorem 6.13 as 1,280 > 1,275 packets. For convenience, the reference values (which also constitute the minimum buffer capacity required at the bottleneck node) are grouped in Table 6.2.

It is evident from the plots presented in Fig. 6.18 that the rate established by each controller is nonnegative and never exceeds the upper bound of 150 packets, precisely as permitted by the network. We can also see from these graphs that all three nonlinear controllers (6.49), (6.59), and (6.65) provide faster reaction to the bandwidth changes than the linear one (6.27). On the other hand, it follows from Fig. 6.19 that each controller guarantees that the queue length does not grow beyond the demand value given in Table 6.2. Consequently, the buffer assignment of y_D

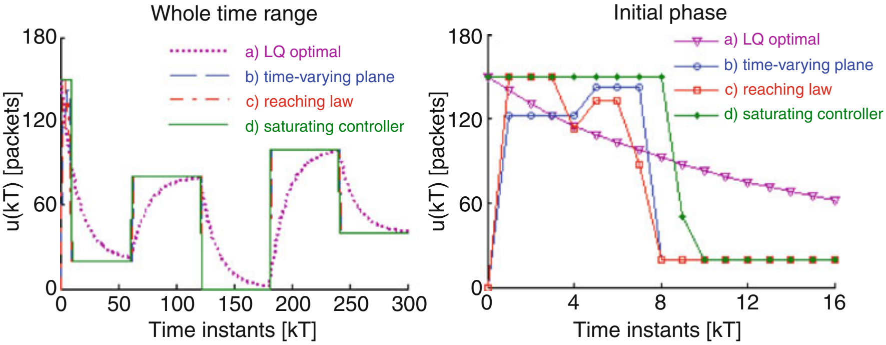
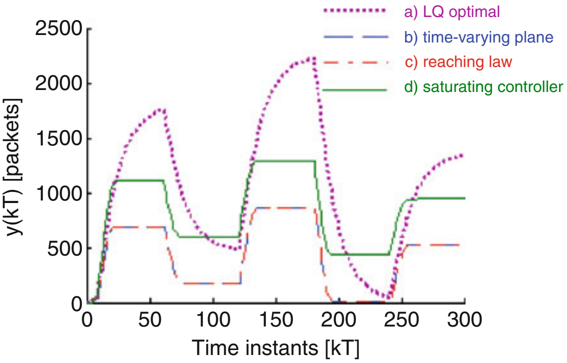


Fig. 6.18 Transmission rate: *a* linear controller (6.27) and *b–d* nonlinear controllers (6.49), (6.59), and (6.65)

Fig. 6.19 Queue length: *a* linear controller (6.27) and *b–d* nonlinear controllers (6.49), (6.59), and (6.65)



ensures loss-free transmission. We can also learn from the plots depicted in Fig. 6.19 that the nonlinear controllers impose smaller buffer capacity requirements than the linear one while still ensuring loss-free and maximally efficient data transfer. The larger average queue length in the case of the linear controller also leads to longer time necessary for draining the queue. Thus, if the average session delay is of major importance, the linear scheme will provide the worst quality of service in the analyzed scenario with rate constraints.

The linear controller guarantees that the plane is attained in a single step, and, afterwards, the system representative point is maintained in a direct vicinity of the plane [0, $\gamma c_n d_{\max} = \gamma d_{\max} = 6.6$ packets). In the case of the nonlinear controllers, with $c_n = 1$, the band around the sliding plane extends to [0, 100 packets). Thus, the faster reaction to the bandwidth changes provided by the nonlinear controllers comes at a price of larger band in the sliding phase. The evolution of the sliding variable in the initial phase is illustrated in Fig. 6.20.

It follows from the preceding discussion that the analyzed nonlinear controllers offer faster dynamics and more efficient memory usage than the linear scheme. It is instructive to compare the proposed controllers also with respect to the quadratic

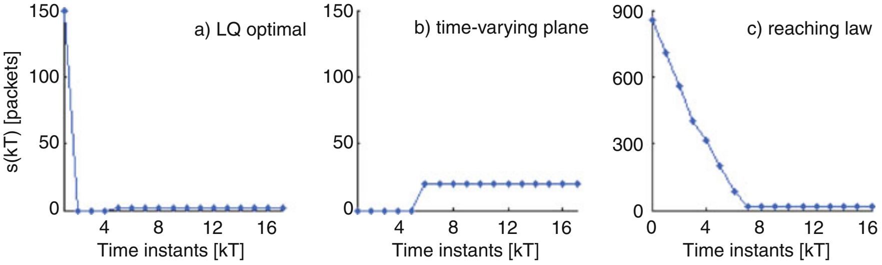


Fig. 6.20 Sliding variable: *a* controller (6.27) with fixed plane, *b* controller (6.49) with variable plane, and *c* reaching-law-based controller (6.59)

Table 6.3 Controller comparison based on performance index (5.41)

Controller	J_2 $(\cdot 10^5)$
Linear with LQ suboptimal plane	2.66
Nonlinear with time-varying plane	0.70
Nonlinear employing reaching law	0.72
Nonlinear with saturation element	1.32

quality criterion. In Table 6.3, we list the values of J_2 (with $w = 0.0047$) obtained for the disturbance-free case $d(\cdot) \equiv 0$. The analysis of the data given in Table 6.3 reveals that the nonlinear controllers perform better according to the quadratic quality criterion than controller (6.27) (which attempts to minimize this criterion!). This somewhat surprising result is easily explained if we compare the demand queue length values given in Table 6.2. In the case of the nonlinear controllers, a much smaller value suffices to ensure full bandwidth utilization while preserving the input constraint than in the case of linear controller (6.27). Therefore, the designed nonlinear controllers require less control effort to follow the reference value, and the output tracking component $y_{\mathrm{D}} - y(kT)$ in (5.41) is more quickly reduced to zero.

Test 2. In the second scenario, we compare performance of the LQ suboptimal controller with the nonlinear one incorporating the saturation element in the circumstances when the available bandwidth at the bottleneck link undergoes rapid fluctuations (since the nonlinear controllers do not apply rate smoothening their response to highly variable bandwidth is similar). Function $d(\cdot)$ following the normal distribution with mean $d_\mu = 50$ packets and standard deviation $d_\delta = 35$ packets is depicted in Fig. 6.14. The controller parameters are adjusted as in Test 1.

The generated transmission rate is shown in Fig. 6.21 and the resulting packet queue length in Fig. 6.22. We can see from the plots depicted in Fig. 6.21 that the input constraint is satisfied. On the other hand, it follows from Fig. 6.22 that the queue length never exceeds the buffer size limitations, which means that packet losses are eliminated. The queue never drops to zero following the initial phase, and, hence, the conditions of maximum throughput are achieved. The queue length variations are similar between the analyzed controllers. However, the transmission rate established by the nonlinear controller undergoes larger fluctuations than the

Fig. 6.21 Transmission rate: *a* linear controller (6.27) and *b* nonlinear controller (6.65)

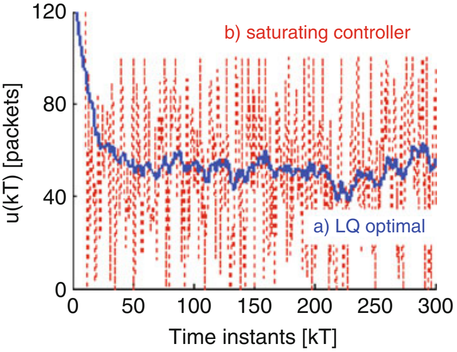


Fig. 6.22 Queue length: *a* linear controller (6.27) and *b* nonlinear controller (6.65)

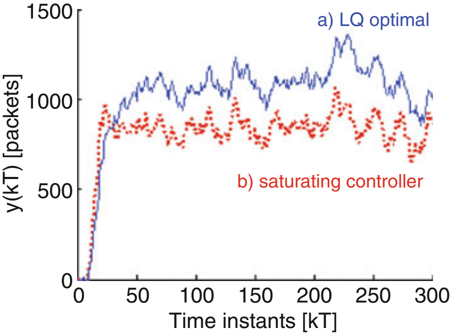


Table 6.4 Signal statistics

	$u(\cdot)$			$y(\cdot)$		
Controller	μ	δ	δ/μ	μ	δ	δ/μ
Linear with LQ suboptimal plane	53	12	0.23	1,015	254	0.25
Nonlinear with saturation element	53	35	0.66	803	179	0.22

μ mean [packets], δ standard deviation [packets], δ/μ coefficient of variation

one generated by linear controller, and thus, it is more difficult to follow by the transmitters. The signal statistics summarized in Table 6.4 show that in the case of the nonlinear controller, rapid bandwidth oscillations are translated to the transmission rate. The LQ suboptimal control law provides both smaller standard deviation and smaller coefficient of variation of the generated rate signal $u(\cdot)$.

6.1.4 Variable Source Number and Time-Varying Rate Allocation

The control strategies considered so far in this chapter eliminate the risk of losing data and guarantee the maximum available throughput in the network modeled as a discrete-time system with multiple input-output delays. However, these favorable properties are demonstrated under the assumption that the rate distribution weights and the overall number of flows are determined in the connection setup phase and remain unchanged during the whole control process. In this section, we present an enhanced version of the strategy introduced in Sect. 6.1.3.3 (proportional controller with saturation), which explicitly takes into account time-varying rate allocation patterns and variable source number.

6.1.4.1 Variable Rate Allocation

Let us consider the set of M connections, $M = \text{const}$, characterized by round-trip times $\text{RTT}_1 = n_1 T$, $\text{RTT}_2 = n_2 T, \ldots$, $\text{RTT}_M = n_M T$, where $n_1, n_2, \ldots, n_M$ are positive integers and T is the discretization period. Without loss of generality, we may order these connections according to their RTTs in the following way:

$$0 < n_1 \leq n_2 \leq \cdots \leq n_M. \tag{6.84}$$

Any of the connections characterized by (6.84) may send data through the bottleneck node at some moment of time $kT > 0$. A particular connection is considered active if the node takes it into account in the allocation of transmission rate. The number of active connections at any time instant kT, denoted by $m(kT)$, satisfies the condition $m(kT) \leq M$. Note that here the number of active connections is an explicit function of time $m(\cdot)$ rather than a constant as was considered so far in this chapter.

Let us denote the fraction of the overall transmission rate allocated at instant kT to connection p by $\lambda_p(kT)$, where $\lambda_p(kT)$ is a real number from the interval [0,1] satisfying $\sum_{p:\text{active}} \lambda_p\,(kT) \leq 1$. Each active connection is allocated a positive fraction of the overall transmission rate $u(\cdot)$. For instance, in the case of equal rate distribution, we will have $\lambda_p(kT) = 1/m(kT)$. In the circumstances, when $\lambda_p(kT) = 1$, only one flow is active, while $\lambda_p(kT) = 0$ implies that the connection characterized by delay $n_p T$ is considered turned-off. Since for inactive connections $\lambda_p(kT) = 0$, we may write

$$\underset{k\geq 0}{\forall} \sum_{p:\text{active}} \lambda_p\,(kT) = \sum_{p=1}^{M} \lambda_p\,(kT) \leq 1. \tag{6.85}$$

In the case of the controlling node being a single bottleneck for the considered set of connections, we will have $\sum_{p=1}^{M} \lambda_p\,(kT) = 1$, as the rate established by the controller is the lowest one on the data route. In the multibottleneck scenario, in turn, we will typically have $\sum_{p=1}^{M} \lambda_p\,(kT) < 1$, as some connections may be throttled at other points in the network, and the controlling node will be forced to allocate smaller rate than the generated one for those connections.

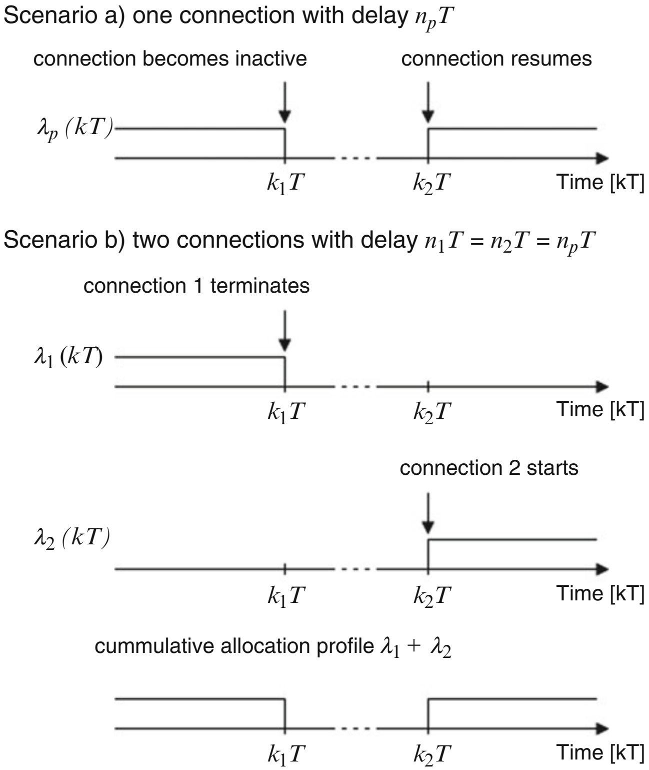


Fig. 6.23 Allocation profile: *a* single connection with different source activity periods and *b* two connections with the same delay and different activity periods

Note that the actual data origin of packet streams is of no importance from the point of view of the rate allocation strategy represented by the set of functions $\lambda_p(\cdot)$. The rate allocation function $\lambda_p(\cdot)$ is related only to the connections with RTT equal to n_pT passing through the controlling node. Thus, if a connection characterized by delay n_pT becomes temporarily inactive, say at k_1T, and resumes at some later time instant $k_2T > k_1T$, we have the allocation profile $\lambda_p(k < k_1) > 0$, $\lambda_p(k_1 \leq k < k_2) = 0$, and $\lambda_p(k \geq k_2) > 0$. The same cumulative allocation profile is obtained if one connection is terminated at k_1T, and another connection is started at k_2T (possibly at a different source), provided that both connections experience the same delay n_pT. This situation is illustrated in Fig. 6.23. Consequently, using the described framework, one can represent arbitrary (feasible) rate allocation strategy and arbitrary source activity pattern.

With the general, time-varying rate allocation policy described above, the queue length dynamics is given by the following relation:

$$y\left[(k+1)\,T\right] = y\,(kT) + \sum_{p=1}^{M} \lambda_p \left(kT - \mathrm{RTT}_p\right) u\left(kT - \mathrm{RTT}_p\right) - h\,(kT)\,. \tag{6.86}$$

Assuming the zero initial conditions, i.e., $y(0) = 0$ and $\forall p$, $\lambda_p(kT) = 0$ for $k < 0$, we get a closed-form expression for the queue length at arbitrary time instant *kT:*

$$y\,(kT) = \sum_{p=1}^{M} \sum_{j=0}^{k-1} \lambda_p \left(jT - \mathrm{RTT}_p\right) u\left(jT - \mathrm{RTT}_p\right) - \sum_{j=0}^{k-1} h\,(jT). \tag{6.87}$$

After applying the definition $\mathrm{RTT}_p = n_p T$, relation (6.87) can be rewritten as

$$\begin{aligned} y\,(kT) &= \sum_{p=1}^{M} \sum_{j=0}^{k-1} \lambda_p \left(jT - n_p T\right) u\left(jT - n_p T\right) - \sum_{j=0}^{k-1} h\,(jT) \\ &= \sum_{p=1}^{M} \sum_{j=-n_p}^{k-n_p-1} \lambda_p\,(jT)\,u\,(jT) - \sum_{j=0}^{k-1} h\,(jT) \\ &= \sum_{p=1}^{M} \sum_{j=0}^{k-n_p-1} \lambda_p\,(jT)\,u\,(jT) - \sum_{j=0}^{k-1} h\,(jT), \end{aligned} \tag{6.88}$$

which is convenient for the closed-loop property analysis conducted in a latter part of this section.

6.1.4.2 Proposed Control Strategy

To control the flow of data in the network with variable resource allocation, we propose to apply the proportional controller combined with saturation nonlinearity which was presented in Sect. 6.1.3.3. However, here we add another limit to the saturation element so that the rate is explicitly restricted to the interval $[0, u_{\max}]$.

The overall transmission rate for the connections passing through the controlling node is determined from the following relation:

$$u\,(kT) = \begin{cases} 0, & \text{if} \quad \omega_\lambda\,(kT) < 0, \\ \omega_\lambda\,(kT)\,, & \text{if} \quad 0 \le \omega_\lambda\,(kT) \le u_{\max}, \\ u_{\max}, & \text{if} \quad \omega_\lambda\,(kT) > u_{\max}, \end{cases} \tag{6.89}$$

where $u_{\max} > d_{\max}$, and function $\omega_\lambda(\cdot)$ is defined as

$$\omega_\lambda(kT) = \gamma \left[y_D - y(kT) - \sum_{p=1}^{m} \sum_{j=k-n_p}^{k-1} \lambda_p(jT)\, u(jT) \right] \tag{6.90}$$

and the gain $\gamma \in (0, 1]$.

It follows directly from (6.89) that the proposed control algorithm satisfies the input constraint $0 \le u(kT) \le u_{\max}$. The controller properties related to handling the flow of data in the considered network are discussed and strictly proved in the next section.

6.1.4.3 Properties of the Proposed Strategy

All the controllers presented so far in this work guarantee lossless transmission and good network efficiency when the protocol designer follows the rules of parameter adjustment specified in the appropriate theorems. Below, we formulate a set of conditions which allow the controller considered in this section to eliminate packet losses (occurring due to congestion) and obtain full bandwidth utilization in the situation when the rate distribution policy and number of active connections change with time during the transmission.

Theorem 6.15. *If controller (6.89) with function $\omega_\lambda(\cdot)$ defined by (6.90) is applied to the system with variable resource allocation, then for any $k \ge 0$, the queue length at the bottleneck node does not exceed y_{max}, where*

$$y_{\max} = y_D + u_{\max}. \tag{6.91}$$

Proof. It follows from the algorithm definition and the system initial conditions that the buffer at the bottleneck node is empty for any $k < n_1$. Consequently, it is sufficient to show that the proposition holds for all $k \ge n_1$. Let us consider some integer $l \ge n_1$ and the value of function $\omega_\lambda(\cdot)$ at time instant lT. Two cases ought to be analyzed: the situation when $\omega_\lambda(lT) \ge 0$ and the circumstances when $\omega_\lambda(lT) < 0$.

Case 1. Investigating the case when $\omega_\lambda(lT) \ge 0$, we get from the definition of $\omega_\lambda(\cdot)$,

$$\omega_\lambda(lT) = \gamma \left[y_D - y(lT) - \sum_{p=1}^{M} \sum_{j=l-n_p}^{l-1} \lambda_p(jT)\, u(jT) \right] \ge 0, \tag{6.92}$$

that

$$y(lT) \le y_D - \sum_{p=1}^{M} \sum_{j=l-n_p}^{l-1} \lambda_p(jT)\, u(jT). \tag{6.93}$$

On the other hand, it follows from (6.89) that $u(\cdot)$ is always nonnegative, which implies $y(lT) \leq y_{\mathrm{D}}$. This ends the first part of the proof.

Case 2. In the second part of the proof, we analyze the situation when $\omega_{\lambda}(lT) < 0$. First, we find the last instant $l_1 T < lT$ when $\omega_{\lambda}(\cdot)$ was nonnegative. According to (6.90) $\omega_{\lambda}(0) = \gamma y_{\mathrm{D}} > 0$ since both γ and y_{D} are positive. This means that $l_1 T$ indeed exists. If $\omega_{\lambda}(l_1 T) \geq 0$, then, similarly as in (6.92) and (6.93), we obtain

$$y\,(l_1 T) \leq y_{\mathrm{D}} - \sum_{p=1}^{M} \sum_{j=l_1-n_p}^{l_1-1} \lambda_p\,(jT)\,u\,(jT). \tag{6.94}$$

The queue length at instant lT can be expressed relative to the queue length at instant $l_1 T$ in the following way:

$$y\,(lT) = y\,(l_1 T) + \sum_{p=1}^{M} \sum_{j=l_1-n_p}^{l-n_p-1} \lambda_p\,(jT)\,u\,(jT) - \sum_{j=l_1}^{l-1} h\,(jT). \tag{6.95}$$

Applying (6.94) and (6.95), we get

$$\begin{aligned} y\,(lT) &\leq y_{\mathrm{D}} - \sum_{p=1}^{M} \sum_{j=l_1-n_p}^{l_1-1} \lambda_p\,(jT)\,u\,(jT) + \sum_{p=1}^{M} \sum_{j=l_1-n_p}^{l-n_p-1} \lambda_p\,(jT)\,u\,(jT) - \sum_{j=l_1}^{l-1} h\,(jT) \\ &\leq y_{\mathrm{D}} + \sum_{p=1}^{M} \sum_{j=l_1}^{l-n_p-1} \lambda_p\,(jT)\,u\,(jT) - \sum_{j=l_1}^{l-1} h\,(jT). \end{aligned} \tag{6.96}$$

The controller established a nonzero transmission rate $u(\cdot)$ for the pool of active connections for the last time before lT at instant $l_1 T$, and this value, $u(l_1 T)$, could be as large as $u_{\max}$. Consequently, taking into account condition (6.85), we can estimate the weighted sum in (6.96) as

$$\sum_{p=1}^{M} \sum_{j=l_1}^{l-n_p-1} \lambda_p\,(jT)\,u\,(jT) = u\,(l_1 T) \sum_{p=1}^{M} \lambda_p\,(l_1 T) \leq u\,(l_1 T) \leq u_{\max}. \tag{6.97}$$

Since the utilized bandwidth $h(\cdot)$ is always nonnegative, then using (6.97), we obtain the following estimate of the queue length at instant lT

$$y\,(lT) \leq y_{\mathrm{D}} + u\,(l_1 T) - 0 \leq y_{\mathrm{D}} + u_{\max}. \tag{6.98}$$

This concludes the second part of the reasoning and completes the proof. □

Theorem 6.15 shows that if the flow of data in the pool of M connections is governed by the proposed controller, then the queue length never exceeds the level of $y_{\max}$. Thus, irrespective of the source activity profile in the pool and assumed rate allocation policy conforming to (6.85), the buffer capacity equal to $y_{\max}$ suffices to eliminate packet losses originating from congestion. In the next proposition, we indicate how the demand queue length should be selected in order to ensure $y(\cdot) > 0$, which implies full bandwidth utilization in the considered data transmission network.

Theorem 6.16. *If controller (6.89) with function $\omega_\lambda(\cdot)$ defined by (6.90) is applied to the system with variable resource allocation so that $\sum_{p=1}^{M} \lambda_p (kT) = 1$, and the demand queue length satisfies the following inequality:*

$$y_{\mathrm{D}} > u_{\max} (n_\lambda + 1/\gamma) + d_{\max}, \tag{6.99}$$

where $n_\lambda = \max\limits_k \sum_{p=1}^{M} \sum_{j=k-n_p}^{k-1} \lambda_p (jT)$, then for any $k \geq n_m + y_{max}/(u_{max} - d_{max})$, the queue length is strictly positive.

Proof. Let us consider some integer $l \geq n_m + y_{\max}/(u_{\max} - d_{\max})$ and the value of signal $\omega_\lambda(\cdot)$ at instant lT. We need to analyze two cases: the situation when $\omega_\lambda(lT) < u_{\max}$ and the circumstances when $\omega_\lambda(lT) \geq u_{\max}$.

Case 1. First, we consider the situation when $\omega_\lambda(lT) < u_{\max}$. Directly from the definition of function $\omega_\lambda(\cdot)$, we get

$$\omega_\lambda (lT) = \gamma \left[y_{\mathrm{D}} - y (lT) - \sum_{p=1}^{M} \sum_{j=l-n_p}^{l-1} \lambda_p (jT) u (jT) \right] < u_{\max}. \tag{6.100}$$

Rearranging the terms in (6.100), we obtain

$$y (lT) > y_{\mathrm{D}} - u_{\max} /\gamma - \sum_{p=1}^{M} \sum_{j=l-n_p}^{l-1} \lambda_p (jT) u (jT). \tag{6.101}$$

The overall transmission rate established according to (6.89) is upper-bounded by $u_{\max}$, which implies that

$$y (lT) > y_{\mathrm{D}} - u_{\max} /\gamma - u_{\max} \sum_{p=1}^{M} \sum_{j=l-n_p}^{l-1} \lambda_p (jT) \geq y_{\mathrm{D}} - u_{\max} (n_\lambda + 1/\gamma) . \tag{6.102}$$

Applying assumption (6.99), we get $y(lT) > d_{\max} > 0$, which concludes the first part of the proof.

Case 2. In the second part of the proof, we investigate the situation when $\omega_\lambda(lT) \geq u_{\max}$. First, we find the last moment l_1T before lT when signal $\omega_\lambda(\cdot)$ was smaller than $u_{\max}$. We know from Theorem 6.15 that the queue length never exceeds the value of $y_{\max}$. Moreover, the consumed bandwidth $h(\cdot)$ is limited by $d_{\max}$. Thus, the maximum interval during which the controller may continuously generate the maximum transmission rate is determined as $y_{\max}/(u_{\max} - d_{\max})$, and instant $l_1 T$ does exist. Furthermore, from the theorem assumptions, we get $l_1 \geq n_m$, and, for any instant kT, $\sum_{p=1}^{M} \lambda_p(kT) = 1$. This means that by the time $l_1 T$, the first packets from all the sources have already reached the node, no matter the value of delay and the actual allocation pattern.

The value of $\omega_\lambda(l_1 T) < u_{\max}$. Consequently, following a similar reasoning as presented in (6.100)–(6.102), we arrive at

$$y(l_1 T) > y_{\mathrm{D}} - u_{\max}/\gamma - \sum_{p=1}^{M} \sum_{j=l_1-n_p}^{l_1-1} \lambda_p(jT)\, u(jT) > 0. \tag{6.103}$$

If we apply (6.103) in (6.95), we may conclude that the queue length at instant lT satisfies the following inequality:

$$\begin{aligned} y(lT) > y_{\mathrm{D}} - \frac{u_{\max}}{\gamma} - \sum_{p=1}^{M} \sum_{j=l_1-n_p}^{l_1-1} \lambda_p(jT)\, u(jT) \\ + \sum_{p=1}^{M} \sum_{j=l_1-n_p}^{l-n_p-1} \lambda_p(jT)\, u(jT) - \sum_{j=l_1}^{l-1} h(jT). \end{aligned} \tag{6.104}$$

Working with the sums in (6.104) leads to

$$\begin{aligned} y(lT) > y_{\mathrm{D}} - \frac{u_{\max}}{\gamma} + \sum_{p=1}^{M} \sum_{j=l_1}^{l-1} \lambda_p(jT)\, u(jT) \\ - \sum_{p=1}^{M} \sum_{j=l-n_p}^{l-1} \lambda_p(jT)\, u(jT) - \sum_{j=l_1}^{l-1} h(jT). \end{aligned} \tag{6.105}$$

Moment $l_1 T$ was the last instant before lT when the controller calculated rate smaller than $u_{\max}$. This rate could be as low as zero. Afterwards, the controller generates the maximum rate value. Hence, since by the theorem assumptions for any kT we have $\sum_{p=1}^{M} \lambda_p(kT) = 1$, the first sum in (6.105) may be estimated in the following way:

$$\sum_{p=1}^{M} \sum_{j=l_1}^{l-1} \lambda_p(jT)\, u(jT) = \sum_{j=l_1}^{l-1} u(jT) = u(l_1 T) + \sum_{j=l_1+1}^{l-1} u(jT) \geq u_{\max}(l - 1 - l_1). \tag{6.106}$$

Since for any integer l, $u(lT) \leq u_{\max}$, the second sum in (6.105)

$$u_{\max} \sum_{p=1}^{M} \sum_{j=l-n_p}^{l-1} \lambda_p(jT) \leq u_{\max} n_\lambda. \tag{6.107}$$

Therefore, applying (6.106) and (6.107) to (6.105), we obtain

$$y(lT) > y_{\mathrm{D}} - \frac{u_{\max}}{\gamma} + u_{\max}(l-1-l_1) - u_{\max} n_\lambda - \sum_{j=l_1}^{l-1} h(jT). \tag{6.108}$$

Since $h(\cdot) \leq d_{\max}$, we get the following estimate of the queue length at instant lT:

$$y(lT) > y_{\mathrm{D}} - u_{\max}/\gamma + u_{\max}(l-1-l_1) - u_{\max} n_\lambda - d_{\max}(l-l_1). \tag{6.109}$$

Using the theorem assumption (6.99), we get

$$\begin{aligned} y(lT) > {} & u_{\max}(n_\lambda + 1/\gamma) + d_{\max} - u_{\max}/\gamma + u_{\max}(l-1-l_1) \\ & - u_{\max} n_\lambda - d_{\max}(l-l_1) = u_{\max}(l-1-l_1) - d_{\max}(l-1-l_1). \end{aligned} \tag{6.110}$$

Finally, since $l > l_1$ and $u_{\max} > d_{\max}$, we arrive at $y(lT) > (u_{\max} - d_{\max})(l-1-l_1) \geq 0$. This concludes the proof. □

Remark 6.3. Let us notice that neither the number of connections, nor their RTTs, can be determined *a priori*. Therefore, appropriate Call Admission Control (CAC) procedures deciding whether to accept or reject a new connection should ensure that condition (6.99) is satisfied. Also, the rate distribution function in the control algorithm may assist in fulfilling (6.99) by allocating the weights in such a way that for any k, $\sum_{p=1}^{M} \sum_{j=k-n_p}^{k-1} \lambda_p(jT) \leq n_\lambda$. If condition (6.99) is not satisfied, then it is no longer guaranteed that the queue length will be strictly positive, and part of the available bandwidth may be left unused. However, violating condition (6.99) does not lead to packet losses. If the buffer capacity is selected according to (6.91), then the entire packet queue can be always stored in the buffer, and congestion is avoided.

Performance of controller (6.89), regulating the flow of data in the considered multisource network with variable resource allocation and connection number, is illustrated in a simulation scenario described in the next section.

6.1.4.4 Simulation Results

We consider a data transmission network serving multiple flows with different RTTs. In the analyzed network, the feedback information is accessible for rate

Table 6.5 Rate allocation profile – numerical data

Flow	Allocation profile		
p	$k < 100$	$100 \leq k < 200$	$k \geq 200$
1	1/6	1/12	0
2	5/12	1/4	1/6
3	1/4	1/3	5/12
4	1/6	1/3	5/12

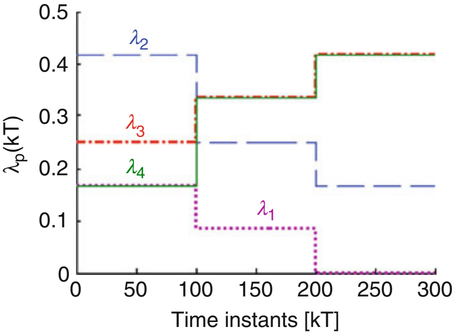


Fig. 6.24 Rate allocation profile – graphical representation

adaptation at discrete-time instants only with the discretization period set as $T = 10$ ms. We run two series of simulations for the available bandwidth illustrated in Fig. 6.2. The applied pattern shows a few abrupt bandwidth transitions occurring at instants $60T$, $120T$, $180T$, and $240T$. In the first simulation, we test performance of the proposed controller (6.89) in the situation of variable resource allocation, whereas in the second one, we focus on the issues related to time-varying number of active connections. In each test, we assume the maximum available bandwidth $d_{\max} = 100$ packets and the maximum transmission rate determined by the controller $u_{\max} = 110$ packets per discretization period. The controller gain is set as $\gamma = 0.618$.

Test 1. In the first scenario, we verify the controller performance when the distribution of the total rate among the flows changes during the transmission, for instance, according to the max-min or proportional fairness criteria. The pool of four flows ($M = 4$) passing through the controlling node is considered in the rate allocation. The flows are characterized by the following RTTs: $\text{RTT}_1 = 3T$, $\text{RTT}_2 = 7T$, $\text{RTT}_3 = 8T$, and $\text{RTT}_4 = 12T$. Depending on the individual limitations, for example, occurring due to the congestion experienced at some node in the network other than the one where the algorithm operates, the flows are assigned a part of the overall rate as listed in Table 6.5. Notice that for $k \geq 200$, no assignment is given to connection 1, which illustrates the situation of a flow being totally blocked at some node in the network (or a source which has already finished transferring its data). The allocation profiles are illustrated in Fig. 6.24.

We run two simulations. In the first one (curve a in the graphs), the demand queue length is set on the basis of the estimate of n_λ computed as $\sum_{p=1}^{M=4} n_p\,/4\ = 7.5$ (i.e., using the initial number of flows and equal weights $\lambda_p = 1/M = 1/4$). In the second

Table 6.6 Demand queue length and buffer capacity setting

Simulation	Demand queue length y_D [packets]	Buffer capacity y_{max} [packets]
1	1,105	1,215
2	1,325	1,435

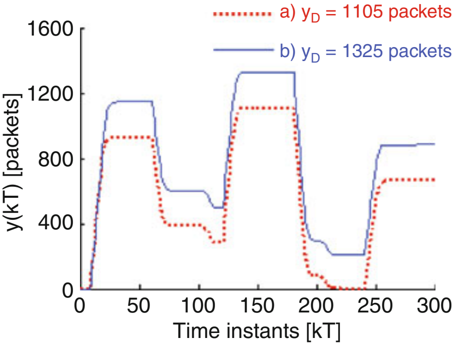


Fig. 6.25 Queue length: *a* $y_D = 1{,}105$ packets and *b* $y_D = 1{,}325$ packets

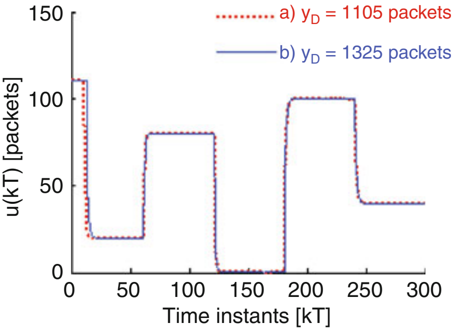


Fig. 6.26 Generated transmission rate: *a* $y_D = 1{,}105$ packets and *b* $y_D = 1{,}325$ packets

simulation (curve b in the graphs), in turn, it is assumed that the exact value of $n_\lambda = 9.5$ is known to the controller. Consequently, the demand queue length in the first simulation is set according to (6.99) as $y_D = 1{,}105 > 1{,}103$ packets and in the second one as $y_D = 1{,}325 > 1{,}323$ packets. The corresponding buffer sizes are set according to Theorem 6.15 as 1,215 and 1,435 packets, respectively. The controller parameter setting is summarized in Table 6.6.

The queue length evolution is shown in Fig. 6.25, the rate calculated by the algorithm in Fig. 6.26, and the rate assigned for each connection in Fig. 6.27. It is clear from the graphs in Fig. 6.25 that the buffer is not overflowed, which means that no packet needs to be dropped. When the controller knows the precise value of n_λ, the buffer is never entirely depleted (curve b) which implies that all of the available bandwidth is consumed for the transmission of data. When the lower value is applied (case a), then the queue length hits the zero level for a short period of time starting at $kT = 220T$, which brings the risk of reduced bandwidth utilization – 98% of the available bandwidth in the considered simulation interval is used efficiently for data

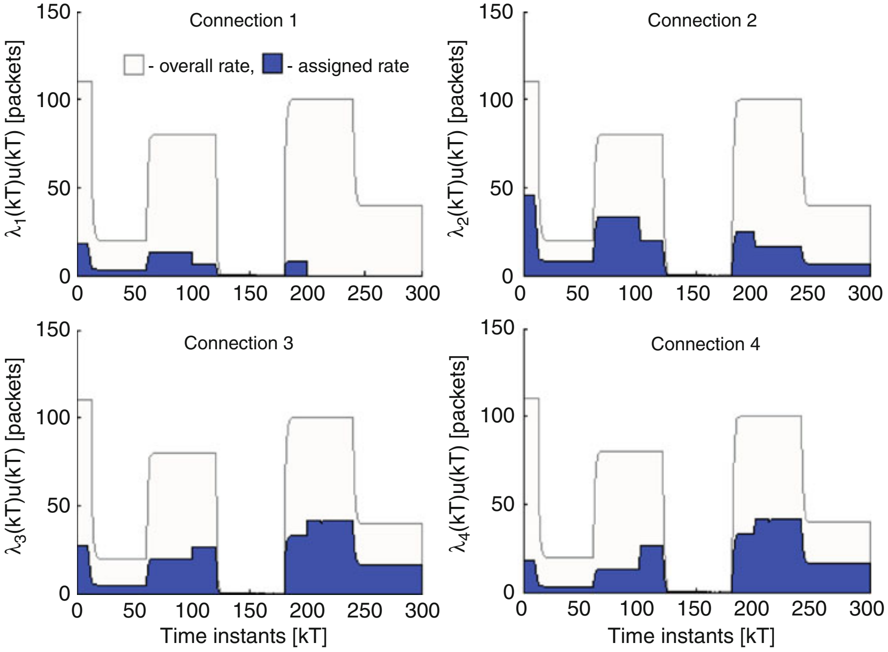


Fig. 6.27 Rate allocated for connections 1–4 ($y_D = 1{,}325$ packets)

transfer. The plots in Fig. 6.26 show that the transmission rate is nonnegative and bounded and quickly follows the changes in the bandwidth trend. In turn, the curves sketched in Fig. 6.27 demonstrate that the rates assigned for the sources follow the assumed allocation pattern (illustrated in Fig. 6.24). Since the curves of the total rate in cases a and b actually overlap except for the initial phase, we limit the display of the individual source rate assignments to the case of correct n_λ estimate b.

Test 2. In the second scenario, we test performance of controller (6.89) in the situation when the number of connections is not known *a priori* and changes with time. We increase the pool of flows to $M = 12$ and assign to each connection a different delay from the interval $[4T, 15T]$. The actual number of flows in the system $m(kT)$ evolves according to the plot depicted in Fig. 6.28. We can see from this plot that initially eight connections pass through the bottleneck node; at $k = 50$, four more connections join the system, whereas at $k = 130$, four (randomly chosen) sources finish the transmission. Again, at $k = 200$, four more connections terminate sending the data, and finally, at $k = 250$, four new flows appear in the network. The controller assigns the rates in the max-min fair way. The demand queue length is set as $y_D = 1{,}325$ packets, which corresponds to $n_\lambda = 9.5$. The actual n_λ did not exceed 9.5 during the whole simulation interval, and the initial estimate proved sufficient for ensuring full bandwidth utilization.

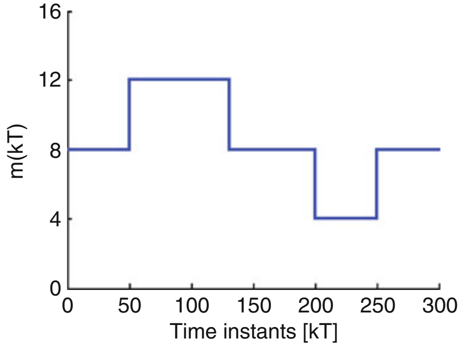


Fig. 6.28 Number of active connections $m(\cdot)$

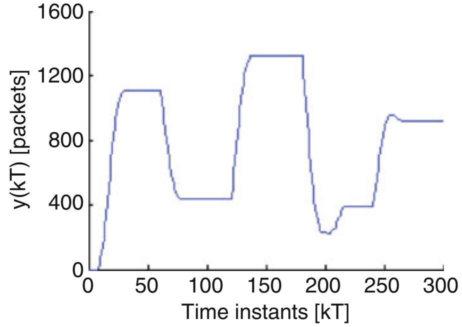


Fig. 6.29 Queue length when the number of active connections varies with time

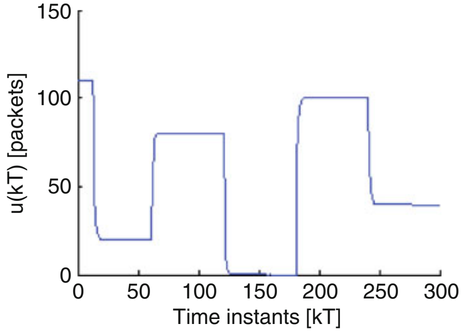


Fig. 6.30 Generated transmission rate when the number of active connections varies with time

The queue length is shown in Fig. 6.29 and the established transmission rate in Fig. 6.30. As we can see from the graph in Fig. 6.29, the queue length does not increase beyond the level of 1,435 packets, which is the buffer capacity assigned according to Theorem 6.15, and does not drop to zero. This implies that losses (occurring due to congestion) are completely eliminated and no bandwidth is wasted. The transmission rate and the queue length follow the bandwidth changes. We can also notice from Fig. 6.29 that the output variable experiences additional decrease for k around 200 even though the available bandwidth remains constant in this interval. This is attributed to the change of the number of active flows and the corresponding shift in the average delay. A similar observation applies to the additional $y(\cdot)$ increase for k around 250 which is a consequence of four new flows joining the system. However, the queue length quickly stabilizes at a steady-state level with no oscillations following the overshoots.

6.2 Flow Control in a Network with Variable Delay

The route the packets follow in the network may not always be assumed fixed during the transmission. The link and node failures, or the congestion occurring in various areas of the network, typically lead to route alteration and propagation latency changes. Also, the variable queuing delay in the buffers at the intermediate nodes related to different bandwidth patterns and local networking conditions influence the overall value of RTT of active connections. Consequently, in order to accurately represent the network dynamics and provide robustness enhancements, delay variations should typically be accounted for in the constructed network model and in the controller design procedure.

In this section, we release the assumption about the constant delay in the feedback loop and analyze the network in which both the propagation and queuing latency may change during the transmission. We consider a multisource traffic scenario with each flow subject to arbitrary and *a priori* unknown RTT fluctuations. We begin with extending the model presented in Sect. 6.1 to the case of variable delay in each of the data streams. Next, we propose two controllers providing the necessary robustness to changing propagation and queuing latency. The first controller combines the benefits of LQ dynamical optimization and saturation nonlinearity to provide a smooth and nonnegative rate signal. The second scheme employs extra delay variability compensator introduced in Sect. 5.2.3 to further decrease the influence of the unknown RTT fluctuations on the quality of the control process.

6.2.1 Network Model

We consider the multisource single-bottleneck communication scenario described in Sect. 6.1. Similarly as in that section, we emphasize here the discrete nature of crucial networking events that is related to the way the feedback information is distributed in the system. Consequently, we analyze the networks in which the feedback information used for the source rate adjustment is extracted from feedback carriers, being acknowledgments (as in TCP/IP-based networks), or control units emitted periodically by the sources (typical of certain connection-oriented solutions, e.g., ATM). Packets, as well as feedback carriers, experience delay as they pass through the nodes and travel on the internode links. Depending on the buffer

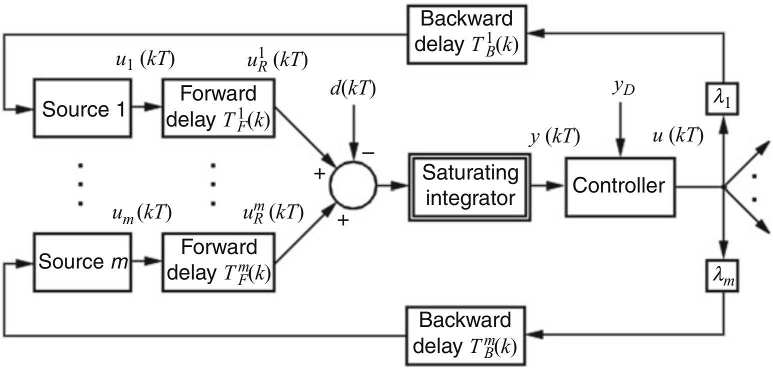


Fig. 6.31 Network model – multiple virtual circuits with time-varying delay

occupancy at the nodes, they are subject to time-varying queuing delay. Similarly, packets (and feedback carries) may be directed along different routes as they travel through the network, which implies variable propagation delay of the subsequent chunks of data streams. Therefore, in this section, we analyze the situation when the overall latency fluctuates according to the network state and the selected transmission path, potentially causing instability.

The model of the network serving multiple flows subject to time-varying delay is illustrated in Fig. 6.31. Similarly as in Sect. 6.1, we assume that the sources send packets at discrete-time instants kT, where T is the discretization period and $k = 0, 1, 2, \ldots,$ in the amounts $u_p(kT)$ determined by the controller placed at the controlling node on their path. After forward delay $T_{\mathrm{F}}^p(k)$, the packets from source p ($p = 1, 2, \ldots, m$) reach the node and are served according to the bandwidth availability at the node output link $d(kT)$. The remaining ones accumulate in the buffer. The packet queue length in the buffer, $y(kT)$, and its demand value y_{D}, are used to calculate the current amount of data $u(kT)$ to be sent by the sources. The total amount is distributed among the connections according to the assumed resource allocation policy. This means that at each time instant kT, a part λ_p of the total rate is assigned for source p. The rate allocation coefficients λ_p are real numbers from the interval [0, 1] satisfying $\sum_{p=1}^{m} \lambda_p = 1$. The information about the current rate is extracted at the sources from feedback carriers (acknowledgements or control units) with backward delay $T_{\mathrm{B}}^p(k)$ after leaving the bottleneck node. Note that here, in contrast to case addressed in Sect. 6.1, both the forward and backward delays are not constants but functions of time, which are typically unknown to the controller. Consequently, in this part of the work, we need to account for the circumstances when the actual amount of arriving packets $u_{\mathrm{R}}^p(kT)$ may differ from the value which was predicted by the controller at the instant of rate allocation. In order to emphasize the phenomena related to delay variations, we assume that the number of controlled flows and the allocation profile do not change during the transmission. However, that additional uncertainty can be treated in a similar way as in Sect. 6.1.4.

The time-varying RTT,

$$\mathrm{RTT}_p(k) = T_{\mathrm{F}}^p(k) + T_{\mathrm{B}}^p(k), \tag{6.111}$$

is assumed to be a multiple of the discretization period, i.e., $\mathrm{RTT}_p(k) = n_p(k)T$, where $n_p(k)$ and its nominal value $\bar{n}_p$ are positive integers satisfying the following inequalities:

$$\underset{p}{\forall}\ (1-\beta)\,\bar{n}_p \le n_p(k) \le (1+\beta)\,\bar{n}_p. \tag{6.112}$$

Parameter $\beta \in [0, 1)$ represents the assumed tolerance of delay variations. Since β is chosen as a relative measure of delay deviation from the nominal value, the delays in long-distance connections may experience larger absolute variations than local flows, as is typically the case in real networks. Without loss of generality, we may order the flows according to the nominal value of their round-trip time $\overline{\mathrm{RTT}}_p = \bar{n}_p T$ in the following way:

$$\overline{\mathrm{RTT}}_1 \le \overline{\mathrm{RTT}}_2 \le \cdots \le \overline{\mathrm{RTT}}_{m-1} \le \overline{\mathrm{RTT}}_m. \tag{6.113}$$

As a result of delay variations, certain feedback information carriers reaching the data sources may appear out of order and concurrently with carriers from other periods. Similarly, the packets directed along different paths may arrive at the node in the order different from the sending one (possibly together with the packets sent according to other assignments). In such situation, we assume that all the rate assignments retrieved from feedback carriers arriving at the sources (and packets reaching the node) in the same period are added to each other so that the congestion problem cannot be solved by some accidental yet advantageous control or input signal distortion.

The available bandwidth $d(kT)$ (the number of packets that may leave the bottleneck node at instant kT) and the utilized bandwidth $h(kT)$ (the number of packets actually leaving the node at instant kT) are modeled as *a priori* unknown, bounded functions of time, as before (relation (5.2)). The rate of change of the queue length depends on the amount of arriving data and on the utilized bandwidth $h(\cdot)$, i.e., we can write

$$y\,[(k+1)\,T] = y\,(kT) + \sum_{p=1}^{m} u_{\mathrm{R}}^p\,(kT) - h\,(kT)\,. \tag{6.114}$$

Since the rate allocation assignment is realized with delay, $u_{\mathrm{R}}^p(kT) = \lambda_p u[kT - \mathrm{RTT}_p(k)]$. Thus, we can represent (6.114) as

$$y\,[(k+1)\,T] = y\,(kT) + \sum_{p=1}^{m} \lambda_p u\left[kT - \mathrm{RTT}_p(k)\right] - h\,(kT)\,. \tag{6.115}$$

Assuming that the buffer is initially empty, i.e., $y(0) = 0$, and the first rates are assigned at $kT = 0$, i.e., $u(kT) = 0$ for $k < 0$, the packet queue length at the node for any $k \geq 0$ may be calculated from the following equation:

$$y\left(kT\right) = \sum_{p=1}^{m}\sum_{j=0}^{k-1}\lambda_p u\left[jT - \mathrm{RTT}_p(j)\right] - \sum_{j=0}^{k-1}h\left(jT\right). \tag{6.116}$$

Let us introduce a function:

$$\xi\left(kT\right) = \xi_+\left(kT\right) - \xi_-\left(kT\right), \tag{6.117}$$

where

$$\xi_+\left(kT\right) = \sum_{p=1}^{m}\sum_{j\in(0,\beta\bar{n}_p]:n_p(k+j)\leq\bar{n}_p-j}\lambda_p u\left[\left(k-\bar{n}_p+j\right)T\right] \tag{6.118}$$

represents the sum of these surplus packets which arrive at the node by the time kT and earlier than expected since their true latency experienced in the neighborhood of kT is smaller than the nominal one. Therefore, function $\xi_+(kT)$ accounts for all the packets from all the sources which in the system with constant delay would reach the node after kT, but in the considered network with variable delay they contribute to the queue buildup by kT since their delay is smaller than the nominal one. Function $\xi_-(kT)$, in turn,

$$\xi_-\left(kT\right) = \sum_{p=1}^{m}\sum_{j\in[0,\beta\bar{n}_p]:n_p(k-j)>\bar{n}_p+j}\lambda_p u\left[\left(k-\bar{n}_p-j\right)T\right], \tag{6.119}$$

represents the sum of these packets which should have arrived at the node by the time kT but which cannot reach it due to the (instantaneous) delay bigger than the nominal one. Thus, $\xi_-(kT)$ accounts for all the packets from all the sources which in the system with constant delay would appear at the node by kT but in the network with time-varying latency are excessively delayed and contribute to the queue buildup at some time instant (or instants) afterwards. The components of function $\xi(\cdot)$ for the case of one flow are illustrated in Fig. 5.45.

Assuming that the rate is bounded by some positive constant $u_{\max}$ (which is the case in any real network), and using (6.112), the following constraint can be formulated:

$$\underset{k\geq 0}{\forall}\ \left|\xi\left(kT\right)\right| \leq \xi_{\max} = u_{\max}\beta\sum_{p=1}^{m}\lambda_p\bar{n}_p. \tag{6.120}$$

With this notation, we can present the formula for the queue length as

$$y\left(kT\right)=\sum_{p=1}^{m}\sum_{j=0}^{k-1}\lambda_{p}u\left(jT-\bar{n}_{p}T\right)+\xi\left(kT\right)-\sum_{j=0}^{k-1}h\left(jT\right),\tag{6.121}$$

which reflects the nominal system operation (packets arriving due to the nominal delay) affected by perturbation $\xi(\cdot)$. Taking into account the initial conditions, relation for $y(kT)$ can be further rewritten as

$$\begin{aligned}y\left(kT\right)&=\sum_{p=1}^{m}\sum_{j=-\bar{n}_{p}}^{k-\bar{n}_{p}-1}\lambda_{p}u\left(jT\right)+\xi\left(kT\right)-\sum_{j=0}^{k-1}h\left(jT\right)\\&=\sum_{p=1}^{m}\sum_{j=0}^{k-\bar{n}_{p}-1}\lambda_{p}u\left(jT\right)+\xi\left(kT\right)-\sum_{j=0}^{k-1}h\left(jT\right).\end{aligned}\tag{6.122}$$

6.2.1.1 State-Space Representation

Denoting the share of connections with the nominal delay $jT(j=1,2,\ldots,\bar{n}_{m})$ in the total rate by $a_{j}=\sum_{p:\bar{n}_{p}T=jT}\lambda_{p}$, the network model with time-varying delay can be presented in the state space similarly as in (5.159), i.e.,

$$\begin{aligned}\mathbf{x}\left[\left(k+1\right)T\right]&=\mathbf{A}\mathbf{x}\left(kT\right)+\mathbf{b}u\left(kT\right)+\mathbf{v}_{1}h\left(kT\right)+\mathbf{v}_{2}\xi\left(kT\right),\\y\left(kT\right)&=\mathbf{q}^{T}\mathbf{x}\left(kT\right),\end{aligned}\tag{6.123}$$

where $\mathbf{x}(\cdot)$ is the $n\times 1$ state vector and $u(\cdot)$ is a scalar. The state matrix $\mathbf{A}_{n\times n}$, input $\mathbf{b}_{n\times 1}$, output $\mathbf{q}_{n\times 1}$, and $n\times 1$ disturbance vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are defined as

$$\mathbf{A}=\begin{bmatrix}1&a_{n-1}&a_{n-2}&\ldots&a_{1}\\0&0&1&\ldots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\ldots&1\\0&0&0&\ldots&0\end{bmatrix},\mathbf{b}=\begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix},\mathbf{q}=\begin{bmatrix}1\\0\\\vdots\\0\\0\end{bmatrix},\mathbf{v}_{1}=\begin{bmatrix}-1\\0\\\vdots\\0\\0\end{bmatrix},\text{ and }\mathbf{v}_{2}=\begin{bmatrix}1\\0\\\vdots\\0\\0\end{bmatrix}.\tag{6.124}$$

The system order $n=\bar{n}_{m}+1$ and the demand state vector $\mathbf{x_d}=[y_{\mathrm{D}}\;0\;0\ldots\;0]^{\mathrm{T}}$.

Further in Sect. 6.2, we present two control algorithms for the multisource network with variable delay (6.123) and (6.124). We formulate a number of properties of the proposed schemes, prove them analytically, and afterwards illustrate in simulations.

Table 6.7 Connection delay range

	RTT_p		
Connection	min	nom	max
1	$3T$	$4T$	$5T$
2	$3T$	$5T$	$7T$
3	$4T$	$6T$	$8T$

6.2.2 SM Controller with Saturation

Before formulating the control law, we first discuss the detrimental effect the delay variation may inflict on the performance of a data transmission network and communication system stability. These issues are addressed in Example 6.2.

Example 6.2. Let us analyze the events taking place in the process of controlling the flow of data in three connections ($m = 3$). We assume that the data streams in the connections originate at different sources and are subject to different latencies. The considered flows are characterized by the following nominal RTTs: $\overline{\mathrm{RTT}}_1 = \bar{n}_1 T = 4T$, $\overline{\mathrm{RTT}}_2 = \bar{n}_2 T = 5T$, and $\overline{\mathrm{RTT}}_3 = \bar{n}_3 T = 6T$. Assuming the delay tolerance $\beta = 0.4$ (40% of a possible deviation from the nominal value), we get the range of latencies listed in Table 6.7.

Let us assume that the source transmission rate is regulated according to control law (6.27) adjusted for a dead-beat scheme, i.e., with the gain $\gamma = 1$. The demand queue length y_{D} is set equal to 30 packets, and the allocation weights $\lambda_1 = \lambda_2 = \lambda_3 = 1/3$. Consequently, the rate calculation is performed according to the following equation:

$$u(kT) = y_{\mathrm{D}} - y(kT) - \sum_{p=1}^{3} \sum_{j=k-\bar{n}_p}^{k-1} \frac{1}{3} u(jT), \tag{6.125}$$

where $u(kT)$ represents the overall number of packets to be sent by the sources determined at instant kT. To simplify the analysis, we assume that the available bandwidth at the controlling node output interface $d(\cdot) \equiv 0$ during the transmission. Hence, we will study the rate allocation and packet arrivals in the multisource network subject to perturbation caused by delay variations only.

Scenario 1. First, we study the rate allocation in the circumstances of constant delay perfectly matching the values used by the controller. Thus, from (6.125), we get

$$u(0) = y_{\mathrm{D}} = 30 \text{ packets},$$
$$u(1 < k \leq 4) = 30 - 0 - 30 = 0.$$

The packets sent by the sources according to the initial assignment arrive at $kT = 4\,T$ (10 packets from source 1), $kT = 5\,T$ (10 packets from source 2), and $kT = 6\,T$ (10 packets from source 3). The packets are stored in the buffer indefinitely since by assumption there is no bandwidth available at the outgoing link. This means that the queue length evolves as follows:

$$\begin{aligned} y\,(k < 5) &= 0, \\ y\,(5T) &= 10 \text{ packets}, \\ y\,(6T) &= 20 \text{ packets}, \\ y\,(7T) &= 30 \text{ packets}, \\ y\,(k > 7) &= 30 \text{ packets}. \end{aligned}$$

Note that the rate $u(kT) = 0$ for $k > 0$, since any decrease in the sum in (6.125) is immediately compensated by an appropriate increase in the queue length $y(kT)$.

Scenario 2. Let us investigate the situation when the delay of connections 1 and 3 is fixed and estimated correctly by the controller. However, the packets from source 2 arrive with delay different from the nominal one. We assume that RTT of the initial assignment in flow 2 equals $4T$. Hence, instead of arriving at $kT = 5T$, as expected by the controller, the first packets sent by source 2 appear at the node one period earlier. Thus, from (6.125), we get

$$\begin{aligned} u(0) &= y_{\mathrm{D}} = 30 \text{ packets}, \\ u\,(1 < k \le 4) &= 30 - 0 - 30 = 0. \end{aligned}$$

However, the queue length

$$y\,(5T) = y\,(4T) + u_{\mathrm{R}}^{1}\,(4T) + u_{\mathrm{R}}^{2}\,(4T) = 0 + 10 + 10 = 20 \text{ packets},$$

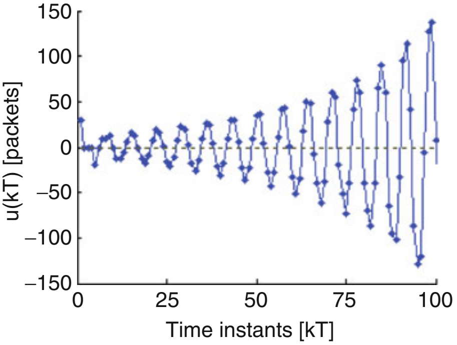
which implies

$$\begin{aligned} u\,(5T) &= y_{\mathrm{D}} - y\,(5T) - \left[\sum_{j=5-4}^{5-1} \frac{1}{3} u\,(jT) + \sum_{j=5-5}^{5-1} \frac{1}{3} u\,(jT) + \sum_{j=5-6}^{5-1} \frac{1}{3} u\,(jT) \right] \\ &= y_{\mathrm{D}} - y\,(5T) - \left[0 + \frac{1}{3} u(0) + \frac{1}{3} u(0) \right] \\ &= 30 - 20 - (10 + 10) = -10 \text{ packets}. \end{aligned}$$

Thus, we obtain an infeasible (negative) rate value at the instant $kT = 5\,T$.

Similarly, if the packets from source 2 arrive at $kT = 6T$, i.e., one period after they are expected by the controller, we have the queue length evolution

Fig. 6.32 Transmission rate



$$
\begin{aligned}
y\,(k < 5) &= 0,\\
y\,(5T) &= 10 \text{ packets},\\
y\,(6T) &= 10 \text{ packets},\\
y\,(7T) &= 30 \text{ packets},
\end{aligned}
$$

and the transmission rate

$$
\begin{aligned}
u(0) &= y_D = 30 \text{ packets},\\
u\,(1 < k \leq 4) &= 30 - 0 - 30 = 0,\\
u\,(5T) &= 30 - 10 - 20 = 0,\\
u\,(6T) &= 30 - 10 - 10 = 10 \text{ packets},\\
u\,(7T) &= 30 - 30 - 10 = -10 \text{ packets}.
\end{aligned}
$$

Thus, an infeasible assignment occurs at instant $kT = 7T$.

It follows from the presented analysis that an improper delay estimate in just a single connection in the entire pool of flows may lead to infeasible rate assignments. It may also cause stability problems as shown below in Scenario 3.

Scenario 3. The discussion in this example concludes with a simulation run for the system with the delay of each flow set at the left margin of the range specified in Table 6.7. In the test, we assume that the traffic intensity in the network is regulated by controller (6.125), which uses the nominal RTT values.

The rate assignments made by the controller (with negative signals permitted for the purpose of analysis) are depicted in Fig. 6.32 and the buffer occupancy in Fig. 6.33. We can see from the plots that the investigated controller, which exhibits excellent dynamical properties in the network with fixed delay, is no longer appropriate for the system with latency mismatch. In the considered scenario, the system becomes unstable.

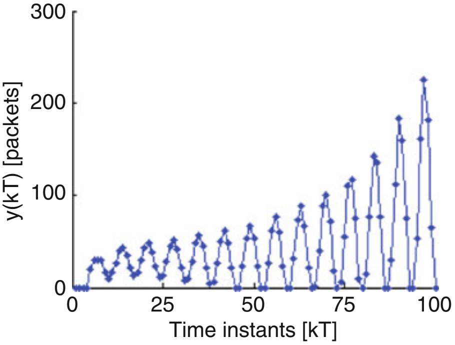


Fig. 6.33 Packet queue length

Example 6.2 clearly shows that even a minor delay mismatch may lead to infeasible rate allocation and stability problems. In the next section, we present a nonlinear control strategy which provides the desired robustness to delay uncertainty.

6.2.2.1 Proposed Control Strategy

So far in this chapter, we described several flow control algorithms which provide prompt reaction to the changes of networking conditions and ensure closed-loop stability for arbitrary known, constant delays. We demonstrated in Example 6.2 that when the delay in the feedback loop is estimated by the controller incorrectly, or it changes in a way unknown to the controller, the control scheme no longer operates as desired. Even if the delay mismatch concerns a single flow, the controller may generate infeasible rate assignments and in certain circumstances make the considered communication system unstable. However, it is still possible to retain the favorable properties of the controllers developed so far in this chapter, if appropriate modifications are introduced to improve robustness. In what follows, the benefits of smooth reaction to bandwidth fluctuations of the LQ optimal SM controller are combined with a saturation nonlinearity to ensure feasible rate allocations in the network with variable delay.

The following flow control algorithm is proposed for the flows exhibiting uncertain fluctuations of delay. The overall transmission rate for the connections passing through the controlling node is calculated as

$$u(kT) = \begin{cases} 0, & \text{if} \quad \omega_\gamma(kT) < 0, \\ \omega_\gamma(kT), & \text{if} \quad 0 \le \omega_\gamma(kT) \le u_{\max}, \\ u_{\max}, & \text{if} \quad \omega_\gamma(kT) > u_{\max}, \end{cases} \tag{6.126}$$

where $u_{\max} > d_{\max}$, and function $\omega_\gamma(\cdot)$ is defined in the following way:

$$\omega_\gamma(kT) = \gamma \left[y_{\mathrm{D}} - y(kT) - \sum_{p=1}^{m} \lambda_p \sum_{j=k-\bar{n}_p}^{k-1} u(jT) \right], \tag{6.127}$$

where the gain $\gamma \in (0, 1]$. As a result of introducing the transmission rate constraint (6.126), the rate established by the proposed robust control law is confined to the interval $[0, u_{\max}]$. In this fashion, a feasible transmission rate allocation is ensured irrespective of the delay and bandwidth oscillations. It follows from the derivations given in Sect. 6.1.4 that this type of controller also provides robustness enhancements in the case of variable flow number and time-varying allocation profile.

In a latter part of this section, it will be shown that the proposed control strategy (6.126) guarantees that packet losses are eliminated and the maximum throughput in the multisource network is achieved even though the delay experienced by packets and feedback carriers varies with time.

6.2.2.2 Properties of the Proposed Strategy

The properties of controller (6.126) will be formulated as two theorems and strictly proved. First, it will be demonstrated that the queue length is limited by a finite value. Therefore, if the buffer capacity is selected to accommodate this maximum number of packets, then the risk of losses related to congestion will be eliminated. Secondly, it will be shown that with the appropriately chosen demand queue length, there are always packets in the buffer to be relayed at the bottleneck link at maximum speed determined by the available bandwidth. Thus, with the indicated y_{D} setting, full bandwidth utilization will be obtained.

Theorem 6.17. *If controller (6.126) with function $\omega_\gamma(\cdot)$ defined by (6.127) is applied to system (6.123) and (6.124), then for any $k \geq 0$, the queue length at the bottleneck node does not exceed $y_{max} = y_D + u_{max} + \xi_{max}$.*

Proof. Taking into account the initial conditions and possible delay variation, the first packets arrive at the node no sooner than at $kT = (1-\beta)\bar{n}_1 T$ and $y(kT) = 0$ for all $k \leq (1-\beta)\bar{n}_1 T$. Consequently, it is sufficient to show that the proposition holds for all $kT > (1-\beta)\bar{n}_1 T$. Let us consider some integer $l > (1-\beta)\bar{n}_1$ and the value of function $\omega_\gamma(\cdot)$ at time instant lT. Two cases ought to be analyzed: the situation when $\omega_\gamma(lT) \geq 0$ and the circumstances when $\omega_\gamma(lT) < 0$.

Case 1. We investigate the situation when $\omega_\gamma(lT) \geq 0$. From the definition of function $\omega_\gamma(\cdot)$, (6.127), we get

$$\omega_\gamma(lT) = \gamma \left[y_{\mathrm{D}} - y(lT) - \sum_{p=1}^{m} \lambda_p \sum_{j=l-\bar{n}_p}^{l-1} u(jT) \right] \geq 0, \tag{6.128}$$

which leads to

$$y(lT) \leq y_{\mathrm{D}} - \sum_{p=1}^{m} \lambda_p \sum_{j=l-\bar{n}_p}^{l-1} u(jT). \tag{6.129}$$

From the algorithm definition, in turn, it follows that $u(\cdot)$ is always nonnegative, which implies $y(lT) \leq y_{\mathrm{D}}$. This completes the first part of the proof.

Case 2. In the second part of the proof, we analyze the situation when $\omega_\gamma(lT) < 0$. First, we find the most recent instant $l_1 T < lT$ when $\omega_\gamma(\cdot)$ was nonnegative. According to (6.127), $\omega_\gamma(0) = \gamma y_{\mathrm{D}} > 0$ since both γ and y_{D} are positive, which means that moment $l_1 T$ does exist. If $\omega_\gamma(l_1 T) \geq 0$, then, with analogy to (6.128) and (6.129), we obtain

$$y\left(l_1 T\right) \leq y_{\mathrm{D}} - \sum_{p=1}^{m} \lambda_p \sum_{j=l_1-\bar{n}_p}^{l_1-1} u\left(jT\right). \tag{6.130}$$

In the analyzed system, the queue length at instant lT can be expressed relative to $y(l_1 T)$ in the following way:

$$y\left(lT\right) = y\left(l_1 T\right) + \sum_{p=1}^{m} \lambda_p \sum_{j=l_1}^{l-1} u\left[jT - n_p(j)T\right] - \sum_{j=l_1}^{l-1} h\left(jT\right). \tag{6.131}$$

Using the concept of disturbance $\xi(\cdot)$, we may rewrite (6.131) as

$$y\left(lT\right) = y\left(l_1 T\right) + \sum_{p=1}^{m} \lambda_p \sum_{j=l_1-\bar{n}_p}^{l-\bar{n}_p-1} u\left(jT\right) + \xi\left(lT\right) - \sum_{j=l_1}^{l-1} h\left(jT\right). \tag{6.132}$$

Consequently, the queue length at instant lT equals:

- the queue length at instant $l_1 T$, $y(l_1 T)$,
- augmented by the sum of packets which arrive at the node in the interval $(l_1 T, lT]$ with the nominal delay, $\sum_{p=1}^{m} \lambda_p \sum_{j=l_1-\bar{n}_p}^{l-\bar{n}_p-1} u\left(jT\right)$, corrected by the effects related to delay variations represented by function $\xi(lT)$,
- decreased by the amount of served packets $\sum_{j=l_1}^{l-1} h\left(jT\right)$.

Applying inequality (6.130) to formula (6.132), we get

$$\begin{aligned} y\left(lT\right) &\leq y_{\mathrm{D}} - \sum_{p=1}^{m} \lambda_p \sum_{j=l_1-\bar{n}_p}^{l_1-1} u\left(jT\right) + \sum_{p=1}^{m} \lambda_p \sum_{j=l_1-\bar{n}_p}^{l-\bar{n}_p-1} u\left(jT\right) + \xi\left(lT\right) - \sum_{j=l_1}^{l-1} h\left(jT\right) \\ &\leq y_{\mathrm{D}} + \sum_{p=1}^{m} \lambda_p \sum_{j=l_1}^{l-\bar{n}_p-1} u\left(jT\right) + \xi\left(lT\right) - \sum_{j=l_1}^{l-1} h\left(jT\right). \end{aligned} \tag{6.133}$$

The controller allocated a nonzero rate for the active connections for the last time before lT at instant l_1T, and this value could be as large as $u_{\max}$. Consequently, since $\sum_{p=1}^{m} \lambda_p = 1$, the sum

$$\sum_{p=1}^{m} \lambda_p \sum_{j=l_1}^{l-\bar{n}_p-1} u\left(jT\right) = \sum_{p=1}^{m} \lambda_p u\left(l_1 T\right) = u\left(l_1 T\right) \leq u_{\max}. \tag{6.134}$$

Since the utilized bandwidth $h(\cdot)$ is always nonnegative, then using the constraint $\xi(lT) \leq \xi_{\max}$, from (6.133) and (6.134), we obtain the following estimate of the queue length at instant lT:

$$y\left(lT\right) \leq y_{\mathrm{D}} + u\left(l_1 T\right) + \xi\left(lT\right) - 0 \leq y_{\mathrm{D}} + u_{\max} + \xi_{\max}. \tag{6.135}$$

This concludes the second part of the reasoning and completes the proof. □

It follows from Theorem 6.17 that if the buffer of capacity $y_{\max} = y_{\mathrm{D}} + u_{\max} + \xi_{\max}$ is reserved at the controlling node, then no packet will need to be dropped in the considered control scheme, and potential losses due to congestion are eliminated. This is achieved even though neither the pattern nor statistics of the actual bandwidth and delay variations are known. Only the estimate of the upper bound of these quantities is required. In the second theorem, presented below, we show how the demand queue length should be selected in order to ensure full bandwidth utilization in the considered multisource network with time-varying delay.

Theorem 6.18. *If controller (6.126) with function $\omega_\gamma(\cdot)$ defined by (6.127) is applied to system (6.123) and (6.124), and the demand queue length satisfies the following inequality:*

$$y_{\mathrm{D}} > u_{\max} \left(\sum_{p=1}^{m} \lambda_p \bar{n}_p + 1/\gamma \right) + d_{\max} + \xi_{\max}, \tag{6.136}$$

then for any $k \geq (1+\beta)\bar{n}_m + n_{\max}$, where $n_{\max} = y_{\max}/(u_{\max} - d_{\max})$, the queue length is strictly positive.

Proof. The theorem assumption implies that we deal with time instants $kT \geq (1+\beta)\bar{n}_m T + n_{\max} T$. Considering some integer $l \geq (1+\beta)\bar{n}_m + n_{\max}$ and the value of signal $\omega_\gamma(\cdot)$ at instant lT, we may discern two complementary cases: the situation when $\omega_\gamma(lT) < u_{\max}$ and the circumstances when $\omega_\gamma(lT) \geq u_{\max}$.

Case 1. First, we consider the situation when $\omega_\gamma(lT) < u_{\max}$. From the definition of function $\omega_\gamma(\cdot)$, we obtain

$$\omega_\gamma(lT) = \gamma\left[y_\mathrm{D} - y(lT) - \sum_{p=1}^{m}\lambda_p \sum_{j=l-\bar{n}_p}^{l-1} u(jT)\right] < u_{\max}, \tag{6.137}$$

which after the term rearrangement gives

$$y(lT) > y_\mathrm{D} - \frac{u_{\max}}{\gamma} - \sum_{p=1}^{m}\lambda_p \sum_{j=l-\bar{n}_p}^{l-1} u(jT). \tag{6.138}$$

The transmission rate generated according to (6.126) is always bounded by $u_{\max}$, which implies that

$$y(lT) > y_\mathrm{D} - \frac{u_{\max}}{\gamma} - \sum_{p=1}^{m}\lambda_p \bar{n}_p u_{\max}. \tag{6.139}$$

Therefore, using assumption (6.136), we get $y(lT) > 0$. This concludes the first part of the proof.

Case 2. In the second part of the proof, we investigate the situation when $\omega_\gamma(lT) \geq u_{\max}$. First, we find the most recent moment $l_1T < lT$ when signal $\omega_\gamma(\cdot)$ was smaller than $u_{\max}$. It follows from Theorem 6.17 that the queue length never exceeds the value of $y_{\max}$. Furthermore, we know that the consumed bandwidth is limited by $d_{\max}$. Thus, the maximum interval $n_{\max}T$ during which the controller may continuously generate the maximum transmission rate is determined as $n_{\max}T = Ty_{\max}/(u_{\max} - d_{\max})$, and instant l_1T does exist. Moreover, from the theorem assumptions, we get $l_1T \geq (1+\beta)\bar{n}_mT$, which means that by the time l_1T, the first packets from all the sources have already reached the node, no matter the delay variation.

The value of $\omega_\gamma(l_1T) < u_{\max}$. Consequently, following a reasoning similar to the one presented in (6.137)–(6.139), we arrive at

$$y(l_1T) > y_\mathrm{D} - \frac{u_{\max}}{\gamma} - \sum_{p=1}^{m}\lambda_p \sum_{j=l_1-\bar{n}_p}^{l_1-1} u(jT) > 0. \tag{6.140}$$

Thus, by referring to (6.132), the queue length at instant lT satisfies the inequality

$$\begin{aligned} y(lT) > y_\mathrm{D} - \frac{u_{\max}}{\gamma} - \sum_{p=1}^{m}\lambda_p \sum_{j=l_1-\bar{n}_p}^{l_1-1} u(jT) \\ + \sum_{p=1}^{m}\lambda_p \sum_{j=l_1-\bar{n}_p}^{l-\bar{n}_p-1} u(jT) + \xi(lT) - \sum_{j=l_1}^{l-1} h(jT). \end{aligned} \tag{6.141}$$

Working with the sums in (6.141) yields

$$y\left(lT\right) > y_{\mathrm{D}} - \frac{u_{\max}}{\gamma} + \sum_{p=1}^{m} \lambda_p \sum_{j=l_1}^{l-1} u\left(jT\right) \\ - \sum_{p=1}^{m} \lambda_p \sum_{j=l-\bar{n}_p}^{l-1} u\left(jT\right) + \xi\left(lT\right) - \sum_{j=l_1}^{l-1} h\left(jT\right). \tag{6.142}$$

Recall that l_1T was the last instant before lT when the controller calculated rate smaller than $u_{\max}$. This rate could be as low as zero. Afterwards, the algorithm generates the maximum transmission rate, and since $\sum_{p=1}^{m} \lambda_p = 1$, the first sum in (6.142),

$$\sum_{p=1}^{m} \lambda_p \sum_{j=l_1}^{l-1} u\left(jT\right) = \sum_{j=l_1}^{l-1} u\left(jT\right) = u\left(l_1 T\right) + \sum_{j=l_1+1}^{l-1} u\left(jT\right) \geq u_{\max}\left(l - 1 - l_1\right). \tag{6.143}$$

Since for any k, $u(kT) \leq u_{\max}$, the second sum is upper-bounded by $u_{\max} \sum_{p=1}^{m} \lambda_p \bar{n}_p$. Therefore, we obtain

$$y\left(lT\right) > y_{\mathrm{D}} - \frac{u_{\max}}{\gamma} + u_{\max}\left(l - 1 - l_1\right) - u_{\max} \sum_{p=1}^{m} \lambda_p \bar{n}_p + \xi\left(lT\right) - \sum_{j=l_1}^{l-1} h\left(jT\right). \tag{6.144}$$

Using the fact that $\xi(\cdot) \geq -\xi_{\max}$ and $h(\cdot) \leq d_{\max}$, we get the following estimate of the queue length at instant lT:

$$y\left(lT\right) > y_{\mathrm{D}} - u_{\max}/\gamma + u_{\max}\left(l - 1 - l_1\right) - u_{\max} \sum_{p=1}^{m} \lambda_p \bar{n}_p - \xi_{\max} - d_{\max}\left(l - l_1\right). \tag{6.145}$$

Applying the theorem assumption (6.136), we obtain

$$y\left(lT\right) > u_{\max} \sum_{p=1}^{m} \lambda_p \bar{n}_p + u_{\max}/\gamma + d_{\max} + \xi_{\max} - u_{\max}/\gamma + u_{\max}\left(l - 1 - l_1\right) \\ - u_{\max} \sum_{p=1}^{m} \lambda_p \bar{n}_p - \xi_{\max} - d_{\max}\left(l - l_1\right) = u_{\max}\left(l - 1 - l_1\right) \\ - d_{\max}\left(l - 1 - l_1\right). \tag{6.146}$$

Finally, since $l > l_1$ and $u_{\max} > d_{\max}$, we arrive at $y(lT) > 0$. This conclusion ends the proof. □

Remark 6.4. Note that function $\xi(\cdot)$ represents the cumulative surplus (or deficiency) of packets arriving at the node. Consequently, the properties stated in Theorems 6.17 and 6.18 will still be ensured if certain flows violate delay constraint (6.112), provided that condition (6.120) is satisfied. The values indicated in the theorems may also serve as guidelines for assessing the level of robustness if the estimate of the maximum delay deviation is not accessible. In other words, a particular value of y_{D} and $y_{\max}$ gives indication how much the system can be perturbed while still operating at the maximum throughput.

Remark 6.5. Even though the disturbance rejection cannot be obtained in the considered network because the invariance property [4] does not hold for systems with mismatched disturbances and systems with discrete-time control [11], a high level of robustness is achieved. Indeed, as stated in Theorems 6.17 and 6.18, the proposed strategy allows us to eliminate data losses and obtain full bandwidth usage in the network even though we do not know the pattern of delay variations, neither incorporate explicit disturbance measurement into the control law. The system stability and reaching conditions are satisfied because the perturbations are accounted for indirectly in the controller operation through the input history $\sum_{p=1}^{m} \lambda_p \sum_{j=k-\bar{n}_p}^{k-1} u(jT)$ and measurement of the output variable $y(kT)$.

In the remainder of this section, we will present simulation results demonstrating the properties of robust control law (6.126) in the considered network with time-varying delay.

6.2.2.3 Simulation Results

We analyze the networking phenomena related to transferring data in the network with variable delay modeled according to the description given in Sect. 6.2.1. We assume that four flows ($m = 4$) participate in the control process and compete for the bandwidth at the outgoing link of the controlling node. The flows are characterized by the following nominal RTTs (assumed known to the controller): $\overline{\mathrm{RTT}}_1 = 3T$, $\overline{\mathrm{RTT}}_2 = 7T$, $\overline{\mathrm{RTT}}_3 = 8T$, and $\overline{\mathrm{RTT}}_4 = 12T$, where the discretization period $T = 10$ ms. The true delay varies in a way unknown to the control algorithm in the range specified by $\beta = 1/3$. We verify the performance of the proposed control law (6.126) with the maximum transmission rate set as $u_{\max} = 110$ packets. With equal rate partitioning, $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1/4$, the estimate of the maximum perturbation $\xi_{\max}$ calculated according to (6.120) amounts to $u_{\max}\beta \sum_{p=1}^{m} \lambda_p \bar{n}_p =$ 275 packets. A number of simulation tests are run for different bandwidth and RTT variations. In each case, the maximum bandwidth equals 100 packets.

Table 6.8 Controller parameters

Controller gain γ	Demand queue length y_D [packets]	Buffer size $y_{\max}$ [packets]
0.259	1,630 > 1,625	2,015
0.618	1,380 > 1,378	1,765
1	1,315 > 1,310	1,700

Test 1. In the first simulation scenario, we verify the controller performance in response to the available bandwidth evolving as depicted in Fig. 6.2. The delay in the feedback loop of each flow p is assumed to vary according to the following equation:

$$\mathrm{RTT}_p(k) = \left\lfloor \left[1 + \delta \sin\left(2\pi kT/\bar{n}_p\right)\right] \bar{n}_p T \right\rfloor, \tag{6.147}$$

where $\lfloor x \rfloor$ denotes the integer part of x. We run simulations for different dynamical settings adjusted through the gain constant γ. The demand queue length is set according to the guidelines of Theorem 6.18 to ensure full bandwidth utilization. In order to guarantee loss-free transmission, the buffer capacity at the node is assigned equal to the value indicated by Theorem 6.17. Parameters used in the test are summarized in Table 6.8.

The rate established by the controller is shown in Fig. 6.34 and the packet queue length in Fig. 6.35. It is apparent from the graphs in Fig. 6.35 that the queue length does not exceed the assigned buffer capacity and remains positive after the initial phase. This means that packet losses originating from congestion are eliminated and, at the same time, all of the available bandwidth is used for serving the data traffic. In consequence, the maximum throughput in the communication system is achieved. Moreover, even though the delay undergoes severe fluctuations (in the range of 33% from the nominal value), the degree of oscillations in the output variable (the queue length) remains negligible during the whole transmission process. Figures 6.34 and 6.35 indicate that with the increase in the controller gain, one obtains faster reaction to the changes of networking conditions. However, in the analyzed network with time-varying delay, better responsiveness comes at a price of amplified oscillations in the control signal, which occur even when the bandwidth reaches a constant steady-state value. Therefore, with the considered scheme applied, γ should not exceed 0.618, which corresponds to the weighting factor in the quadratic performance index equal to 1. This setting (the golden-ratio controller) reflects the case of equal penalization of the input signal and error at the output and constitutes a reasonable balance between the desirable smoothness of rate transitions and good responsiveness to the changing networking conditions.

Test 2. In the second simulation scenario, we test the controller performance in a stochastic setting. We assume that the available bandwidth evolves as shown in Fig. 6.14 (it follows the normal distribution with mean equal to 50 packets and standard deviation 35 packets). In the test, the RTT for each assignment is selected

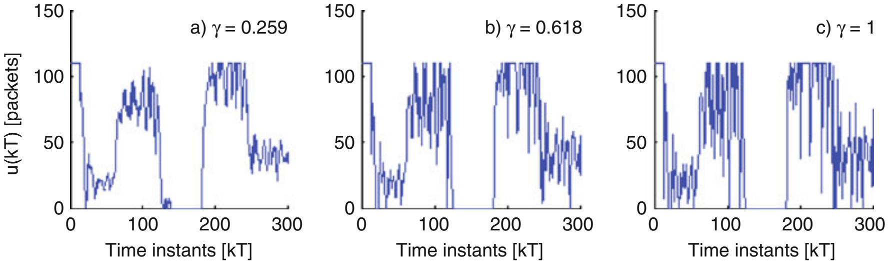


Fig. 6.34 Generated transmission rate: *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, and *c* $\gamma = 1$

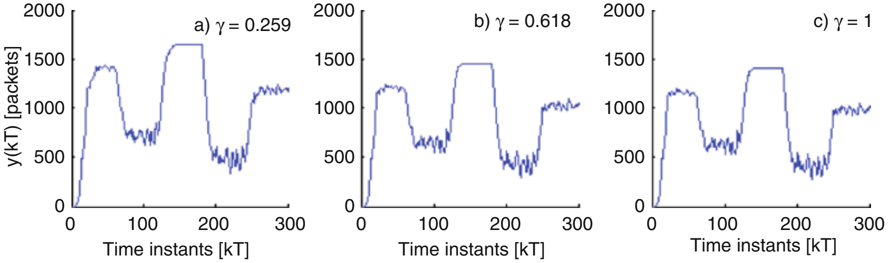


Fig. 6.35 Queue length: *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, and *c* $\gamma = 1$

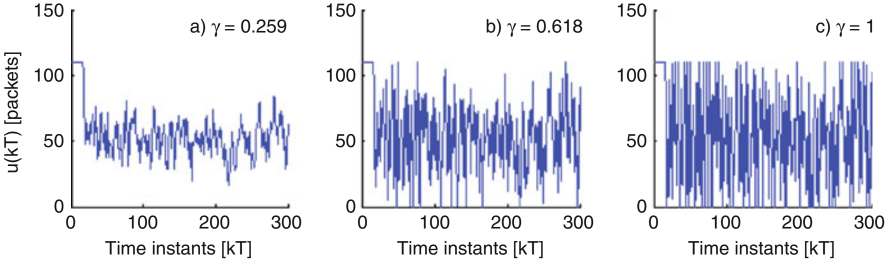


Fig. 6.36 Generated transmission rate: *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, and *c* $\gamma = 1$

as a random number from the interval $[(1-\beta)\bar{n}_p T,\ (1+\beta)\,\bar{n}_p T]$. The controller parameters are set as indicated in Table 6.8. The test results are shown in Figs. 6.36 and 6.37: the rate established by the controller in Fig. 6.36 and the packet queue length in Fig. 6.37.

In the stochastic setting, the controller maintains its properties related to handling the flow of data. The queue length illustrated in Fig. 6.37 remains within the allocated buffer space, which implies that packet losses are eliminated. It remains strictly positive after the initial phase. Therefore, full bandwidth usage is ensured and the system operates with maximum efficiency. If we compare the plots in Figs. 6.36 and 6.37 with the curves obtained for a similar network model with

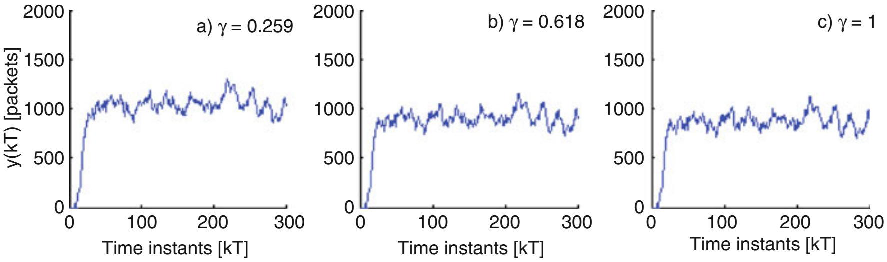


Fig. 6.37 Queue length: *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, and *c* $\gamma = 1$

constant delay presented in Figs. 6.21 and 6.22, we can notice a slightly increased rate of oscillations both in the control and in the output signal when the delay fluctuates. However, despite arbitrary bandwidth and delay variations, the system remains stable in the Lyapunov sense.

Test 3. In the earlier proposals for solving the flow control problem formulated for a similar network model with variable delay, for example, [3, 8–10], the essential system properties, such as stability, were defined in relation to the maximum latency experienced by the stream of packets. This brings the consequence of reducing the controller responsiveness (to the changes of networking conditions) in order to maintain stability if long delays are expected in the feedback loop. As a result, a single long-distance flow sharing the output link of the controlling node with local connections may throttle the entire system dynamics.

In the third simulation, we compare the operation of our scheme with one of the outstanding earlier solutions – the PID controller presented in [3]. The pool of connections considered in Tests 1 and 2 is augmented by the flow with the nominal delay equal to $24T$, i.e., the one which experiences two times bigger latency as compared with the longest flow considered in those tests. With the uncertainty $\beta = 1/3$, the maximum expected RTT amounts to $32T$. Since we are dealing with increased delay, the bandwidth pattern (and the duration of the simulation) is modified as shown in Fig. 6.38. For the clarity of exposition, we assume that both controllers apply equal rate partitioning. With the gain $\gamma = 0.618$, the demand queue length y_D is adjusted to $1{,}865 > 1{,}862$ packets, precisely as dictated by Theorem 6.18.

The results of operation of the PID (curve a) and SM (curve b) controllers are shown in Fig. 6.39. It is clear from the graphs that our scheme provides faster reaction to the changes of networking conditions without rendering the closed-loop system unstable. It also requires a smaller buffer to accommodate the entire packet queue (thus eliminating losses) and to keep the throughput at the maximum.

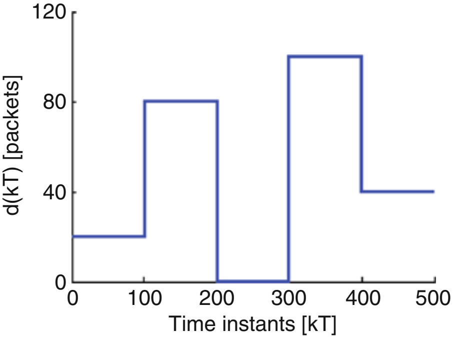


Fig. 6.38 Available bandwidth in Test 3

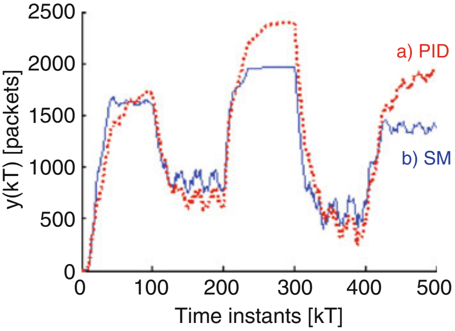


Fig. 6.39 Queue length: *a* PID and *b* SM controller

6.2.3 Delay Variability Compensation

In the previous section, we introduced a controller which provides robustness to delay variations in the multisource data transmission network modeled as an *n*th-order discrete-time system. The controller guarantees that the packet queue length at the controlling node is finite. With appropriately adjusted demand queue length, the controller also ensures that all of the available bandwidth at the node output link is used efficiently for data transfer. These properties are achieved despite arbitrary delay variations, including reordering and packet accumulation from several intervals in one discretization period. In contrast to other solutions reported in the past for a similar network model, robustness and stability provided by the proposed scheme depend on the average RTT of the served connections and not the maximum expected delay. In this way, the system dynamics (in particular the response to bandwidth changes) can be improved. However, it follows from the presented simulation example that by increasing the controller gain to boost the dynamics, one may amplify the oscillations of the control signal. As a result, in the case of more responsive controller configurations, the generated transmission rate may be difficult to follow by data sources, and the quality of the entire control process will degrade.

In this section, we apply the method of compensating the adverse effects of delay variations based on input rate measurements which was introduced in Chap. 5. We demonstrate that the proposed technique can eliminate oscillations caused by input distortion from the transmission rate signal established by the controller. In consequence, fast dynamics is provided without rendering the closed-loop system unstable. We also show that in the multisource network considered here, the input rate measurement may be based solely on the aggregate incoming rate values (without differentiating among the individual packet streams), thus ensuring scalability in the case of numerous active connections.

6.2.3.1 Proposed Control Strategy

In order to improve the smoothness of control signal generated in the network with time-varying RTTs, we modify the controller presented in Sect. 6.2.2 by incorporating the delay variability compensator. Consequently, the transmission rate is determined from the following equation:

$$u(kT) = \begin{cases} 0, & \text{if} \quad \omega_{\text{comp}}(kT) < 0, \\ \omega_{\text{comp}}(kT), & \text{if} \quad 0 \le \omega_{\text{comp}}(kT) \le u_{\max}, \\ u_{\max}, & \text{if} \quad \omega_{\text{comp}}(kT) > u_{\max}, \end{cases} \tag{6.148}$$

where $u_{\max} > d_{\max}$ and function $\omega_{\text{comp}}(\cdot)$ is defined as

$$\omega_{\text{comp}}(kT) = \gamma \left\{ y_{\text{D}} - y(kT) - \sum_{p=1}^{m} \lambda_p \sum_{j=k-\bar{n}_p}^{k-1} u(jT) + \varepsilon \sum_{j=0}^{k-1} \left[u_{\text{R}}(jT) - \bar{u}_{\text{R}}(jT) \right] \right\}. \tag{6.149}$$

The last sum in (6.149) represents the delay variability compensator in the network with multiple connections. The operation of the compensator is based on the measurements of the aggregate incoming rate $u_{\text{R}}(kT) = \sum_{p=1}^{m} u_{\text{R}}^{p}(kT)$ and comparison with the predicted incoming rate $\bar{u}_{\text{R}}(kT) = \sum_{p=1}^{m} \lambda_p u\left(kT - \overline{\text{RTT}}_p\right)$ calculated from the rate history stored in the node memory. The compensator structure is illustrated in Fig. 6.40. Note that implementation of the proposed mechanism does not require intensity measurement of the individual flows. Since only aggregate values are sufficient for DT compensation, scalability is ensured even for large number of packet streams. The influence of the compensation mechanism is adjusted through the tuning coefficient $\varepsilon \in [0, 1]$. Setting $\varepsilon = 1$ corresponds to the case of full compensation, and $\varepsilon = 0$ reflects the case of the compensation turned-off. Actually, with the compensator absent ($\varepsilon = 0$), the presented scheme reduces to controller (6.126) analyzed in the previous section.

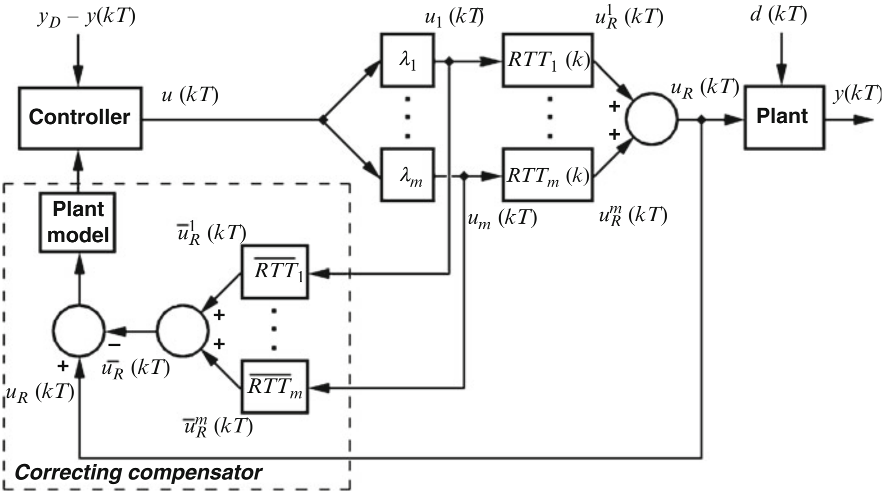


Fig. 6.40 Delay variability compensator in network with multiple connections

In the next section, we state the most important properties of the developed controller (6.148), and we prove them analytically. We will show that by incorporating the proposed delay variability compensator (6.149), the presented scheme can retain good system dynamics and high bandwidth utilization without the risk of loosing stability. In fact, the closed-loop stability is ensured for any delay and any bounded variation pattern satisfying condition (6.112).

6.2.3.2 Properties of the Proposed Strategy

The properties of the proposed control strategy are given in two theorems. The first proposition demonstrates that the packet queue length in the controlling node buffer is finite. The second one shows how the demand queue length should be selected to ensure full bandwidth utilization.

Theorem 6.19. *If controller (6.148) with function $\omega_{comp}(\cdot)$ defined by (6.149) is applied to system (6.123) and (6.124), then the queue length in the bottleneck node buffer is always upper-bounded, i.e., for any $k \geq 0$,*

$$y\,(kT) \leq y_{\max} = y_{\mathrm{D}} + u_{\max} + (1+\varepsilon)\,\xi_{\max}. \tag{6.150}$$

Proof. Using (6.116), (6.121), and the definition of function $\xi(\cdot)$, we may notice that the term compensating the effects of delay variations in (6.149) satisfies the following relation:

$$\begin{aligned}\sum_{j=0}^{k-1}\left[u_{\mathrm{R}}(jT)-\bar{u}_{\mathrm{R}}(jT)\right] &= \sum_{j=0}^{k-1}\left\{\sum_{p=1}^{m}\lambda_p u\left[jT-\mathrm{RTT}_p(j)\right]-\sum_{p=1}^{m}\lambda_p u\left(jT-\overline{\mathrm{RTT}}_p\right)\right\}\\ &= \sum_{j=0}^{k-1}\xi(jT)=\xi(kT).\end{aligned} \tag{6.151}$$

Therefore, function $\omega_{\mathrm{comp}}(\cdot)$ can be rewritten as

$$\varphi_{\mathrm{comp}}(kT)=\gamma\left[y_{\mathrm{D}}-y(kT)-\sum_{p=1}^{m}\lambda_p\sum_{j=k-\bar{n}_p}^{k-1}u(jT)+\varepsilon\xi(kT)\right]. \tag{6.152}$$

It follows from the system initial conditions that the buffer at the bottleneck node is empty for any $k \leq (1-\beta)\bar{n}_1$. Consequently, in order to prove the theorem, we need to show that $y(kT) \leq y_{\max}$ for all $k > (1-\beta)\bar{n}_1$. Let us consider some integer $l > (1-\beta)\bar{n}_1$ and the value of $\omega_{\mathrm{comp}}(\cdot)$ at instant lT. Two cases ought to be analyzed: the situation when $\omega_{\mathrm{comp}}(lT) \geq 0$ and the circumstances when $\omega_{\mathrm{comp}}(lT) < 0$.

Case 1. We investigate the case when $\omega_{\mathrm{comp}}(lT) \geq 0$. Directly from (6.152), we get

$$\omega_{\mathrm{comp}}(lT)=\gamma\left[y_{\mathrm{D}}-y(lT)-\sum_{p=1}^{m}\lambda_p\sum_{j=l-\bar{n}_p}^{l-1}u(jT)+\varepsilon\xi(lT)\right]\geq 0, \tag{6.153}$$

which leads to

$$y(lT)\leq y_{\mathrm{D}}+\varepsilon\xi(lT)-\sum_{p=1}^{m}\lambda_p\sum_{j=l-\bar{n}_p}^{l-1}u(jT). \tag{6.154}$$

It follows from the algorithm definition that $u(\cdot)$ is always nonnegative, which implies $y(lT) \leq y_{\mathrm{D}} + \varepsilon\xi(lT)$. Moreover, since $\xi(lT) \leq \xi_{\max}$, we obtain $y(lT) \leq y_D + \varepsilon\xi_{\max} < y_{\max}$, which ends the first part of the proof.

Case 2. In the second part of the proof, we analyze the situation when $\varphi_{\mathrm{comp}}(lT) < 0$. First, we find the most recent instant $l_1T < lT$ when $\varphi_{\mathrm{comp}}(\cdot)$ was nonnegative. According to (6.152), $\varphi_{\mathrm{comp}}(0) = \gamma y_{\mathrm{D}} > 0$, so the moment l_1T indeed exists and the value of $y(l_1T)$ satisfies an inequality similar to (6.154), i.e.,

$$y(l_1T)\leq y_{\mathrm{D}}+\varepsilon\xi(l_1T)-\sum_{p=1}^{m}\lambda_p\sum_{j=l_1-\bar{n}_p}^{l_1-1}u(jT). \tag{6.155}$$

The queue length at instant lT can be expressed relative to $y(l_1T)$ as in (6.132), which after applying (6.155) gives

$$y(lT) \le y_{\mathrm{D}} + \varepsilon\xi(l_1T) - \sum_{p=1}^{m}\lambda_p \sum_{j=l_1-\bar{n}_p}^{l_1-1} u(jT)$$
$$+ \sum_{p=1}^{m}\lambda_p \sum_{j=l_1-\bar{n}_p}^{l-\bar{n}_p-1} u(jT) + \xi(lT) - \sum_{j=l_1}^{l-1} h(jT). \tag{6.156}$$

Performing algebraic manipulations on the sums in (6.156), we arrive at

$$y(lT) \le y_{\mathrm{D}} + \varepsilon\xi(l_1T) + \xi(lT) + \sum_{p=1}^{m}\lambda_p \sum_{j=l_1}^{l-\bar{n}_p-1} u(jT) - \sum_{j=l_1}^{l-1} h(jT). \tag{6.157}$$

The controller set a nonzero transmission rate for the last time before lT at instant l_1T, and this value could have been as large as $u_{\max}$. Consequently, the sum $\sum_{p=1}^{m}\lambda_p \sum_{j=l_1}^{l-\bar{n}_p-1} u(jT) = u(l_1T) \le u_{\max}$. Since the utilized bandwidth is always nonnegative, then using the fact that for arbitrary time $\xi(\cdot) \le \xi_{\max}$, we obtain from (6.157) the following estimate of the queue length at instant lT:

$$y(lT) \le y_{\mathrm{D}} + \varepsilon\xi(l_1T) + \xi(lT) + u_{\max} - 0$$
$$\le y_{\mathrm{D}} + (1+\varepsilon)\,\xi_{\max} + u_{\max} = y_{\max}. \tag{6.158}$$

This concludes the second part of the reasoning and completes the proof. □

Theorem 6.19 shows that the packet queue length under the control of the proposed strategy remains finite and never exceeds the level of $y_{\max}$. This means that if the buffer of capacity $y_{\max}$ is reserved at the controlling node, then irrespective of bandwidth and delay variations, packet losses are eliminated. The second proposition, given below, demonstrates that with appropriately selected demand queue length y_{D}, the queue length $y(\cdot) > 0$, which guarantees full bandwidth utilization.

Theorem 6.20. *If controller (6.148) with function $\omega_{comp}(\cdot)$ defined by (6.149) is applied to system (6.123) and (6.124), and the demand queue length satisfies the following inequality:*

$$y_{\mathrm{D}} > u_{\max}\left(\sum_{p=1}^{m}\lambda_p\bar{n}_p + \frac{1}{\gamma}\right) + d_{\max} + (1+\varepsilon)\,\xi_{\max}, \tag{6.159}$$

then for any $k \ge (1+\beta)\bar{n}_m + n_{\max}$, where $n_{\max} = y_{\max}/(u_{\max} - d_{\max})$, the queue length is strictly positive.

Proof. The theorem assumption implies that we deal with time instants $kT \geq (1+\beta)\bar{n}_m T + n_{\max} T$. Taking some integer $l \geq (1+\beta)\bar{n}_m + n_{\max}$ and the value of signal $\omega_{\text{comp}}(\cdot)$ at instant lT, we need to consider two cases: the situation when $\omega_{\text{comp}}(lT) < u_{\max}$ and the circumstances when $\omega_{\text{comp}}(lT) \geq u_{\max}$.

Case 1. First, we analyze the case when $\omega_{\text{comp}}(lT) < u_{\max}$. From (6.152), we obtain

$$\omega_{\text{comp}}(lT) = \gamma \left[y_{\text{D}} - y(lT) - \sum_{p=1}^{m} \lambda_p \sum_{j=l-\bar{n}_p}^{l-1} u(jT) + \varepsilon \xi(lT) \right] < u_{\max}, \tag{6.160}$$

which after the term rearrangement reduces to

$$y(lT) > y_{\text{D}} - u_{\max}/\gamma - \sum_{p=1}^{m} \lambda_p \sum_{j=l-\bar{n}_p}^{l-1} u(jT) + \varepsilon \xi(lT). \tag{6.161}$$

The transmission rate is always bounded by $u_{\max}$, which implies

$$y(lT) > y_{\text{D}} - u_{\max}/\gamma - u_{\max} \sum_{p=1}^{m} \lambda_p \bar{n}_p + \varepsilon \xi(lT). \tag{6.162}$$

Since $\xi(\cdot) \geq -\xi_{\max}$, we get

$$y(lT) > y_{\text{D}} - u_{\max}/\gamma - u_{\max} \sum_{p=1}^{m} \lambda_p \bar{n}_p - \varepsilon \xi_{\max}. \tag{6.163}$$

Using assumption (6.159), we get $y(lT) > 0$, which concludes the first part of the proof.

Case 2. In the second part of the proof, we analyze the situation when $\omega_{\text{comp}}(lT) \geq u_{\max}$. First, we find the last moment $l_1 T < lT$ when function $\omega_{\text{comp}}(\cdot)$ was smaller than $u_{\max}$. It comes from Theorem 6.19 that the queue length never exceeds the value of $y_{\max}$. Moreover, we know that the consumed bandwidth is limited by $d_{\max}$. Thus, the maximum interval $n_{\max}T$ in which the controller may continuously generate the maximum transmission rate for the sources is determined as $n_{\max}T = T y_{\max}/(u_{\max} - d_{\max})$, and instant $l_1 T$ does exist. Furthermore, from the theorem assumptions we get $l_1 T \geq (1+\beta)\bar{n}_m T$, which means that by the time $l_1 T$, the first packets reach the node no matter the delay variation in any of the flows.

The value of $\omega_{\mathrm{comp}}(l_1 T) < u_{\max}$. Thus, following similar reasoning as presented in (6.160)–(6.163), we arrive at $y(l_1 T) > 0$ and

$$y\,(lT) > y_{\mathrm{D}} - \frac{u_{\max}}{\gamma} - \sum_{p=1}^{m} \lambda_p \sum_{j=l_1-\bar{n}_p}^{l_1-1} u\,(jT) + \varepsilon\xi\,(l_1 T) \\ + \sum_{p=1}^{m} \lambda_p \sum_{j=l_1-\bar{n}_p}^{l-\bar{n}_p-1} u\,(jT) + \xi\,(lT) - \sum_{j=l_1}^{l-1} h\,(jT). \tag{6.164}$$

Working with the sums in (6.164), we get

$$\sum_{p=1}^{m} \lambda_p \sum_{j=l_1-\bar{n}_p}^{l-\bar{n}_p-1} u\,(jT) - \sum_{p=1}^{m} \lambda_p \sum_{j=l_1-\bar{n}_p}^{l_1-1} u\,(jT) = \sum_{p=1}^{m} \lambda_p \sum_{j=l_1}^{l-1} u\,(jT) \\ - \sum_{p=1}^{m} \lambda_p \sum_{j=l-\bar{n}_p}^{l-1} u\,(jT) = \sum_{j=l_1}^{l-1} u\,(jT) - \sum_{p=1}^{m} \lambda_p \sum_{j=l-\bar{n}_p}^{l-1} u\,(jT) \tag{6.165}$$

Instant $l_1 T$ was the last one before lT when the controller calculated rate smaller than $u_{\max}$. This rate could be as low as zero. Afterwards, the controller generates the maximum rate, and the first sum in (6.165) reduces to $u_{\max}(l - 1 - l_1)$. Since for any k, $u(kT) \le u_{\max}$, the second sum is upper-bounded by $u_{\max} \sum_{p=1}^{m} \lambda_p \bar{n}_p$. Consequently, we obtain the following estimate of the queue length at instant lT:

$$y\,(lT) > y_{\mathrm{D}} - \frac{u_{\max}}{\gamma} + u_{\max}\,(l - 1 - l_1) - u_{\max} \sum_{p=1}^{m} \lambda_p \bar{n}_p + \varepsilon\xi\,(l_1 T) \\ + \xi\,(lT) - \sum_{j=l_1}^{l-1} h\,(jT). \tag{6.166}$$

Using the fact that $\xi(\cdot) \ge -\xi_{\max}$ and $h(\cdot) \le d_{\max}$, we get further estimate of $y(lT)$:

$$y\,(lT) > y_{\mathrm{D}} - u_{\max}/\gamma + u_{\max}\,(l - 1 - l_1) - u_{\max} \sum_{p=1}^{m} \lambda_p \bar{n}_p \\ - \varepsilon\xi_{\max} - \xi_{\max} - d_{\max}\,(l - l_1)\,. \tag{6.167}$$

Finally, using the theorem assumption (6.159), we get

$$\begin{aligned} y\left(lT\right) >& u_{\max} \sum_{p=1}^{m} \lambda_p \bar{n}_p + u_{\max}/\gamma + d_{\max} + (1+\varepsilon)\,\xi_{\max} - u_{\max}/\gamma \\ & + u_{\max}\left(l-1-l_1\right) - u_{\max} \sum_{p=1}^{m} \lambda_p \bar{n}_p - (1+\varepsilon)\,\xi_{\max} - d_{\max}\left(l-l_1\right) \\ =& u_{\max}\left(l-1-l_1\right) - d_{\max}\left(l-1-l_1\right). \end{aligned} \tag{6.168}$$

Since $l > l_1$ and $u_{\max} > d_{\max}$, we arrive at $y(lT) > (u_{\max} - d_{\max})(l-1-l_1) \geq 0$. This conclusion ends the proof. □

In the next section we present the results of simulations demonstrating the properties of the proposed controller (6.148). In particular, we comment on the control quality improvements originating from the application of delay variability compensator.

6.2.3.3 Simulation Results

We assume that four flows ($m = 4$) pass through the bottleneck node and are subject to the control action. The flows are characterized by the following nominal RTTs: $\overline{\mathrm{RTT}}_1 = 3T$, $\overline{\mathrm{RTT}}_2 = 7T$, $\overline{\mathrm{RTT}}_3 = 8T$, and $\overline{\mathrm{RTT}}_4 = 12T$, where the discretization period $T = 10$ ms. The actual delay in the feedback loop varies in a way unknown to the controller in the range specified by $\beta = 1/3$. Performance of controller (6.148) is verified for different dynamical settings with the gain chosen as either 0.259, 0.618, or 1. In each test, the maximum transmission rate is set as $u_{\max} = 110$ packets. Three series of simulations are run: two for the bandwidth illustrated in Fig. 6.2 and one for the stochastic pattern depicted in Fig. 6.14. In the simulations, it is assumed that the rate is distributed evenly among the flows: $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1/4$, and the maximum bandwidth is adjusted to $d_{\max} = 100$ packets.

Test 1. In the first simulation scenario we verify the controller performance in response to the available bandwidth depicted in Fig. 6.2. The delay is subject to sinusoidal variations described by (6.147). We run several simulations for different gain settings. In each simulation, full compensation of delay variations is applied, i.e., $\varepsilon = 1$. In order to guarantee that all of the available bandwidth is used efficiently, the demand queue length is selected according to (6.159). The buffer capacity is adjusted according to (6.150) so that loss-free transmission can be ensured. Parameters used in the test are summarized in Table 6.9.

The test results are shown in Figs. 6.41–6.43: the transmission rate established by the controller in Fig. 6.41, the incoming packet stream in Fig. 6.42, and the buffer queue length in Fig. 6.43. We can see from the graphs that the proposed controller quickly responds to the sudden changes of the available bandwidth. In contrast to

Table 6.9 Controller parameters

Controller gain γ	Demand queue length y_D [packets]	Buffer size y_{max} [packets]
0.259	1,905 > 1,900	2,565
0.618	1,655 > 1,653	2,315
1	1,590 > 1,585	2,250

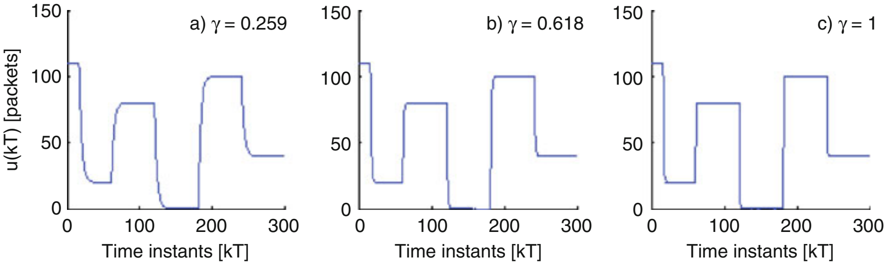


Fig. 6.41 Generated transmission rate: *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, and *c* $\gamma = 1$

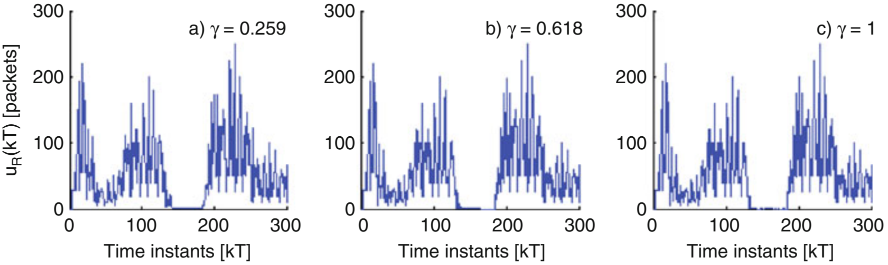


Fig. 6.42 Incoming rate: *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, and *c* $\gamma = 1$

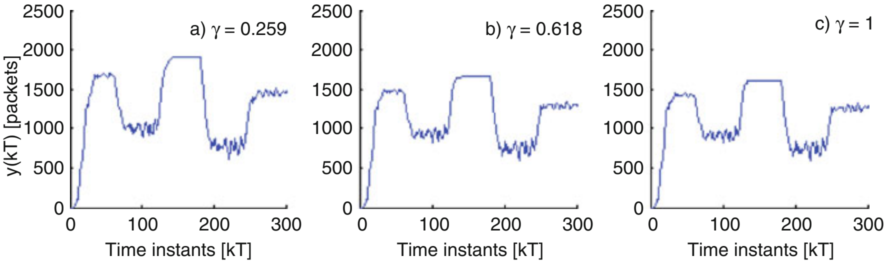


Fig. 6.43 Queue length: *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, and *c* $\gamma = 1$

controller (6.126) analyzed in the previous section for the identical network model, controller (6.148) does not allow the delay variations to disrupt the generated control signal. This clearly demonstrates the efficiency of the proposed compensation mechanism in protecting the rate signal from contamination by the effect of delay changes. In fact, the applied DT compensator regenerates the input signal in the network with uncertain, time-varying delay as if RTTs were fixed and known to the controller. On the other hand, it is clear from Fig. 6.43 that the queue length does not increase beyond the buffer capacity specified in Table 6.9, and it does not fall to zero following the initial phase. As a result, packet losses are eliminated and maximum throughput is guaranteed. Comparing the queue length evolution depicted in Fig. 6.35, which results from the application of controller (6.126), with the one obtained here and given in Fig. 6.43, we can see a similar degree of oscillations which are caused by the disrupted incoming rate. The proposed delay variability compensator regenerates the control signal, but, obviously, it has no influence on the pattern of RTT oscillations. As a result, it cannot guard the output variable $y(\cdot)$ against the incoming rate perturbations caused by delay changes.

Test 2. In the second simulation scenario, the controller robustness is verified against an external, multiplicative perturbation $\pi(\cdot)$ acting in the input channel. The value of $\pi(\cdot) > 1$ reflects the situation of nonconforming sources (which transmit data at a rate greater than the assigned one), and when $\pi(\cdot) < 1$, some transmitters experience data shortage or are blocked due to multibottleneck congestion. Therefore, in addition to delay variations, the incoming packet rate is subject to extra unknown disturbance related to the distributed congestion and source nonideality. We assume that function $\pi(\cdot)$ follows the normal distribution with mean equal to 1 and variance 0.05, which results in the values of $\pi(\cdot)$ in the range [0.4, 1.6]. Performance of controller (6.148) is verified for the case of the compensation turned-off a and with full compensation applied b. The controller parameters are set as follows: the gain $\gamma = 0.618$, the maximum rate $u_{\max} = 110$ packets, and the demand queue length $y_{\mathrm{D}} = 1{,}655$ packets (see Table 6.9). Two series of simulations are conducted: in the first one, RTTs are assumed constant and equal to the value known to the controller, whereas in the second test, the delay exhibits sinusoidal oscillations (6.147).

The rate generated by the controller is shown in Fig. 6.44, the perturbed incoming rate in Fig. 6.45, and the packet queue length in Fig. 6.46. The plots in Fig. 6.44 demonstrate that the proposed DT compensator successfully counteracts all the perturbations acting in the input channel. The rate established by the controller in the fully compensated case is smooth and nonoscillatory. Moreover, although the compensator cannot prevent the external perturbation from altering the packet incoming rate (obviously, it can anticipate neither the exact form of source nonidealities nor packet losses occurring at some other node in the network), it improves the system robustness. It can be seen from Fig. 6.46 that the queue length in the case of the compensation applied is less oscillatory and always remains within the assigned buffer space.

Test 3. In the third simulation scenario, the controller performance is verified in a stochastic setting with the bandwidth following the normal distribution with mean $d_{\mu} = 50$ packets and standard deviation $d_{\delta} = 35$ packets, illustrated in Fig. 6.14. The delay is subject to uniform variations in the interval $[(1-\beta)\bar{n}_p T,\, (1+\beta)\bar{n}_p T]$.

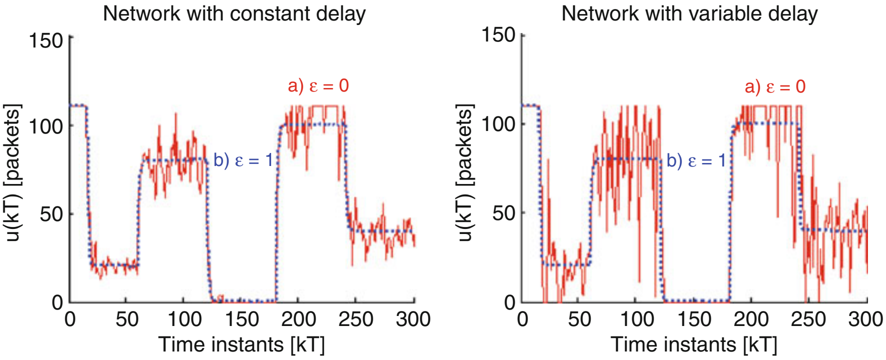


Fig. 6.44 Generated transmission rate: *a* without compensation, $\varepsilon = 0$, and *b* with full compensation applied, $\varepsilon = 1$

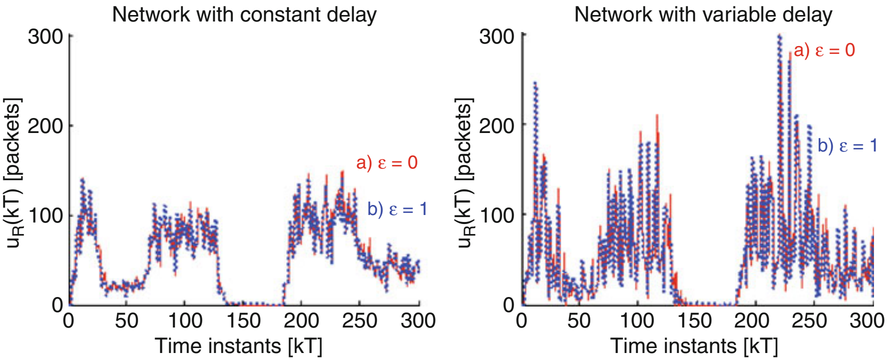


Fig. 6.45 Incoming rate: *a* without compensation and *b* with full compensation applied

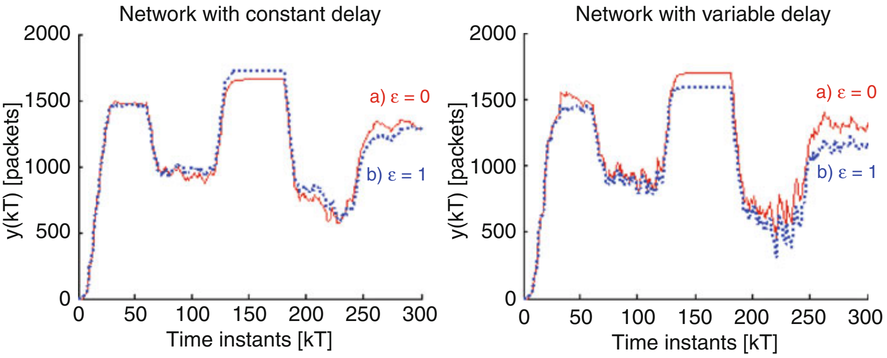


Fig. 6.46 Queue length: *a* without compensation and *b* with full compensation applied

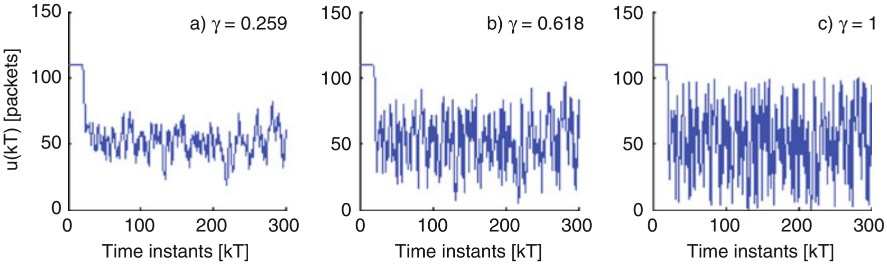


Fig. 6.47 Generated transmission rate: *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, and *c* $\gamma = 1$

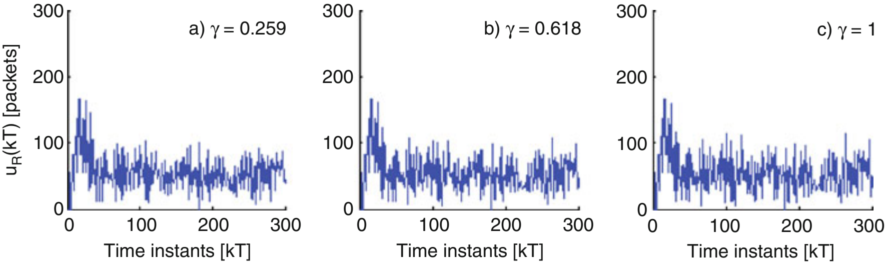


Fig. 6.48 Incoming rate: *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, and *c* $\gamma = 1$

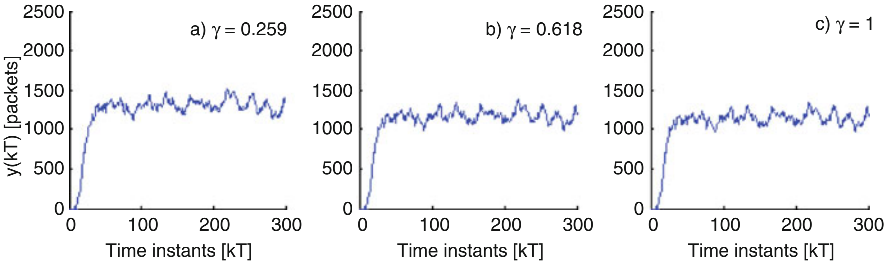


Fig. 6.49 Queue length: *a* $\gamma = 0.259$, *b* $\gamma = 0.618$, and *c* $\gamma = 1$

In the tests, full delay variability compensation is applied, i.e., $\varepsilon = 1$. The controller parameters are set exactly as in Test 1. Their values are listed in Table 6.9.

The test results are illustrated in Figs. 6.47–6.49: the transmission rate established by the controller in Fig. 6.47, the incoming packet rate in Fig. 6.48, and the buffer occupancy in Fig. 6.49. The analysis of the plots from Fig. 6.47 shows that by decreasing the gain, one obtains less oscillatory control signal in response to fluctuating bandwidth. Moreover, it follows from the plots shown in Fig. 6.49 that the queue length always stays within the assigned buffer space, which means that packet losses are eliminated. Following the initial phase, $y(\cdot) > 0$, which implies full bandwidth utilization and maximum throughput.

6.3 Chapter Summary

In this chapter, we addressed the problem of data flow control in a communication network serving multiple connections. We emphasized the need for explicit delay treatment in the network modeling and control. In order to account for the discrete nature of crucial networking events, such as transmission rate calculation by the node and rate update at the data sources, both the modeling and the controller design were conducted directly in discrete-time domain.

In Sect. 6.1, we focused on the fundamental control process of the network serving multiple flows characterized by different, yet constant, RTTs. The network was modeled as an nth-order discrete-time system. A number of control laws were designed using the principles of discrete-time SMC. First, dynamical optimization with quadratic performance index was applied to the unconstrained model to find the sliding-plane parameters for a linear SM controller. It was noted that in the network with flows characterized by different RTTs, the optimal solution generates negative transmission rates, which is clearly infeasible. Therefore, a suboptimal controller was developed in the modified optimization procedure. The modified LQ problem was solved analytically, which allowed us to express the obtained control law in a closed-form. A number of advantageous properties of the proposed controller were formulated and proved. First of all, we showed that the transmission rate established by the obtained control law is always nonnegative and bounded, thus being feasible in network implementation. Secondly, it was demonstrated that the packet queue length in the bottleneck node buffer remains finite, which leads to loss-free transmission if the buffer capacity is selected according to the indicated queue length bound. It was also shown that the queue length under the control of the presented scheme is strictly positive. This condition implies that all of the available bandwidth is efficiently consumed for data transfer, and, as a result, the maximum throughput in the system is achieved. Next, the fundamental controller design was adapted to the network with transmission rate limitations. Three methods of enforcing the input to fit in the specified range were discussed: the application of a time-varying sliding plane, reaching-law-based design, and the use of a saturation element. Each method was demonstrated to ensure loss elimination and full bandwidth usage in addition to satisfying the imposed input constraint. A significant advantage of the proposed controllers is separation of the functions related to flow regulation from fairness control. In the last part of Sect. 6.1, we analyzed network configuration with arbitrary, time-varying rate allocation and variable connection number. We showed that different fairness criteria can be incorporated in the overall control scheme while maintaining high efficiency of data flow control, which is imperative in modern networks.

In Sect. 6.2, we extended the model of the multisource network introduced in Sect. 6.1 for the case of time-varying delays. The presented concept is shown to be general enough to reflect any bounded latency fluctuations, including the issues related to packet and feedback information carriers reordering. For the described model, two robust control laws were proposed. Both laws combine the benefits of discrete-time SM controller with LQ suboptimal sliding plane and bounding nonlinearity of the saturation element. The communication system under the control of the proposed schemes is stable and transmits data with maximum efficiency. Delay variations are shown to constitute a limitation for the first controller when one attempts to increase the system dynamics. Thanks to the applied delay variability compensator, the second controller is insensitive to distortions of the incoming rate in the input channel and generates smooth control signal in any dynamical setting. In consequence, the established transmission rate is easier to follow by the sources, and the quality of network traffic control can be improved.

The rate calculation performed by each of the discussed controllers is based on simple arithmetic (additions and multiplications) and logic (comparison with a constant) operations. Thus, it can be efficiently implemented either at the software or hardware level (e.g., with the use of shift registers and comparators).

References

1. Bartoszewicz A (1996) Remarks on 'discrete-time variable structure control systems'. IEEE Trans Ind Electron 43:235–238
2. Bartoszewicz A (1998) Discrete time quasi-sliding mode control strategies. IEEE Trans Ind Electron 45:633–637
3. Blanchini F, Lo Cigno R, Tempo R (2002) Robust rate control for integrated services packet networks. IEEE/ACM Trans Netw 10:644–652
4. Draženović B (1969) The invariance conditions in variable structure systems. Automatica 5:287–295
5. Gao W, Wang Y, Homaifa A (1995) Discrete-time variable structure control systems. IEEE Trans Ind Electron 42:117–122
6. Golo G, Milosavljević Č (2000) Robust discrete-time chattering free sliding mode control. Syst Control Lett 41:19–28
7. Katabi D, Handley M, Rohrs ChE (2002) Congestion control for high bandwidth-delay product networks. In: Proceedings of the ACM SIGCOMM, Pittsburgh, USA, pp 89–102
8. Pietrabissa A, Delli Priscoli F, Fiaschetti A, Di Paolo F (2006) A robust adaptive congestion control for communication networks with time-varying delays. In: Proceedings of the 2006 IEEE international conference control applications, Munich, Germany, pp 2093–2098
9. Quet PF, Ataşlar B, İftar A, Özbay H, Kalyanaraman S, Kang T (2002) Rate-based flow controllers for communication networks in the presence of uncertain time-varying multiple time-delays. Automatica 38:917–928
10. Ren T, Gao Z, Kong W, Jing Y, Yang M, Dimirovski GM (2009) Performance and robustness analysis of a fuzzy immune flow controller in ATM networks with time-varying multiple time delays. J Control Theory Appl 6:253–258
11. Su WCh, Drakunov SV, Özgüner Ü (2000) An O(T^2) boundary layer in sliding mode for sampled-data systems. IEEE Trans Autom Control 45:482–485

Chapter 7
Flow Control in Sampled Data Systems

For telecommunication operators, the cost of running a particular algorithm is an important factor when deciding about its applicability in the supervised network. Therefore, the business point of view will generally favor such control strategies which allow for the explicit specification (or at least estimation) of the amount of the exchanged feedback information in relation to the transferred users' data. In consequence, emitting feedback information carriers at regular time intervals (which is assumed in traditional approaches for discrete-time network modeling, e.g., [2–5]) is not cost efficient in the economical terms. Instead, one can send a feedback information carrier every N data packets and, in this way, place a direct limit on the extent of the transmitted management traffic with respect to the profit generating transmission of the users' data. Since this method relies neither on continuous feedback information availability nor on maintaining the synchronization of constant sampling period (which is a serious challenge in multisource systems [1]), it is more scalable and requires less control effort than the classical regulation schemes presented in the literature. However, sending a control unit every N data packets, and not every T seconds, means that (after RTT) the feedback information will be available for rate adaptation at the sources at irregular time instants. Consequently, in order to maintain adequate system performance, the variable, input-dependent sampling period should be explicitly accounted for in the design of flow control algorithm. We will show, however, that the controllers developed for the system with constant discretization period in Chaps. 5 and 6 can be quite intuitively adapted for the case of variable sampling rate analyzed here. Moreover, we will demonstrate that provided that certain additional constraints are met, the new strategies maintain the favorable properties of constant-sampling-rate controllers. In particular, the proposed schemes will be shown to eliminate packet losses originating from unknown bandwidth variations and to ensure full bandwidth usage.

The chapter is organized in the following way. First, in Sect. 7.1, the model of the network with variable discretization period is described with the emphasis placed on the principles of feedback information exchange. Afterwards, in Sect. 7.2, the

P. Ignaciuk and A. Bartoszewicz, *Congestion Control in Data Transmission Networks: Sliding Mode and Other Designs*, Communications and Control Engineering,
DOI 10.1007/978-1-4471-4147-1__7, © Springer-Verlag London 2013

fundamental control strategy obtained from the LQ optimization conducted in the previous chapter is reformulated for the case of nonuniform sampling. Appropriate conditions allowing loss-free transmission and full bandwidth utilization are formulated and proved. The stable operation of the controller is guaranteed even though the sources adapt their rate at irregular (input dependent) time instants. In that section, also the robustness issues related to imprecise delay calculation are addressed. It is demonstrated that the algorithm maintains its properties despite possible mismatch between the real RTTs and those estimated by the controller in the connection setup phase. Similarly as in Chaps. 5 and 6, the enhanced robustness is achieved at the expense of increased buffer size. In Sect. 7.3, a modified strategy employing bandwidth compensation is presented. It is shown that this strategy with feed-forward component can assure insensitivity of the steady-state queue length with respect to the available bandwidth. In consequence, the delay jitter (delay variation of subsequent fragments of a data stream) is decreased, which allows for QoS improvements in the communication system. As the delay jitter is reduced, the network designed for serving best-effort traffic becomes also suitable for the transfer of delay-sensitive multimedia streams. Finally, in Sect. 7.4, the problem of maintaining high bandwidth utilization in the network with nonpersistent data sources is addressed. The properties of the proposed strategies are verified by simulation examples presented at the end of each section. Finally, the overall concluding remarks are given in Sect. 7.5.

7.1 Network Model

The communication network considered in Chaps. 5 and 6 was modeled as a purely discrete-time system. In real networks, however, the primary network variables, such as the source rate or the buffer queue length, are measurable at any time instant and thus are better described by functions continuous with respect to time. Nevertheless, the feedback information used to control the input rate remains accessible at discrete time instants only. This, in principle, makes the input rate a sampled-time signal with the values adapted at the instants of feedback information retrieval. Consequently, the model of multisource single-bottleneck network presented in Sect. 6.1.1 needs to be extended to reflect the real nature of signals describing the primary network variables in a sampled data system.

Similarly as in the model introduced in the previous chapter, it is assumed here that the feedback mechanism for the transfer rate adaptation of m data flows passing through the bottleneck node is provided by means of feedback information carriers – control units. Control units are generated by the sources, delivered to the destinations with priority over data packets, and finally returned to the sources with the feedback information incorporated. However, unlike the situation described in Chap. 6, control units are emitted by the sources not every T but after sending N data packets. Since control units are generated every N ordinary packets, the time period between their arrivals at the sources depends on the emission rate

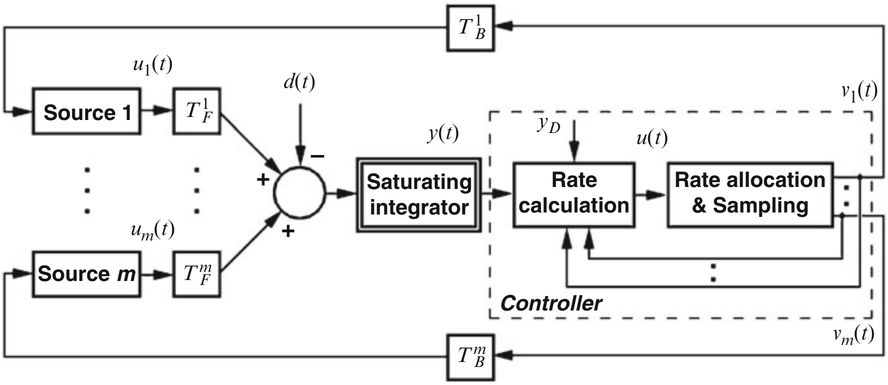


Fig. 7.1 Network model – sampled data system with multiple virtual circuits

RTT earlier, which, in turn, changes according to the variations of the network state, and makes the feedback asynchronous. To protect the communication system from a management standstill in the case of transmission rate falling to zero, it is assumed that each source probes the network at least every T_{C}, where T_{C} denotes the maximum control interval.

The presented model is illustrated in Fig. 7.1. The fundamental quantities indicated in the graph are similar to the ones already introduced in the previous chapters. The primary difference is the modified feedback mechanism (control units sent at irregular time instants), and the time reference changed from the discrete variable kT to the continuous one t. For convenience, the description of the plant dynamics is recalled below with the emphasis placed on the indicated model changes.

With reference to Fig. 7.1, the flow control process in the analyzed network goes through the following stages. Once a connection, say, p ($p = 1, 2, \ldots, m$), is accepted by the network, the corresponding source sends data at the rate $u_p(t)$. The packets belonging to the flow originating at source p reach the bottleneck node with forward delay T_{F}^p and are transmitted towards the destination according to the bandwidth availability at the node output link $d(t)$. The remaining data accumulates in the queue stored in the buffer allocated for the output link. The controller placed at the bottleneck node compares the current queue length, which at time t will be denoted as $y(t)$, with its demand value y_{D}, and calculates the aggregate transmission rate $u(t)$ for the sources. An appropriate part of the total rate, $v_p(t)$, is recorded as the feedback information in every control unit belonging to flow p passing through the node. Once the control units from source p appear at the destination, they are turned back to arrive at their origin with backward delay T_{B}^p after being processed by the bottleneck node. The source extracts information about the assigned flow rate from the received control units and adjusts the transfer speed accordingly. Since control units are not subject to the queuing delays as they are served with priority over data packets, their round-trip time $\mathrm{RTT}_p = T_{\mathrm{F}}^p + T_{\mathrm{B}}^p$ is assumed to remain constant for the duration of connection for each flow (the case of variable RTT may be analyzed in a similar way as was presented in Sect. 6.2). It should be stressed that the assumption about constant RTT concerns only the priority-served control units and not the packets carrying data that are subject to the standard queuing delays in the buffers along the established data route.

The available bandwidth at the output link of the bottleneck node, $d(t)$, cannot be determined *a priori*. It is only known that it does not exceed a maximum $d_{\max}$. We expect $d_{\max}$ to be greater than mN/T_C to ensure that at least during certain periods of time the individual sources will have a chance to transmit data at a rate greater than N/T_C (i.e., more than N packets in the maximum control interval). If at time t there are enough packets ready for transmission in the bottleneck link buffer, then the bandwidth actually consumed by all the sources, $h(t)$, will be equal to the available one. Otherwise, the output link is underutilized and the consumed bandwidth matches the data arrival rate at the bottleneck node. Consequently, similarly as in the previous chapters, we obtain the following relation involving $d(\cdot)$ and $h(\cdot)$ at any time instant t:

$$0 \le h(t) \le d(t) \le d_{\max}.$$

The rate of source p is determined by the controller placed at the bottleneck node. Let us denote by $v_p(t)$ the rate calculated by the controller and sent for source p at the instant of a control unit passing through the node. Assuming that the sources begin transmission at the time instant $t = 0$ at the rate established in the connection setup phase, the following is true:

$$\underset{p}{\forall}\ \underset{t<0}{\forall}\ u_p(t) = 0 \quad \text{and} \quad \underset{p}{\forall}\ \underset{t\ge 0}{\forall}\ u_p(t) = v_p\left(t - T_B^p\right). \tag{7.1}$$

Since signal $v_p(t)$ constitutes a vital part of the proposed control scheme, its proper definition will be given together with the description of the flow regulation strategy in the subsequent section.

The queue length at any instant of time depends on the data arrival speed and on the consumed bandwidth $h(\cdot)$. Suppose that initially the bottleneck link buffer is empty, i.e., $y(0) = 0$. Then, it follows from (7.1) that no packet arrives at the bottleneck node before $T_{\mathrm{F}\min} = \min_{p=1,2,\ldots,m} (T_F^p)$, and $y(t) = 0$ for any time instant t smaller than or equal to T_{Fmin}. Afterwards, for any $t > T_{\mathrm{Fmin}}$, the buffer queue length may be calculated from the following equation:

$$y(t) = \sum_{p=1}^{m} \int_{T_F^p}^{t} u_p\left(\tau - T_F^p\right) \mathrm{d}\tau - \int_{0}^{t} h(\tau)\mathrm{d}\tau. \tag{7.2}$$

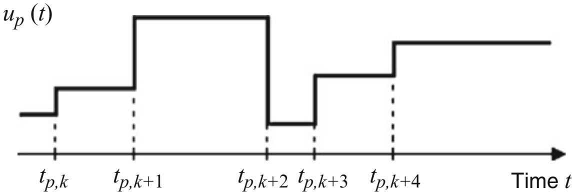


Fig. 7.2 Example source rate profile

Let us denote by $t_{p,k}$ the kth moment of time ($k = 1, 2, \ldots$) when a control unit carrying the feedback information for connection p arrives back at source p. Since the sources adjust the transmission rate only when control units return with the network feedback incorporated, then

$$\forall_{t \in [t_{p,k}; t_{p,k+1})} \; u_p(t) = u_p\left(t_{p,k}\right) = v_p\left(t_{p,k} - T_{\mathrm{B}}^{p}\right) = \text{const.} \tag{7.3}$$

It is assumed that the first piece to be transferred by any source is a control unit so that the information about the current network state could be received at the data origin as quickly as possible. As the sources begin transmission at the instant $t = 0$, then for $k = 1$, we have $t_{p,1} = \mathrm{RTT}_p$. Furthermore, since control units are sent every N data packets and not less frequently than the maximum control period T_{C} one after another, $t_{p,k+1}$ is specified by the relation

$$t_{p,k+1} = \min\left(t_{p,k} + \chi_{p,k}, t_{p,k} + T_{\mathrm{C}}\right), \tag{7.4}$$

where $\chi_{p,k}$ can be determined from the following equation:

$$\int\limits_{t_{p,k}}^{t_{p,k}+\chi_{p,k}} u_p\left(\tau - \mathrm{RTT}_p\right) d\tau = N. \tag{7.5}$$

Definitions (7.4) and (7.5) make sense only for nonnegative rates $u_p(t)$. Clearly, any control algorithm should be constructed in such a way that this condition is satisfied for every signal $u_p(t)$. Otherwise, a negative $u_p(t)$ would imply that the packets already injected into the network should be extracted by the sources before reaching the destination. Finally, it should be pointed out that Eqs. (7.4) and (7.5) present the main novelty of the network model used in this chapter. These two equations explicitly account for the time-varying, input-dependent sampling period in the considered system. An example source rate profile when the feedback information from the network is accessible at discrete, irregularly spaced time instants is illustrated in Fig. 7.2.

7.2 Principal Control Strategy

Modern telecommunication systems demand efficient resource usage. Therefore, any control strategy to be applied in these systems should guarantee that:

(i) Data is not lost due to congestion, as this allows one to minimize the network overhead corresponding to retransmissions.
(ii) There is always some data ready for transmission in the bottleneck link buffer so that the bandwidth at that link is entirely utilized.

Further in this section, we formulate a nonlinear flow control strategy, which ensures that these – partly contradictory – objectives are achieved even though the feedback information used for rate adjustment at the sources is received aperiodically. The proposed strategy combines the benefits of LQ optimal control obtained in the previous chapter for the system with constant sampling and the saturation element enhancing the robustness to parametric uncertainty.

7.2.1 Flow Control Strategy

The rate calculated by the controller, $u(t)$, is determined from the relation given below:

$$u(t) = \begin{cases} 0, & \text{if} \quad \omega(t) < 0, \\ \omega(t), & \text{if} \quad 0 \le \omega(t) \le u_{\max}, \\ u_{\max}, & \text{if} \quad \omega(t) > u_{\max}, \end{cases} \tag{7.6}$$

where $u_{\max} > 0$ denotes the upper saturation limit (usually set equal to the link capacity). Function $\omega(t)$ is defined as

$$\omega(t) = \Gamma \left[y_{\mathrm{D}} - y(t) - \Omega(t) \right], \tag{7.7}$$

where $\Gamma > 0$ is the controller gain and $y_{\mathrm{D}} > 0$ is the demand queue length. The component

$$\Omega(t) = \sum_{p=1}^{m} \int_{t-\mathrm{RTT}_p}^{t} v_p(\tau)\, \mathrm{d}\tau \tag{7.8}$$

represents the in-flight data (the number of packets allowed to be transmitted by the sources within the last RTT, but which yet not arrived at the bottleneck node due to delay). It corresponds to the summation term in (6.27). Rate $v_p(t)$, sent and recorded by the bottleneck node for source p at the instant of a control unit passing through the node, is determined from the following equation:

$$v_p(t) = \begin{cases} 0, & \text{for} \quad t < T_{\mathrm{F}}^p, \\ \lambda_p u\left(t_{p,k} - T_{\mathrm{B}}^p\right), & \text{for} \quad t \geq T_{\mathrm{F}}^p \quad \text{and} \quad t \in \left[t_{p,k} - T_{\mathrm{B}}^p, t_{p,k+1} - T_{\mathrm{B}}^p\right). \end{cases} \tag{7.9}$$

Therefore, initially, source p sends only control units (no data packets) until it receives the information about the first nonzero rate at $T_{\mathrm{F}}^p + \mathrm{RTT}_p$. Afterwards, it delivers packets at the rate recorded by the controller at discrete-time instants $t_{p,k} - T_{\mathrm{B}}^p$. This rate is determined according to (7.6)–(7.9) with the allocation strategy satisfying the condition $\sum_{p=1}^m \lambda_p = 1$. Since the individual source rate is limited by $\lambda_p u_{\max}$, the amount of the in-flight packets $\Omega(\cdot)$ at any moment of time t will be subject to the constraint

$$0 \leq \Omega(t) \leq u_{\max} \sum_{p=1}^{m} \lambda_p \mathrm{RTT}_p = u_{\max} \mathrm{RTT}_\mu, \tag{7.10}$$

where RTT_μ denotes the weighted mean RTT of the considered set of m connections determined according to the rate allocation strategy represented by the coefficients $\lambda_1, \lambda_{2,} \ldots, \lambda_m$.

7.2.2 Properties of the Proposed Strategy

In this section, appropriate conditions that are necessary to fulfill design objectives (i) and (ii) are formulated as two theorems. The first proposition indicates the amount of memory which needs to be reserved at the bottleneck node for data storage so that no packet is discarded irrespective of the available bandwidth changes. The second theorem shows how the demand queue length should be selected to guarantee full resource usage at the bottleneck link. Obviously, definitions (7.6) and (7.9) ensure that the rate of the sources is always nonnegative and bounded, and the fundamental applicability requirement of a data flow control algorithm is fulfilled.

Theorem 7.1. *If the sources in the considered network transmit data according to the conditions formulated in (7.1), (7.3), and (7.9), then for any $t \geq 0$, the queue length at the bottleneck node does not exceed the value given by the following inequality*:

$$y(t) \leq y_{\mathrm{D}} + u_{\max} T_{\mathrm{C}}. \tag{7.11}$$

Proof. First, notice that as a consequence of the source transfer speed adjustment at discrete moments of time, the total arrival rate at the bottleneck node may also change only at discrete time instants, further denoted by θ_i $(i = 1, 2, \ldots)$ with $\theta_1 = T_{\mathrm{Fmin}}$. The interval

$$\alpha_i = \theta_{i+1} - \theta_i, \tag{7.12}$$

between any two consecutive potential changes of the total incoming rate at the bottleneck node is subject to the constraint $0 \le \alpha_i \le T_{\mathrm{C}}$. The zero-length interval reflects the case when modification of the transmission speed which occurred at two or more sources influences the aggregate rate at the node at the same moment of time, and the upper bound is the maximum control unit interarrival period. Note that θ_i represents an instant of a possible change of the incoming rate. Since the sources adapt the rate at discrete time instants only, the incoming rate is constant between θ_i and θ_{i+1}.

The bottleneck link buffer is empty until $\theta_1 = T_{\mathrm{Fmin}}$. Therefore, $y(t \le \theta_1) = 0 < y_{\mathrm{D}} + u_{\max} T_{\mathrm{C}}$, and the proposition holds for any $t \le \theta_1$. Let us consider some $i > 1$ and the queue length at a time instant from the interval $[\theta_i, \theta_{i+1})$. Denoting this instant by $t = \theta_i + \eta$, where $\eta \in [0, \alpha_i)$ with α_i defined by (7.12), the queue length $y(t)$ can be expressed as the sum of three terms

$$y(t) = y(\theta_i) + \Phi(\theta_i, t) - \int_{\theta_i}^{t} h(\tau)\,\mathrm{d}\tau = y(\theta_i) + \Phi(\theta_i, \theta_i + \eta) - \int_{\theta_i}^{\theta_i+\eta} h(\tau)\,\mathrm{d}\tau. \tag{7.13}$$

The first element in (7.13) represents the queue length at instant θ_i. Function

$$\Phi(t_1, t_2) = \sum_{p=1}^{m} \int_{t_1}^{t_2} u_p\left(\tau - T_{\mathrm{F}}^{p}\right)\mathrm{d}\tau \tag{7.14}$$

reflects the amount of data which arrived at the bottleneck node between time instants t_1 and t_2, $t_2 \ge t_1$. Using (7.3) and (7.9), we can rewrite (7.14) as

$$\Phi(t_1, t_2) = \sum_{p=1}^{m} \int_{t_1}^{t_2} v_p\left(\tau - \mathrm{RTT}_p\right)\mathrm{d}\tau = \sum_{p=1}^{m} \int_{t_1-\mathrm{RTT}_p}^{t_2-\mathrm{RTT}_p} v_p(\tau)\,\mathrm{d}\tau \le u_{\max}(t_2 - t_1). \tag{7.15}$$

Finally, the last term in (7.13) represents the amount of data actually transferred at the bottleneck link between the instants θ_i and $\theta_i + \eta$.

In order to analyze the queue length variations in the interval $[\theta_i, \theta_{i+1})$, we need to examine the behavior of function $\omega(\cdot)$. We will consider two cases: first, the situation when $\omega(\theta_i) \ge 0$, and, afterwards, the circumstances when $\omega(\theta_i) < 0$.

Case 1. We analyze the situation when $\omega(\theta_i) \geq 0$. From the definition of function $\omega(\cdot)$ (7.7), we get

$$\omega\left(\theta_i\right) = \Gamma\left[y_{\mathrm{D}} - y\left(\theta_i\right) - \Omega\left(\theta_i\right)\right] \geq 0, \tag{7.16}$$

which after the term rearrangement reduces to

$$y\left(\theta_i\right) \leq y_{\mathrm{D}} - \Omega\left(\theta_i\right). \tag{7.17}$$

Applying (7.17) to (7.13), we obtain the following estimate of the queue length at instant $t = \theta_i + \eta$:

$$y\left(\theta_i + \eta\right) \leq y_{\mathrm{D}} - \Omega\left(\theta_i\right) + \Phi\left(\theta_i, \theta_i + \eta\right) - \int\limits_{\theta_i}^{\theta_i + \eta} h\left(\tau\right) \mathrm{d}\tau. \tag{7.18}$$

Since the individual source rate is upper-bounded by $\lambda_p u_{\max}$ and lower-bounded by zero, the difference $\Phi(\theta_i, \theta_i + \eta) - \Omega(\theta_i)$ can be evaluated as

$$\Phi\left(\theta_i, \theta_i + \eta\right) - \Omega\left(\theta_i\right) \leq u_{\max}\left(\theta_i + \eta - \theta_i\right) - 0 = u_{\max}\eta. \tag{7.19}$$

Substituting (7.19) into (7.18), we get

$$y\left(\theta_i + \eta\right) \leq y_{\mathrm{D}} + u_{\max}\eta - \int\limits_{\theta_i}^{\theta_i + \eta} h\left(\tau\right) \mathrm{d}\tau. \tag{7.20}$$

The utilized bandwidth, $h(\cdot)$, is always nonnegative; hence,

$$y\left(\theta_i + \eta\right) \leq y_{\mathrm{D}} + u_{\max}\eta. \tag{7.21}$$

Furthermore, since η is upper-bounded by T_{C}, we get

$$y\left(\theta_i + \eta\right) \leq y_{\mathrm{D}} + u_{\max}T_{\mathrm{C}}, \tag{7.22}$$

which ends the first part of the proof.

Case 2. Now, let us examine the situation when $\omega(\theta_i) < 0$. First, we find the last moment $t^* < \theta_i$ when signal $\omega(\cdot)$ was greater than or equal to zero. It should be stressed at this point that since control unit emission at the sources and rate generation at the bottleneck node are not synchronized, t^* does not have to coincide with any of time instants θ_i. According to (7.7) and (7.9),

$$\omega\left(t \leq T_{\mathrm{F\,min}}\right) = \Gamma\left(y_{\mathrm{D}} - 0 - 0\right) = \Gamma y_{\mathrm{D}} > 0. \tag{7.23}$$

In consequence, the first moment, when signal $\omega(\cdot)$ may attain a value smaller than zero, is greater than T_{Fmin}, and instant t^* actually exists. The value of $\omega(t^*)$ satisfies the following inequality:

$$\omega\left(t^*\right) = \Gamma\left[y_{\text{D}} - y\left(t^*\right) - \Omega\left(t^*\right)\right] > 0, \tag{7.24}$$

which after the term rearrangement leads to

$$y\left(t^*\right) < y_{\text{D}} - \Omega\left(t^*\right). \tag{7.25}$$

The queue length at a time instant $t \in [\theta_i, \theta_{i+1})$ can be expressed as

$$y(t) = y\left(t^*\right) + \Phi\left(t^*, t\right) - \int\limits_{t^*}^{t} h\left(\tau\right) \mathrm{d}\tau. \tag{7.26}$$

Substituting (7.25) for $y(t^*)$, we get

$$y(t) < y_{\text{D}} - \Omega\left(t^*\right) + \Phi\left(t^*, t\right) - \int\limits_{t^*}^{t} h\left(\tau\right) \mathrm{d}\tau. \tag{7.27}$$

The term $\Phi(t^*, t) - \Omega(t^*)$ describes the difference between the number of packets which arrived at the bottleneck node from t^* to t and the number of packets still "in-flight" at t^*, i.e., those for which the sending rate has already been calculated and which have not yet arrived at the bottleneck node due to the delay. From (7.10) and (7.15), we obtain

$$\begin{aligned}\Phi\left(t^*, t\right) - \Omega\left(t^*\right) &= \sum_{p=1}^{m} \int\limits_{t^*-\text{RTT}_p}^{t-\text{RTT}_p} v_p\left(\tau\right) \mathrm{d}\tau - \sum_{p=1}^{m} \int\limits_{t^*-\text{RTT}_p}^{t^*} v_p\left(\tau\right) \mathrm{d}\tau \\ &= \sum_{p=1}^{m} \int\limits_{t^*}^{t-\text{RTT}_p} v_p\left(\tau\right) \mathrm{d}\tau = \sum_{p=1}^{m} \int\limits_{t^*}^{t} v_p\left(\tau\right) \mathrm{d}\tau - \sum_{p=1}^{m} \int\limits_{t-\text{RTT}_p}^{t} v_p\left(\tau\right) \mathrm{d}\tau.\end{aligned} \tag{7.28}$$

In consequence, the analyzed difference may be represented as

$$\Phi\left(t^*, t\right) - \Omega\left(t^*\right) = \sum_{p=1}^{m} \int\limits_{t^*}^{t} v_p\left(\tau\right) \mathrm{d}\tau - \Omega(t). \tag{7.29}$$

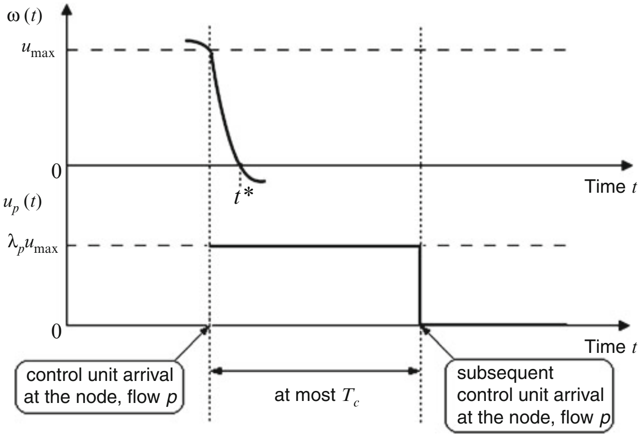


Fig. 7.3 Transmission rate assignment in the vicinity of *t** – rate decrease

Since the transfer speed generated by the controller in the interval (*t**, *t*] is equal to zero, and the rate assignment can be delayed by T_{C} (see Fig. 7.3), (7.29) can be estimated as

$$\Phi\left(t^*, t\right) - \Omega\left(t^*\right) \leq u_{\max} T_{\mathrm{C}} - 0 = u_{\max} T_{\mathrm{C}}. \tag{7.30}$$

Function $h(\cdot)$ is always nonnegative, so we can state that the queue length at instant *t* given by (7.27) will be limited by the following value:

$$\begin{aligned} y(t) &< y_{\mathrm{D}} + \Phi\left(t^*, t\right) - \Omega\left(t^*\right) - \int_{t^*}^{t} h(\tau)\, \mathrm{d}\tau \\ &\leq y_{\mathrm{D}} + \Phi\left(t^*, t\right) - \Omega\left(t^*\right) \leq y_{\mathrm{D}} + u_{\max} T_{\mathrm{C}}. \end{aligned} \tag{7.31}$$

This concludes the proof. □

Full link utilization, as formulated by (ii), requires the presence of sufficient number of packets in the bottleneck node buffer at any instant of time. If the queue length is greater than zero, then the total available bandwidth of the bottleneck link is consumed. The theorem presented below shows how the demand queue length should be selected to meet this objective and to ensure that the entire available bandwidth at the output connection is used for data traffic.

Theorem 7.2. *If the sources in the considered network transmit data according to the conditions formulated in (7.1), (7.3), and (7.9), the maximum rate $u_{max} > d_{max}$, and the demand queue length satisfies the following inequality*:

$$y_{\mathrm{D}} > u_{\max}\left(\mathrm{RTT}_{\mu} + \Gamma^{-1} + T_{\mathrm{C}}\right), \tag{7.32}$$

then for any $t > T_{Fmax} + T_C + T_{max}$, *where* $T_{\mathrm{F\,max}} = \max_{p=1,2,\ldots,m}\left(T_{\mathrm{F}}^{p}\right)$ *and*

$$T_{\max} = (y_{\mathrm{D}} + u_{\max} T_{\mathrm{C}}) \,/\, (u_{\max} - d_{\max}), \tag{7.33}$$

the queue length is always greater than zero.

Proof. The theorem assumption implies that we deal with time instants $t > T_{\mathrm{Fmax}} + T_{\mathrm{C}} + T_{\max} > \theta_1$. Considering some $i > 1$ and the value of signal $\omega(\cdot)$ at the moment of input rate modification at the bottleneck node θ_i, we may distinguish two cases: the situation when $\omega(\theta_i) < u_{\max}$ and the circumstances when $\omega(\theta_i) \geq u_{\max}$.

Case 1. First, we consider the situation when $\omega(\theta_i) < u_{\max}$. Directly from the definition of function $\omega(\cdot)$, we obtain

$$\omega\left(\theta_i\right) = \Gamma\left[y_{\mathrm{D}} - y\left(\theta_i\right) - \Omega\left(\theta_i\right)\right] < u_{\max}, \tag{7.34}$$

or

$$y\left(\theta_i\right) > y_{\mathrm{D}} - u_{\max}/\Gamma - \Omega\left(\theta_i\right). \tag{7.35}$$

According to (7.10), $\Omega(\theta_i) \leq u_{\max}\mathrm{RTT}_{\mu}$, which implies

$$y\left(\theta_i\right) > y_{\mathrm{D}} - u_{\max}/\Gamma - u_{\max}\mathrm{RTT}_{\mu}. \tag{7.36}$$

Using assumption (7.32), we obtain

$$y\left(\theta_i\right) > u_{\max}\left(\mathrm{RTT}_{\mu} + \Gamma^{-1} + T_{\mathrm{C}}\right) - u_{\max}/\Gamma - \Omega_{\max} = u_{\max}T_{\mathrm{C}} > 0. \tag{7.37}$$

Let us examine the queue length at some time instant $t = \theta_i + \eta$, as defined by (7.13). The minimum rate which can be assigned to each source is zero; hence, $\Phi(\theta_i, \theta_i + \eta) \geq 0$. On the other hand, the maximum available bandwidth equals $d_{\max}$, which implies $\int_{\theta_i}^{\theta_i+\eta} h\left(\tau\right)\mathrm{d}\tau \leq d_{\max}\eta$. Consequently, we get from (7.13)

$$y\left(\theta_i + \eta\right) \geq y\left(\theta_i\right) + 0 - d_{\max}\eta. \tag{7.38}$$

Applying (7.37), and (7.38), we get

$$y(t) > u_{\max}T_{\mathrm{C}} - d_{\max}\eta. \tag{7.39}$$

Since $u_{\max} > d_{\max}$ and $T_{\mathrm{C}} > \eta$, we conclude that $y(t) > 0$, which completes the first part of the proof.

Case 2. Now, let us study the situation when $\omega(\theta_i) \geq u_{\max}$. First, we find the last moment $t* < \theta_i$ when signal $\omega(\cdot)$ was smaller than $u_{\max}$. It comes from Theorem

7.1 that the queue length never exceeds the value of $y_{\mathrm{D}} + u_{\max} T_{\mathrm{C}}$. Furthermore, the packet depletion rate is limited by $d_{\max}$. Thus, the maximum period of time $T_{\max}$ during which the controller may continuously set rate $u_{\max}$ for the sources is determined as $T_{\max} = (y_{\mathrm{D}} + u_{\max} T_{\mathrm{C}})/(u_{\max} - d_{\max})$. Therefore, instant t^* indeed exists. Since t^* is the last instant, when signal $\omega(\cdot)$ was smaller than $u_{\max}$ and the actual rate assignment could be delayed by not more than T_{C},

$$t^* \geq \theta_i - (T_{\max} + T_{\mathrm{C}})\,. \tag{7.40}$$

From the theorem assumption, it also comes that $\theta_i > T_{\mathrm{Fmax}} + T_{\max} + T_{\mathrm{C}}$, which implies $t^* > T_{\mathrm{Fmax}}$. This means that by the time instant t^*, the control units from all the sources have already reached the bottleneck node, and all m flows are subject to the control action (and rate updates at least every T_{C}).

The value of $\omega(t^*)$ satisfies the inequality given below:

$$\omega\left(t^*\right) = \Gamma\left[y_{\mathrm{D}} - y\left(t^*\right) - \Omega\left(t^*\right)\right] < u_{\max}. \tag{7.41}$$

Following a similar reasoning as presented in (7.34)–(7.37), we arrive at

$$y\left(t^*\right) > y_{\mathrm{D}} - u_{\max}/\,\Gamma - \Omega\left(t^*\right) > u_{\max} T_{\mathrm{C}} > 0. \tag{7.42}$$

The queue length at any time instant $t \in [\theta_i, \theta_{i+1})$ may be expressed as in (7.26). Applying (7.42) to (7.26), we get

$$y(t) > u_{\max} T_{\mathrm{C}} + \Phi\left(t^*, t\right) - \int\limits_{t^*}^{t} h\left(\tau\right) \mathrm{d}\tau. \tag{7.43}$$

Recall that t^* was the last instant before t when the controller calculated rate smaller than $u_{\max}$. This rate could be as low as zero. Afterwards, the algorithm generates the maximum rate. Since control units appear at discrete time instants, the transfer speed assignment can be delayed but not more than by T_{C} (see Fig. 7.4). Thus, the amount of the incoming data $\Phi(\cdot)$ in the interval $(t^*, t]$ will satisfy the following inequality:

$$\Phi\left(t^*, t\right) \geq 0 \cdot T_{\mathrm{C}} + u_{\max}\left(t - t^* - T_{\mathrm{C}}\right) = u_{\max}\left(t - t^* - T_{\mathrm{C}}\right) \tag{7.44}$$

and the queue length given by (7.43) can be estimated as

$$y(t) > u_{\max} T_{\mathrm{C}} + u_{\max}\left(t - t^* - T_{\mathrm{C}}\right) - \int\limits_{t^*}^{t} h\left(\tau\right) \mathrm{d}\tau = u_{\max}\left(t - t^*\right) - \int\limits_{t^*}^{t} h\left(\tau\right) \mathrm{d}\tau. \tag{7.45}$$

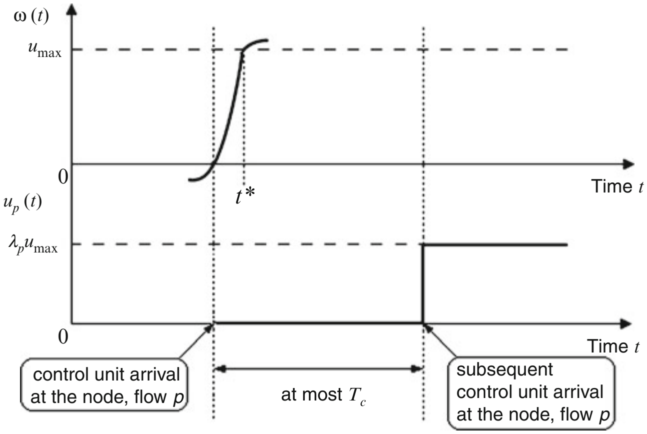

ω (t)
u_{max}
0
t*
Time t
u_p (t)
$\lambda_p u_{max}$
0
Time t
control unit arrival at the node, flow p
at most T_c
subsequent control unit arrival at the node, flow p

Fig. 7.4 Transmission rate assignment in the vicinity of t^* – rate increase

For any t, the utilized bandwidth $h(t) \le d_{\max}$, and

$$y(t) > u_{\max}\left(t - t^*\right) - d_{\max}\left(t - t^*\right). \tag{7.46}$$

According to the assumption $u_{\max} > d_{\max}$ and the fact that $t > t^*$, we may state that

$$y(t) > (u_{\max} - d_{\max})\left(t - t^*\right) > 0. \tag{7.47}$$

This ends the proof of Theorem 7.2. □

Remark 7.1. The demand queue length necessary to ensure full bandwidth utilization (7.32) and the buffer size required to eliminate losses (7.11) are specified following the worst-case disturbance scenario. If the node is faced by a stochastic bandwidth with mean d_μ much lower than $d_{\max}$, then instead of the maximum values of the transmission rate $u_{\max}$ and the control unit interarrival period T_C smaller quantities may be used in (7.11) and (7.32). For instance, one may use d_μ in place of $d_{\max}$, and mN/d_μ instead of T_C (if $T_C > mN/d_\mu$). Then, the maximum throughput will not be guaranteed at all times, yet substantial buffer capacity savings can be obtained while maintaining the bandwidth utilization close to maximum.

7.2.3 Simulation Results

First, the model of wide area network with irregular period of feedback information availability is constructed according to the description provided in Sect. 7.1. Similarly as in the previous chapter, four connections ($m = 4$) are selected to participate in the flow regulation process. They are characterized by the same

Table 7.1 Connection RTTs

Connection	Delay [ms]		
p	T_F^p	T_F^p	RTT_p
1	10	20	30
2	25	45	70
3	30	50	80
4	50	70	120

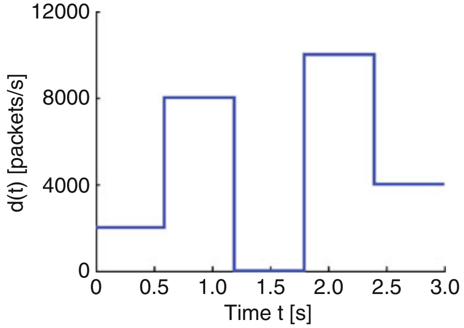


Fig. 7.5 Available bandwidth

RTT*s* repeated for convenience in Table 7.1. The overall transfer rate determined by the controller is distributed evenly among the flows. Hence, the weights $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1/m = 1/4$.

Parameters of the feedback information exchange are adjusted according to the guidelines of the ATM/ABR standard – the chief example of the network with input-dependent sampling. Consequently, in the considered model, each source sends a control unit every $N = 32$ equal-size data pieces but not less frequently than every $T_C = 100$ ms. The maximum available bandwidth $d_{\max}$ is set as 10,000 packets/s, and the upper bound of the overall source rate $u_{\max}$ is adjusted to 11,000 packets/s $= 1.1d_{\max}$. The bandwidth actually available for the controlled connections at the bottleneck node $d(\cdot)$ follows the pattern shown in Fig. 7.5.

It follows from the derivation presented in Chap. 6 that the optimal gain for the purely discrete-time system with constant discretization period T=10 ms and w=1 equals 0.618 (the golden-ratio controller). For the sampled data system with variable discretization period, instead of T, we use the theoretical average sampling rate,

$$T_{\mathrm{avg}} = \left(T_C + mN/u_{\max}\right)/2 = (0.100 + 4 \cdot 32/11,000)/2 = 0.056\mathrm{s}, \qquad (7.48)$$

and set the gain $\Gamma = 0.618/T_{\mathrm{avg}} = 11.036\ \mathrm{s}^{-1}$. Consequently, in order to fulfill the requirements imposed by Theorem 7.2, the demand queue length y_D is chosen as 2,930 packets $> u_{\max}(\mathrm{RTT}_\mu + \Gamma^{-1} + T_C) = 11{,}000\cdot(0.075+1/11.036+0.1)$=2,922 packets.

Function $\omega(\cdot)$ and the overall rate calculated by the controller are illustrated in Figs. 7.6 and 7.7, respectively. Although signal $\omega(\cdot)$ drops below zero in the interval [1.52 s, 1.80 s], the rate established by the controller, presented in Fig. 7.7, is always nonnegative and limited, as required for the applicability purposes in communication systems.

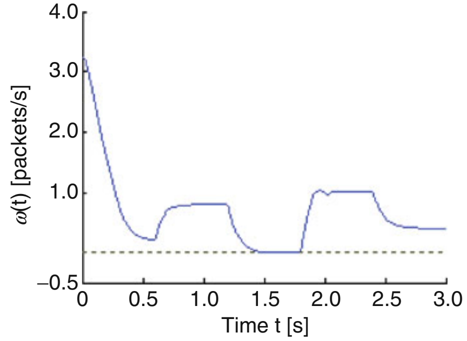


Fig. 7.6 Signal ω(t)

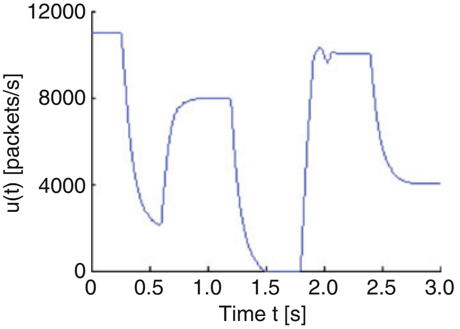


Fig. 7.7 Transmission rate generated by the controller

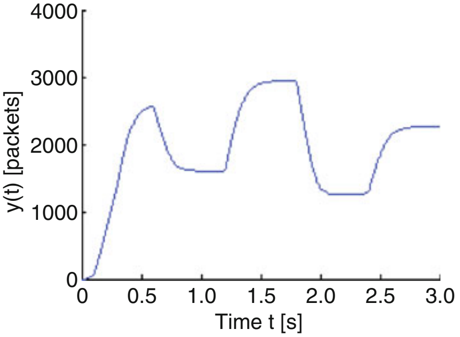


Fig. 7.8 Buffer occupancy

The queue length resulting from the operation of the designed controller is shown in Fig. 7.8. As we can see, the value of $y(\cdot)$ never exceeds the level of $y_D + u_{\max} T_C = 4{,}030$ packets and does not drop to zero. These two properties imply no buffer overflow and full bottleneck link utilization.

In Fig. 7.9, we show transmission rate of the sources. We can see from the graphs that each transmitter p, $p \in \{1, 2, 3, 4\}$, starts sending packets at RTT_p. Afterwards, if the rate is high (e.g., in the interval [0.8 s, 1.3 s] or [2.0 s, 2.4 s]), control units are emitted more frequently and the rate is adjusted at shorter time intervals. If, on the

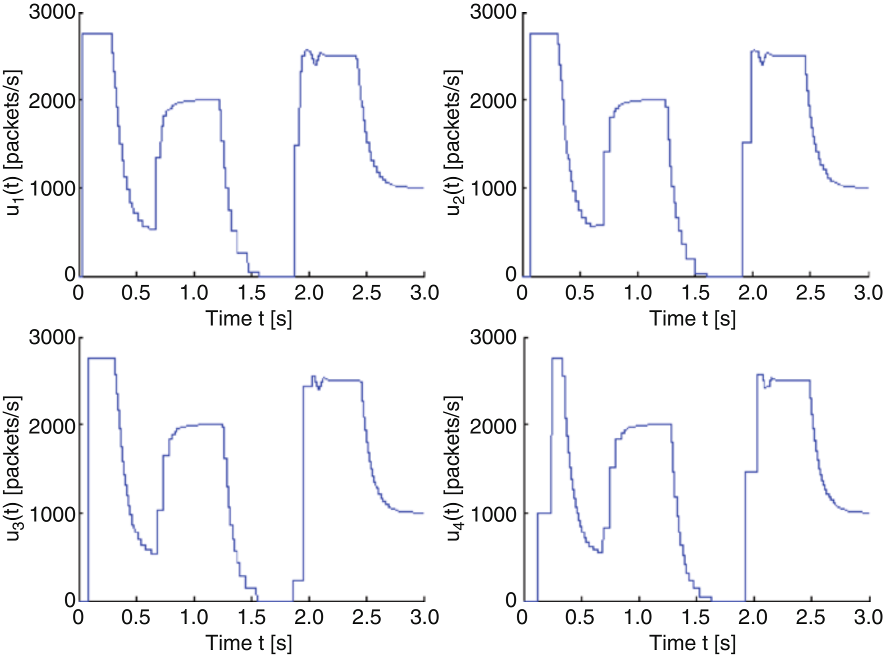


Fig. 7.9 Transmission rate of the sources

other hand, the rate diminishes, the resulting sampling period grows, and the transfer speed is adapted at longer time intervals. This can be observed between 0.5 and 0.7 s, and between 1.4 and 1.9 s. It is also clear from the plots in Fig. 7.9 that the control unit emission and rate adjustment at various sources are not synchronized with each other. Each transmitter has its own packet counter (and T_C timer) and sends control units every $N = 32$ data packets (and at least every $T_C = 100$ ms) according to (7.4) and (7.5). Hence, the control unit emission in one flow is only indirectly related to the phenomena occurring in other flows sharing the resources of the bottleneck node. Despite significant variations in the sampling rate, the favorable properties of the applied control scheme (no losses and full bandwidth utilization) are achieved, as demonstrated in the theorems.

7.3 Robustness Issues

The properties of the controller presented in the previous section were demonstrated under the assumption that RTTs of the established connections are known exactly at the bottleneck node. However, in real networks, the measurement of delay in the connection setup phase is prone to errors, and parameters of the network model used by the controller cannot always be determined with good accuracy. The mismatch between the real and estimated delay may also occur if two or more control units belonging to different flows arrive at the node at the same moment of time. Then, only one of them can be forwarded immediately, and RTT of other flows increases. Therefore, in this section, we study how possible differences between the true delays in the network and those used by the controller for rate calculation, further denoted by $\overline{\mathrm{RTT}}_p$, may influence the flow regulation process.

7.3.1 Flow Control Strategy

Equations (7.6) and (7.9), formulated for the principal control strategy, remain valid. However, in order to take into account the discrepancies in delay parameter values, the definition of function $\omega(\cdot)$ stated in (7.7) needs to be updated in the following way:

$$\omega(t) = \bar{\omega}(t) = \Gamma \left[y_{\mathrm{D}} - y(t) - \overline{\Omega}(t) \right], \tag{7.49}$$

where, similarly as in (7.7), $\Gamma > 0$ is the controller gain and $y_{\mathrm{D}} > 0$ is the demand queue length. The component representing the in-flight data,

$$\overline{\Omega}(t) = \sum_{p=1}^{m} \int\limits_{t-\overline{\mathrm{RTT}}_p}^{t} v_p\,(\tau)\,\mathrm{d}\tau, \tag{7.50}$$

where $\overline{\mathrm{RTT}}_p > 0$ represents the round-trip time of flow p measured by the controller in the connection setup phase, is subject to the constraint

$$0 \leq \overline{\Omega}(t) \leq u_{\max}\overline{\mathrm{RTT}}_{\mu}. \tag{7.51}$$

Symbol $\overline{\mathrm{RTT}}_{\mu}$ in (7.51) denotes the mean estimated RTT of the considered set of m connections. When the overall flow rate generated by the controller is distributed among the flows according to the policy determined by the coefficients $\lambda_{1,}\lambda_{2,}\ldots,$ λ_m, then $\overline{\mathrm{RTT}}_{\mu} = \sum_{p=1}^{m} \lambda_p \overline{\mathrm{RTT}}_p$.

7.3.2 Properties of the Proposed Strategy

Theorems 7.1 and 7.2, which were formulated for the strategy described in Sect. 7.2, no longer hold. Still though, we can modify the requirements for buffer space allocation so that the desirable algorithm properties, specified by (i) and (ii), will be preserved.

Theorem 7.3. *If the sources in the considered network transmit data according to the conditions formulated in (7.1), (7.3), and (7.9), with signal $\omega(\cdot) \triangleq \bar{\omega}(\cdot)$ defined by (7.49), then the queue length at the bottleneck node is always upper-bounded, i.e., for any $t \geq 0$,*

$$y(t) \leq y_{\mathrm{D}} + u_{\max} T_C + \Delta_{\max}, \tag{7.52}$$

where

$$\Delta_{\max} = u_{\max} \sum_{p:\mathrm{RTT}_p > \overline{\mathrm{RTT}}_p} \lambda_p \left(\mathrm{RTT}_p - \overline{\mathrm{RTT}}_p\right). \tag{7.53}$$

Proof. No packet arrives at the bottleneck node before $t = \theta_1 = T_{\mathrm{Fmin}}$. Hence, taking into account the initial conditions, $y(t \leq \theta_1) = 0 < y_{\mathrm{D}} + u_{\max} T_{\mathrm{C}} + \Delta_{\max}$, and the proposition is satisfied for any $t \leq \theta_1$. Let us consider the value of signal $\bar{\omega}(\cdot)$ at instant θ_i for some $i > 1$. Two situations need to be analyzed: the first occurs when $\bar{\omega}(\theta_i) \geq 0$ and the other when $\bar{\omega}(\theta_i) < 0$.

Case 1. Investigating the case

$$\bar{\omega}(\theta_i) = \Gamma\left[y_{\mathrm{D}} - y(\theta_i) - \overline{\Omega}(\theta_i)\right] \geq 0 \tag{7.54}$$

leads to the inequality

$$y(\theta_i) \leq y_{\mathrm{D}} - \overline{\Omega}(\theta_i) \leq y_{\mathrm{D}}. \tag{7.55}$$

Applying (7.55) to (7.13), we get the following estimate of the queue length in the bottleneck node buffer at some time instant $t \in [\theta_i, \theta_{i+1})$:

$$y(t = \theta_i + \eta) \leq y_{\mathrm{D}} + \Phi(\theta_i, \theta_i + \eta) - \int_{\theta_i}^{\theta_i + \eta} h(\tau)\,\mathrm{d}\tau, \tag{7.56}$$

where $\eta \in [0, \alpha_i)$ with α_i defined by (7.12). According to (7.15), $\Phi(\theta_i, \theta_i + \eta) \leq u_{\max}\eta$. On the other hand, $h(\cdot) \geq 0$, which implies

$$y(t) \leq y_{\mathrm{D}} + u_{\max}\eta - 0 \leq y_{\mathrm{D}} + u_{\max} T_{\mathrm{C}} + \Delta_{\max}. \tag{7.57}$$

This ends the first part of the proof.

Case 2. Now, let us examine the circumstances, when $\bar{\omega}(\theta_i) < 0$. First, we seek the last instant $t^* < \theta_i$ when signal $\bar{\omega}(\cdot)$ was nonnegative. Since $\bar{\omega}(t \leq T_{\mathrm{Fmin}}) = \Gamma y_{\mathrm{D}} > 0$, the first moment, when $\bar{\omega}(\cdot)$ may attain a value smaller than zero, is greater than T_{Fmin}. The value of $\bar{\omega}(t^*)$ satisfies the following inequality:

$$\bar{\omega}\left(t^*\right) = \Gamma\left[y_{\mathrm{D}} - y\left(t^*\right) - \overline{\Omega}\left(t^*\right)\right] \geq 0. \tag{7.58}$$

Thus, after the term rearrangement, we obtain $y(t^*) \leq y_D - \overline{\Omega}(t^*)$. Using this inequality in formula (7.26) describing the queue length at any time instant $t \in [\theta_i, \theta_{i+1})$, we get

$$y(t) \leq y_D - \overline{\Omega}\left(t^*\right) + \Phi\left(t^*, t\right) - \int_{t^*}^{t} h\left(\tau\right) \mathrm{d}\tau. \tag{7.59}$$

Let us define a function $\Delta(t)$

$$\Delta(t) = \sum_{p=1}^{m} \int_{t-\mathrm{RTT}_p}^{t-\overline{\mathrm{RTT}}_p} v_p\left(\tau\right) \mathrm{d}\tau. \tag{7.60}$$

It is bounded by the following inequalities:

$$\Delta(t) \geq u_{\max} \sum_{p:\mathrm{RTT}_p<\overline{\mathrm{RTT}}_p} \lambda_p\left(\mathrm{RTT}_p - \overline{\mathrm{RTT}}_p\right) = -\Delta_{\min}, \tag{7.61}$$

$$\Delta(t) \leq u_{\max} \sum_{p:\mathrm{RTT}_p>\overline{\mathrm{RTT}}_p} \lambda_p\left(\mathrm{RTT}_p - \overline{\mathrm{RTT}}_p\right) = \Delta_{\max}, \tag{7.62}$$

where $\Delta_{\min}$ and $\Delta_{\max}$ are nonnegative real numbers. With this notation, we get

$$\begin{aligned}\overline{\Omega}(t) =& \sum_{p=1}^{m} \int_{t-\overline{\mathrm{RTT}}_p}^{t} v_p\left(\tau\right) \mathrm{d}\tau = \sum_{p=1}^{m} \int_{t-\mathrm{RTT}_p}^{t} v_p\left(\tau\right) \mathrm{d}\tau - \sum_{p=1}^{m} \int_{t-\mathrm{RTT}_p}^{t-\overline{\mathrm{RTT}}_p} v_p\left(\tau\right) \mathrm{d}\tau \\ =&\Omega(t) - \Delta(t).\end{aligned} \tag{7.63}$$

Substituting (7.63) into (7.59), we obtain the following estimate of the packet queue length at instant t:

$$y(t) \leq y_D - \Omega\left(t^*\right) + \Delta\left(t^*\right) + \Phi\left(t^*, t\right) - \int_{t^*}^{t} h\left(\tau\right) \mathrm{d}\tau. \tag{7.64}$$

Following a similar reasoning as presented in (7.27)–(7.31), we get $y(t) \leq y_D + u_{\max} T_C + \Delta(t^*)$, and, using (7.62), $y(t) \leq y_D + u_{\max} T_C + \Delta_{\max}$. This ends the proof. □

The next theorem shows that if the bottleneck node buffer is selected properly, then all of the bandwidth available at the bottleneck link will be consumed even though the delays are determined by the controller with limited accuracy.

Theorem 7.4. *If the sources in the considered network transmit data according to the conditions formulated in (7.1), (7.3), and (7.9), signal $\omega(\cdot) \triangleq \bar{\omega}(\cdot)$ is defined by (7.49), the maximum rate $u_{max} > d_{max}$, and the demand value of the queue length satisfies the following inequality*:

$$y_{\mathrm{D}} > u_{\max}\left(\mathrm{RTT}_{\mu} + \Gamma^{-1} + T_{\mathrm{C}}\right) + \Delta_{\min}, \tag{7.65}$$

where Δ_{min} is given by (7.61), then for any $t > T_{\mathrm{F}max} + T_{\mathrm{C}} + T_{max}$, where $T_{max} = (y_{\mathrm{D}} + u_{max}T_{\mathrm{C}} + \Delta_{max})/(u_{max} - d_{max})$, the queue length is strictly positive.

Proof. The theorem assumption implies that we deal with time instants $t > T_{\mathrm{Fmax}} + T_{\mathrm{C}} + T_{\max} > \theta_1$. Let us consider some $i > 1$ and the value of $\bar{\omega}(\cdot)$ at the corresponding time instant θ_i. Similarly as in the proof of Theorem 7.2, we analyze two cases: the first is when $\bar{\omega}(\theta_i) < u_{\max}$ and the second one when $\bar{\omega}(\theta_i) \geq u_{\max}$.

Case 1. We begin with the situation when $\bar{\omega}(\theta_i) < u_{\max}$. Then,

$$\bar{\omega}\left(\theta_i\right) = \Gamma\left[y_{\mathrm{D}} - y\left(\theta_i\right) - \overline{\Omega}\left(\theta_i\right)\right] < u_{\max}, \tag{7.66}$$

and according to (7.63), $\overline{\Omega}(t) = \Omega(t) - \Delta(t)$. Therefore, after the term rearrangement in (7.66), we get

$$y\left(\theta_i\right) > y_{\mathrm{D}} - u_{\max}/\Gamma - \bar{\Omega}\left(\theta_i\right) = y_{\mathrm{D}} - u_{\max}/\Gamma - \Omega\left(\theta_i\right) + \Delta\left(\theta_i\right). \tag{7.67}$$

Since for any time instant t, $\Omega(t) \leq u_{\max}\mathrm{RTT}_{\mu}$ and $\Delta(t) \geq -\Delta_{\min}$ (definition (7.61)), then

$$y\left(\theta_i\right) > y_{\mathrm{D}} - u_{\max}/\Gamma - u_{\max}\mathrm{RTT}_{\mu} - \Delta_{\min}. \tag{7.68}$$

From the theorem assumption (7.65), we get $y(\theta_i) > u_{\max}T_{\mathrm{C}}$. Applying this inequality to formula (7.13) describing the queue length at the time instant $t = \theta_i + \eta \in [\theta_i, \theta_{i+1})$, we obtain

$$y\left(\theta_i + \eta\right) > u_{\max}T_{\mathrm{C}} + \Phi\left(\theta_i, \theta_i + \eta\right) - \int_{\theta_i}^{\theta_i+\eta} h\left(\tau\right)\mathrm{d}\tau. \tag{7.69}$$

The minimum incoming flow rate $\Phi(\cdot)$ equals zero. The utilized bandwidth, in turn, can be as large as $d_{\max} < u_{\max}$. Since $\eta < T_{\mathrm{C}}$,

$$y\left(\theta_i + \eta\right) > u_{\max}T_{\mathrm{C}} - d_{\max}\eta > 0, \tag{7.70}$$

which completes the first part of the proof.

Case 2. In the second part of the proof, we investigate the circumstances when $\bar{\omega}(\theta_i) \geq u_{\max}$. First, we find the last moment $t^* < \theta_i$ when signal $\bar{\omega}(\cdot)$ was smaller than $u_{\max}$. Theorem 7.3 implies that the queue length never exceeds the value of $y_{\mathrm{D}} + u_{\max}T_{\mathrm{C}} + \Delta_{\max}$ despite possible lack of precision in RTTs estimation performed by the controller. On the other hand, the packet depletion rate is limited by $d_{\max}$. Thus, the maximum period of time $T_{\max}$, during which the controller may continuously set rate $u_{\max}$ for the sources, is estimated as $T_{\max} = (y_{\mathrm{D}} + u_{\max}T_{\mathrm{C}} + \Delta_{\max})/(u_{\max} - d_{\max})$, and instant t^* indeed exists. Since t^* is the last instant, when signal $\bar{\omega}(\cdot)$ was smaller than $u_{\max}$, and the actual rate assignment could be delayed by not more than T_{C}, then $t^* \geq \theta_i - (T_{\max} + T_{\mathrm{C}})$. From the theorem assumption, it also comes that $t^* > T_{\mathrm{Fmax}} + T_{\max} + T_{\mathrm{C}} - (T_{\max} + T_{\mathrm{C}}) = T_{\mathrm{Fmax}}$, which means that by t^* all the sources delivered at least one control unit to the node, and are subject to the control action with sampling at least every T_{C}. The value of $\bar{\omega}(t^*)$ satisfies the inequality given below:

$$\bar{\omega}\left(t^*\right) = \Gamma\left[y_{\mathrm{D}} - y\left(t^*\right) - \overline{\Omega}\left(t^*\right)\right] < u_{\max}. \tag{7.71}$$

Using (7.63), we can rewrite (7.71) as

$$y\left(t^*\right) > y_{\mathrm{D}} - u_{\max}/\,\Gamma - \Omega\left(t^*\right) + \Delta\left(t^*\right). \tag{7.72}$$

Following a similar reasoning as presented in (7.67) and (7.68), we arrive at $y(t^*) > u_{\max}T_{\mathrm{C}}$, which is applied to formula (7.26) describing the queue length at some time instant $t \in [\theta_i,\ \theta_{i+1})$. Taking analogous steps to those already considered in the proof of Theorem 7.2, namely, (7.43)–(7.47), we can estimate $y(t)$ as $y(t) > u_{\max}(t - t^*) - d_{\max}(t - t^*)$. The assumption $u_{\max} > d_{\max}$ and the fact that $t > t^*$ imply $y(t) > 0$. This ends the proof of Theorem 7.4. □

Theorems 7.3 and 7.4 show how to select the buffer capacity in the bottleneck node to fulfill design goals (i) – zero loss rate, and (ii) – full resource usage, in the situation when the propagation latency of the controlled flows is estimated with limited accuracy. Note that the true delay value is not known at the bottleneck node. Therefore, the values given in $\Delta_{\min}$ and $\Delta_{\max}$ should be treated as an indication of the maximum delay deviation that can be tolerated without downgrading the system performance. The increase of the required memory in the error-prone environment relative to the maximum queue length in the ideal case specified by (7.11),

$$(\Delta_{\min} + \Delta_{\max})\,/\,(y_{\mathrm{D}} + u_{\max}T_{\mathrm{C}}) \cdot 100\%, \tag{7.73}$$

constitutes a quantitative measure of the acceptable degree of delay uncertainty. One should also notice a similarity between the conditions for maintaining the maximum throughput specified in Theorems 7.3 and 7.4, and the ones formulated for the uncertain system with time-varying delay in Sect. 6.2.2. In fact, any delay variation within the limits permitted by $\Delta_{\min}$ and $\Delta_{\max}$ will not violate the stability of the system with aperiodic sampling analyzed in this chapter. The enhancements originating from the incoming rate measurements discussed in Sect. 6.2.3 in the context of fixed sampling rate apply also to the systems with variable discretization period investigated in this chapter.

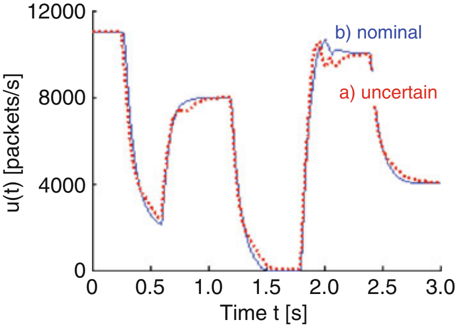


Fig. 7.10 Transmission rate generated by the controller: *a* $\Delta_{\min} = 25$ packets, $\Delta_{\max} = 176$ packets; *b* $\Delta_{\min} = 0$, $\Delta_{\max} = 0$

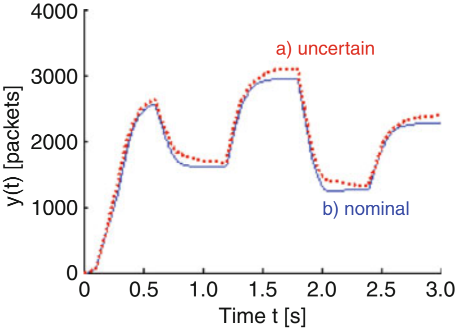


Fig. 7.11 Buffer occupancy: *a* $\Delta_{\min} = 25$ packets, $\Delta_{\max} = 176$ packets; *b* $\Delta_{\min} = 0$, $\Delta_{\max} = 0$

7.3.3 Simulation Results

In the tests of the controller robustness, the network model from Sect. 7.2.3 is applied. For the sake of comparison, all the parameters used in simulations in Sect. 7.2.3 remain unchanged. This time, however, it is assumed that the controller determines the RTTs of the regulated flows with decreased accuracy. Thus, the real delays are the same as in the previous test, but the latencies estimated by the controller are modified as follows: $\overline{\mathrm{RTT}}_1 = 21$ ms, $\overline{\mathrm{RTT}}_2 = 86$ ms, $\overline{\mathrm{RTT}}_3 = 98$ ms, and $\overline{\mathrm{RTT}}_4 = 150$ ms. Hence, the buffer space extensions $\Delta_{\min} = 176$ packets and $\Delta_{\max} = 25$ packets. In order to provide full bottleneck node bandwidth usage, the demand queue length is adjusted to the value of $3{,}100 > 3{,}098$ packets. The resulting $u(t)$ and $y(t)$ functions are illustrated in Figs. 7.10 and 7.11 (curve a). Notice that the algorithm preserves its favorable features, i.e., the queue length does not grow beyond the level of 4,225 packets (calculated from (7.52)) and $y(t)$ is strictly positive. This incurs no data loss in the examined system and full bandwidth utilization.

For the sake of comparison, we present in Figs. 7.10 and 7.11 (curve b) the plots of the transmission rate and queue length obtained for the ideal case, i.e., when the controller determines the delays accurately ($\Delta_{\min} = 0$ and $\Delta_{\max} = 0$). It is clear from the graphs that in order to take into account imperfections in latency estimation and, at the same time, keep the throughput at the maximum, the buffer space needs to be increased. However, despite large differences in the delay estimates (in the range of 22–30% from the actual values), the required buffer extension (7.73) amounts only to $(25 + 176)/4{,}030.100\% \approx 5\%$.

7.4 Feed-Forward Bandwidth Compensation

With the growing popularity of multimedia services in recent years (video and audio streaming, interactive telephony, etc.), much stress has been placed on enhancing the QoS of data transmission in the traditionally best-effort delivery networks. For this type of traffic, the variation of data transfer latency becomes a critical parameter. The delay variation, or delay jitter, originates mainly from different queuing times for the subsequent chunks of data stream at the nodes. Therefore, the control strategies regulating the flow of delay-sensitive, multimedia traffic should attempt to minimize the packet queue length fluctuations. Further in this chapter, a new controller is defined, which not only prevents data losses and ensures full resource usage but also decreases the steady-state queue length changes caused by variations of the available bandwidth. The mechanism employed to reduce the queue length drift from the target value in the presence of persistent positive bandwidth is based on the measurements of the utilized bandwidth. The applied feed-forward disturbance compensation allows for eliminating the error at the output and thus decreases the queuing time differences for subsequent packets. Consequently, the delay jitter is reduced.

7.4.1 Flow Control Strategy

The overall transmission rate established by the controller, $u(t)$, is distributed among the flows according to (7.9), and it is subject to the constraint specified in (7.6), i.e., $0 \leq u(t) \leq u_{\max}$. In order to reduce the queue length sensitivity to the available bandwidth, a different function needs to be applied in place of $\omega(\cdot)$. The following modification is proposed:

$$\omega(t) = \omega_{\mathrm{FF}}(t) = \Gamma\left[y_{\mathrm{D}} - y(t) - \Omega(t) + \varepsilon h(t)\mathrm{RTT}_{\mu}\right]. \tag{7.74}$$

The last term in Eq. (7.74) represents the feed-forward bandwidth compensation, which is used to throttle queue length fluctuations. The influence of the compensation on the system dynamics is tuned by a nonnegative real constant ε. When $\varepsilon = 0$, no compensation is applied, and as ε rises, the significance of the feed-forward term increases. Symbol RTT_{μ} represents the mean RTT of the flows defined in (7.10). For the purpose of exposition, it is assumed in the property derivation that the controller has exact knowledge of the true propagation delays in the network. The presented results extend in a straightforward way to the uncertain delay case using the methods discussed in Sect. 7.3.

7.4.2 Properties of the Proposed Strategy

We begin our analysis with specifying the conditions that will guarantee no data loss and full bandwidth utilization in the considered network. These conditions are given in the following two theorems.

Theorem 7.5. *If the sources in the considered network transmit data according to the conditions formulated in (7.1), (7.3), and (7.9), with function $\omega(\cdot) \triangleq \omega_{FF}(\cdot)$ determined by (7.74), then for any $t \geq 0$, the queue length at the bottleneck node does not exceed the value given below*:

$$y(t) \leq y_{\mathrm{D}} + u_{\max} T_{\mathrm{C}} + \varepsilon d_{\max} \mathrm{RTT}_{\mu}. \tag{7.75}$$

Proof. No packets arrive at the bottleneck node before T_{Fmin}. Similarly as in the analysis conducted earlier in this chapter (the proof of Theorem 7.1), for $i = 1$ and $\theta_1 = T_{\mathrm{Fmin}}$, we can write $y(\theta_1) = 0 < y_{\mathrm{D}} + u_{\max} T_{\mathrm{C}} + \varepsilon d_{\max} \mathrm{RTT}_{\mu}$. Thus, the proposition is valid for any moment of time $t \leq T_{\mathrm{Fmin}}$.

Let us consider some $i > 1$ and the value of signal $\omega_{\mathrm{FF}}(\cdot)$ at a time instant θ_i representing a possible change in the packet incoming rate at the node. Two cases need to be considered: the situation when $\omega_{\mathrm{FF}}(\theta_i) \geq 0$ and the other when $\omega_{\mathrm{FF}}(\theta_i) < 0$.

Case 1. Investigating the circumstances when $\omega_{\mathrm{FF}}(\theta_i) \geq 0$, we get

$$\omega_{\mathrm{FF}}(\theta) = \Gamma\left[y_{\mathrm{D}} - y(\theta_i) - \Omega(\theta_i) + \varepsilon h(\theta_i) \mathrm{RTT}_{\mu}\right] \geq 0, \tag{7.76}$$

which after the term rearrangement becomes

$$y(\theta_i) \leq y_{\mathrm{D}} - \Omega(\theta_i) + \varepsilon h(\theta_i) \mathrm{RTT}_{\mu}. \tag{7.77}$$

Since $\Omega(\theta_i) \geq 0$ and $h(\theta_i) \leq d_{\max}$,

$$y(\theta_i) \leq y_{\mathrm{D}} - 0 + \varepsilon h(\theta_i) \mathrm{RTT}_{\mu} \leq y_{\mathrm{D}} + \varepsilon d_{\max} \mathrm{RTT}_{\mu}. \tag{7.78}$$

Applying (7.78) to (7.13), we get the following estimate of the queue length in the bottleneck node buffer at some time instant $t = \theta_i + \eta \in [\theta_i, \theta_{i+1})$:

$$y(\theta_i + \eta) \leq y_{\mathrm{D}} + \varepsilon d_{\max} \mathrm{RTT}_{\mu} + \Phi(\theta_i, \theta_i + \eta) - \int_{\theta_i}^{\theta_i + \eta} h(\tau)\,\mathrm{d}\tau. \tag{7.79}$$

According to (7.15), $\Phi(\theta_i,\ \theta_i+\eta) \le u_{\max}\eta$. Consequently, since the utilized bandwidth is always nonnegative and $\eta < T_{\mathrm{C}}$,

$$y\left(\theta_i + \eta\right) \le y_{\mathrm{D}} + \varepsilon d_{\max}\mathrm{RTT}_\mu + u_{\max}\eta - 0 = y_{\mathrm{D}} + \varepsilon d_{\max}\mathrm{RTT}_\mu + u_{\max}T_{\mathrm{C}}, \tag{7.80}$$

which completes the first part of the proof.

Case 2. Now, let us examine the situation when $\omega_{\mathrm{FF}}(\theta_i) < 0$. First, we find the last moment $t^* < \theta_i$, when signal $\omega_{\mathrm{FF}}(\cdot)$ was greater than zero. Indeed, such an instant exists since

$$\omega_{\mathrm{FF}}\left(t \le T_{\mathrm{F\,min}}\right) = \Gamma\left(y_{\mathrm{D}} - 0 - 0 + 0\right) = \Gamma y_{\mathrm{D}} > 0. \tag{7.81}$$

Notice also that the first moment when function $\omega_{\mathrm{FF}}(\cdot)$ may attain a negative value is greater than T_{Fmin}, and all the flows are subject to the control action and transmission rate updates at least every T_{C}. The value of $\omega_{\mathrm{FF}}(t^*)$ satisfies the following inequality:

$$\omega_{\mathrm{FF}}\left(t^*\right) = \Gamma\left[y_{\mathrm{D}} - y\left(t^*\right) - \Omega\left(t^*\right) + \varepsilon h\left(t^*\right)\mathrm{RTT}_\mu\right] > 0. \tag{7.82}$$

Consequently,

$$y\left(t^*\right) < y_{\mathrm{D}} + \varepsilon h\left(t^*\right)\mathrm{RTT}_\mu - \Omega\left(t^*\right). \tag{7.83}$$

The expression for the queue length at a time instant $t \in [\theta_i,\ \theta_{i+1})$ is identical to (7.26). Therefore, applying (7.83) to (7.26) results in

$$y(t) < y_{\mathrm{D}} + \varepsilon h\left(t^*\right)\mathrm{RTT}_\mu - \Omega\left(t^*\right) + \Phi\left(t^*, t\right) - \int_{t^*}^{t} h\left(\tau\right)\mathrm{d}\tau. \tag{7.84}$$

According to (7.30), the difference between the number of the incoming packets in the interval $(t^*,\ t]$ and the number of the predicted in-flight packets at instant t^*, $\Phi(t^*,\ t) - \Omega(t^*)$, does not exceed $u_{\max}T_{\mathrm{C}}$. Therefore, since for any t, the utilized bandwidth is subject to the constraint $0 \le h(t) \le d_{\max}$, we get

$$y(t) < y_{\mathrm{D}} + \varepsilon d_{\max}\mathrm{RTT}_\mu + u_{\max}T_{\mathrm{C}}. \tag{7.85}$$

This completes the proof. □

Theorem 7.6. *If the sources in the considered network transmit data according to the conditions formulated in (7.1), (7.3), and (7.9), together with function* $\omega(\cdot) \triangleq \omega_{\mathrm{FF}}(\cdot)$ *defined by (7.74), the maximum rate* $u_{max} > d_{max}$ *and the demand queue length satisfies inequality (7.32), then for any* $t > T_{Fmax} + T_{\mathrm{C}} + T_{max}$*, where* $T_{max} = (y_{\mathrm{D}} + u_{max}T_{\mathrm{C}} + \varepsilon d_{max}\mathrm{RTT}_\mu)/(u_{max} - d_{max})$*, the queue length is always greater than zero.*

Proof. The assumptions of Theorem 7.6 match those of Theorem 7.2. Since the utilized bandwidth is always nonnegative, and the rate history is taken into account in (7.74) in the same way as in (7.7), the proposition holds as a consequence of Theorem 7.2. This concludes the proof. □

Remark 7.2. The steady-state queue length y_{ss}, i.e., the queue length, when the available bandwidth $d_{\mathrm{ss}} > 0$ is constant, can be expressed in the following way:

$$y_{\mathrm{ss}} = y_{\mathrm{D}} - d_{\mathrm{ss}}/\Gamma - (1-\varepsilon)\, d_{\mathrm{ss}}\mathrm{RTT}_{\mu}. \tag{7.86}$$

When

$$\varepsilon = 1 + \left(\Gamma\mathrm{RTT}_{\mu}\right)^{-1}, \tag{7.87}$$

then the steady-state queue length $y_{\mathrm{ss}} = y_{\mathrm{D}}$, which implies complete insensitivity of y_{ss} to the available bandwidth at the bottleneck link. As the influence of d_{ss} on the queue length diminishes, the delay jitter for the data transferred in a stream is reduced. This helps achieve better QoS in the network and enables provision of a broad class of services related to video and audio streaming.

In the next section, the discussed properties will be verified in a simulation scenario.

7.4.3 Simulation Results

Parameters used in the simulations are adjusted to the values introduced in Sect. 7.2.3. It is assumed that the controller possesses precise knowledge of the delays for the regulated connections, i.e., $\mathrm{RTT}_1 = 30$ ms, $\mathrm{RTT}_2 = 70$ ms, $\mathrm{RTT}_3 = 80$ ms, and $\mathrm{RTT}_4 = 120$ ms. This time, however, the rate estimation function is replaced by that defined in (7.74). The feed-forward tuning coefficient is set according to (7.87) as

$$\varepsilon = 1 + \left(\Gamma\mathrm{RTT}_{\mu}\right)^{-1} = 1 + (11.036 \cdot 0.075)^{-1} = 2.208. \tag{7.88}$$

The demand queue length, selected according to Theorem 7.6, is adjusted to 2,930 packets. Consequently, in order to eliminate the risk of losses in the network, the buffer of 5,686 packets should be reserved at the bottleneck node to store the incoming data, as stated in Theorem 7.5. The rate established by the controller is illustrated in Fig. 7.12 and the queue length in Fig. 7.13. The $y(t)$ evolution presented in Fig. 7.13 (curve a) shows that the buffer capacity is not exceeded, which means that packets are never dropped. The queue length is positive for $t > 0.01$ s, and, hence, all of the available bandwidth at the output link of the node is used for data transfer. If we compare the obtained plot with the curve from Fig. 7.9, we can notice

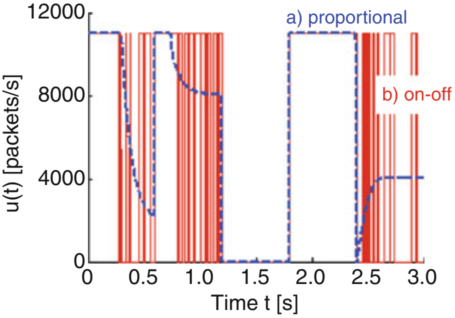


Fig. 7.12 Transmission rate established by: *a* proportional, *b* on-off controller

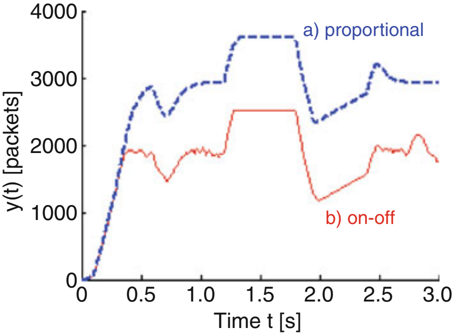


Fig. 7.13 Buffer occupancy: *a* proportional, *b* on-off controller

that when the feed-forward term is introduced in rate computations, $y(t)$ approaches the level of $y_D = 2{,}930$ packets in the steady state, i.e., in the intervals [0.5 s, 0.6 s], [0.9 s, 1.2 s], and [2.6 s, 3.0 s]. We can also see from the plot shown in Fig. 7.13 that in the interval [1.3 s, 1.8 s], the queue length is constant and close to $3{,}600 > 2{,}930$ packets. This is a consequence of small bandwidth in the indicated interval, which does not allow the packets to leave the buffer and to stabilize the queue length at the demand value. Once the available bandwidth grows at 1.8 s, the queue length changes, ultimately approaching 2,930 packets in the steady state.

As the controller gain is increased, one may expect faster convergence to the steady state and, according to (7.32) and (7.75), also smaller buffer capacity requirements. In order to verify this statement, we repeat the test for the on-off controller (4.7) (with $\sigma(t)$ replaced by $\omega_{FF}(t)$), which can be interpreted as the proportional controller with saturation and infinite gain. The demand queue length is adjusted as 1,930 packets, which corresponds to the value calculated from (7.32) with $\Gamma = \infty$. Curve b in the figures reflects the behavior of the on-off scheme. The queue length obtained from the operation of this controller is indeed smaller than in the case of the proportional one. This allows us to decrease the packet waiting time in the queue and allocate smaller buffers. However, the reduced buffer size and smaller queuing time come at a price of nonsmooth rate transitions and oscillations at the output. If one compares the queue length evolution in the case of continuous feedback delivery illustrated in Fig. 4.16 with the one plotted in Fig. 7.13, one can notice small-amplitude oscillations in $y(t)$ in the case of finite, nonuniform sampling rate. However, even though the performance slightly degrades, the management overhead is reduced as fewer feedback carriers need to be transmitted along the data traffic (which is of vital importance in the circumstances of insufficient bandwidth).

Simulation results presented in this section demonstrate that using a feed-forward term in the rate calculation allows one to decrease the queue length sensitivity to the available bandwidth at the bottleneck link, which is traditionally obtained with PI controllers. PI schemes, however, tend to introduce overshoots and oscillations of the queue length, which should be avoided to keep the throughput variations low. We can see no oscillations in the plot in Fig. 7.13 representing the operation of the proportional controller with saturation (curve a), which further helps in reducing the delay jitter. This makes the presented controller a better candidate for enhancing QoS than the on-off scheme (curve b) or the PI one.

7.5 Nonpersistent Sources

The control strategies presented so far in this chapter assume that the lower bound of the assigned rate is fixed at the zero level, leaving only the upper saturation limit flexible for the tuning purposes. Similarly, the network model assumes the minimum available bandwidth at the bottleneck link to be equal to zero. However, this is not always the case in real networks as the link configuration may allow some capacity to be always accessible for serving the data traffic. Therefore, in this section, a more general situation will be analyzed, i.e., the circumstances when the bandwidth may change from some $d_{\min}$ to an upper bound $d_{\max}$, and either of the controller rate saturation limits can be adjusted according to the existing system resources (e.g., link transfer capacity and node memory).

Once the modeling extension is introduced, we will move on to the key issue considered in this section, which concerns nonpersistent nature of data sources. Notice that the controllers described so far in this chapter guarantee achieving the maximum throughput in the communication system only if the sources send data exactly at the rate determined by the controller. In real networks, however, the sources cannot always follow the controller command and may emit packets at a rate lower than the one assigned by the algorithm due to instantaneous or permanent transfer limitations. This may happen, for instance, if the source temporarily stops sending the controlled traffic (e.g., a file transferred in the background) in order to deliver an e-mail or a high-priority business report. The smaller transmission rate of the sources may also result from the congestion occurring at some other node on the data path (which is a situation that cannot be anticipated at the moment of rate assignment at the controlling node). Therefore, in this section, the nonideal behavior of data sources is explicitly taken into account in the controller performance analysis. Also appropriate conditions for obtaining full bandwidth utilization are specified and strictly proved.

7.5.1 Model Extensions

The network model considered in this section is similar to the one introduced in Sect. 7.1. However, in order to study the phenomena related to the existence of nonpersistent data sources (and distributed congestion problem) in the context of maintaining high throughput, certain modifications need to be introduced. The necessary model changes are indicated below.

First of all, it is assumed that the node configuration allows for reserving certain part of the bottleneck link capacity to be always accessible by the flows. Consequently, here, we consider the situation when $d(\cdot)$ is lower-bounded by a positive real constant $d_{\min}$ instead of zero. Similarly as was analyzed before, the available bandwidth is limited from above by a positive constant $d_{\max} \geq d_{\min}$ ($d_{\max} = d_{\min}$ reflects the case of constant bandwidth). Thus, for any time instant t, the function representing bandwidth variations satisfies the following inequalities:

$$0 < d_{\min} \leq d(t) \leq d_{\max}. \tag{7.89}$$

The sources attempt to adjust the packet transmission rate to the value determined by the algorithm implemented at the bottleneck node. However, they cannot always obey the controller command due to transfer limitations. Therefore, since the sources begin the transmission at the time $t = 0$, the rate of source p, $u_p(t)$, can be expressed as (compare with (7.1))

$$\underset{p}{\forall} \ \underset{t<0}{\forall} \ u_p\,(t) = 0 \text{ and } \underset{p}{\forall} \ \underset{t\geq 0}{\forall} \ u_p\,(t) = f_p\,(t)\, v_p\left(t - T_{\mathrm{B}}^{p}\right), \tag{7.90}$$

where function $f_p(t)$, representing the transfer limitations of source p, is subject to the constraint

$$0 < f_{\min} \leq f_p(t) \leq 1, \tag{7.91}$$

for some $f_{\min} \in (0,\ 1]$. Although the sources cannot always transmit data at the rate established by the controller, it is assumed that in order to provide basic responsiveness to the changing networking conditions, $u_p(t)$ cannot be lower than the minimum rate $\lambda_p u_{\min}$, where $0 < u_{\min} \leq d_{\min}$, so that control unit emission and rate adaptation occurs at least every

$$T_{\mathrm{C}} = N / \min_{p=1,2,\ldots,m} \left(\lambda_p u_{\min}\right). \tag{7.92}$$

Consequently, the fixed constraint on the control unit emission interval defined in (7.4) is replaced with the implicit condition (7.92) related to the minimum level of network transfer guarantee $u_{\min}$.

7.5.2 Flow Control Strategy

The following control strategy is proposed to regulate the flow of data in the network with nonpersistent sources modeled according to the description provided in Sect. 7.5.1. The overall transmission rate generated by the controller, $u(t)$, is distributed among the flows according to (7.9). The overall rate is subject to the following constraint:

$$u(t) = \begin{cases} u_{\min}, & \text{if} \quad \omega(t) < u_{\min}, \\ \omega(t), & \text{if} \quad u_{\min} \leq \omega(t) \leq u_{\max}, \\ u_{\max}, & \text{if} \quad \omega(t) > u_{\max}, \end{cases} \tag{7.93}$$

where $u_{\min}$ and $u_{\max}$ denote the limits of possible flow rate values and $0 < u_{\min} < u_{\max}$. Function $\omega(t)$ is given by (7.7), i.e.,

$$\omega(t) = \Gamma\left[y_{\mathrm{D}} - y(t) - \Omega(t)\right] = \Gamma\left[y_{\mathrm{D}} - y(t) - \sum_{p=1}^{m} \int_{t-\mathrm{RTT}_p}^{t} v_p(\tau)\,\mathrm{d}\tau\right],$$

and rate $v_p(\cdot)$, sent and recorded by the node for source p at the instant of a control unit passing through the node, is determined from the following equation:

$$v_p(t) = \begin{cases} 0, & \text{for} \quad t < -T_{\mathrm{B}}^{p}, \\ \lambda_p u_{\min}, & \text{for} \quad -T_{\mathrm{B}}^{p} \leq t < T_{\mathrm{F}}^{p}, \\ \lambda_p u\left(t_{p,k} - T_{\mathrm{B}}^{p}\right), & \text{for} \quad t \geq T_{\mathrm{F}}^{p} \quad \text{and} \quad t \in \left[t_{p,k} - T_{\mathrm{B}}^{p}, t_{p,k+1} - T_{\mathrm{B}}^{p}\right). \end{cases} \tag{7.94}$$

Therefore, initially (in the [0, RTT_p) interval), source p sends data at the minimum rate $\lambda_p u_{\min}$. Afterwards, it is allowed to deliver packets at the rate determined by the controller at discrete-time instants $t_{p,k} - T_{\mathrm{B}}^{p}$ according to relations (7.7) and (7.93). Let us recall that $t_{p,k}$ is the kth moment of time when a control unit belonging to flow p brings the feedback information to source p.

Using (7.93) and (7.94), the term representing the in-flight data in (7.7) is subject to the constraint

$$u_{\min}\mathrm{RTT}_{\mu} \leq \Omega(t) \leq u_{\max}\mathrm{RTT}_{\mu}, \tag{7.95}$$

where $\mathrm{RTT}_{\mu} = \sum_{p=1}^{m} \lambda_p \mathrm{RTT}_p$ denotes the mean RTT of the flows passing through the bottleneck link. Moreover, it is assumed that in order for the network to be able to transport the users' data at least at the minimum rate, $u_{\min}$ should not exceed $d_{\min}$. On the other hand, to make it possible to always exploit all of the available bandwidth, it is expected that $u_{\max} \geq d_{\max}$. Therefore, since the utilized bandwidth is limited by either the available bandwidth or the incoming flow rate (when the queue length is zero), for any $t \geq 0$, $h(t)$ satisfies the following inequalities:

$$0 < u_{\min} \leq h(t) \leq d_{\max} \leq u_{\max}. \tag{7.96}$$

Next, appropriate conditions allowing us to achieve loss elimination and full available bandwidth usage will be formulated and proved analytically with the explicit consideration of the nonpersistent nature of data sources.

7.5.3 Properties of the Proposed Strategy

From the implementation point of view of any flow control algorithm, it is necessary to specify the buffer space that needs to be assigned at the controlling node in order to store all the incoming packets before they are further relayed to the subsequent node. The proposition formulated below defines the buffer size required to accommodate the maximum packet queue length.

Theorem 7.7. *If strategy (7.93) and (7.94) is applied to control the flow of data in the considered network, then for any $t \geq 0$, the queue length at the bottleneck node is upper-bounded by y_{max}, where*

$$y_{\max} = y_{\mathrm{D}} - u_{\min}\left(\mathrm{RTT}_{\mu} + \Gamma^{-1}\right) + (u_{\max} - u_{\min})\,T_{\mathrm{C}}. \tag{7.97}$$

Proof. The rate of source p is adjusted at $t_{p,k}$ time instants. The effect of these modifications influences the total arrival rate at the controlling node with forward delay at instants θ_i, where $i = 1, 2, 3, \ldots$ and $\theta_1 = T_{\mathrm{Fmin}}$.

The bottleneck node buffer is empty until the first packets arrive at $t = \theta_1 = T_{\mathrm{Fmin}}$. In consequence, the theorem holds for any $t \leq T_{\mathrm{Fmin}}$. Let us consider some $i > 1$ and a moment of time $t > T_{\mathrm{Fmin}}$ belonging to the interval $[\theta_i,\ \theta_{i+1})$. The queue length $y(t = \theta_i + \eta)$, where $\eta \in [0, \alpha_i)$ with α_i defined in (7.12), can be expressed as in (7.13). Taking into account the rate limitations specified in (7.93), and the assumption that $u_p(t)$ cannot be lower than the minimum rate $\lambda_p u_{\min}$, function $\Phi(\theta_i,\ \theta_i + \eta)$, representing the amount of the data arriving at the node between θ_i and $\theta_i + \eta$,

$$\Phi\left(\theta_i, \theta_i + \eta\right) = \sum_{p=1}^{m} \int_{\theta_i}^{\theta_i + \eta} u_p\left(\tau - T_{\mathrm{F}}^{p}\right) \mathrm{d}\tau = \sum_{p=1}^{m} \int_{\theta_i}^{\theta_i + \eta} f_p\left(\tau - T_{\mathrm{F}}^{p}\right) v_p\left(\tau - \mathrm{RTT}_p\right) \mathrm{d}\tau, \tag{7.98}$$

is subject to the following constraint:

$$u_{\min}\eta \leq \Phi\left(\theta_i, \theta_i + \eta\right) \leq u_{\max}\eta. \tag{7.99}$$

The lower bound of $\Phi(\cdot,\cdot)$ reflects the most congested network state, i.e., the situation when the transmission rate is forced to $u_{\min}$. The upper limit, in turn, is expected in the situation when all the sources follow exactly the controller command (for every p, function $f_p(\cdot) \equiv 1$), which was to transmit at the maximum rate $u_{\max}$.

In order to analyze the queue length variations in the interval $[\theta_i, \theta_{i+1})$, we examine the behavior of function $\omega(\cdot)$. Two complementary cases will be considered: first, the situation when $\omega(\theta_i) \geq u_{\min}$, and afterwards, the circumstances when $\omega(\theta_i) < u_{\min}$.

Case 1. In the worst case, we analyze the situation when $\omega(\theta_i) \geq u_{\min}$. From the definition of function $\omega(\cdot)$, we get

$$\omega\left(\theta_i\right) = \Gamma\left[y_D - y\left(\theta_i\right) - \Omega\left(\theta_i\right)\right] \geq u_{\min}, \tag{7.100}$$

which after the term rearrangement reduces to

$$y\left(\theta_i\right) \leq y_D - u_{\min}/\Gamma - \Omega\left(\theta_i\right). \tag{7.101}$$

Substituting (7.101) for $y(\theta_i)$ in (7.13) results in

$$y\left(\theta_i + \eta\right) \leq y_{\mathrm{D}} - u_{\min}/\Gamma - \Omega\left(\theta_i\right) + \Phi\left(\theta_i, \theta_i + \eta\right) - \int_{\theta_i}^{\theta_i + \eta} h\left(\tau\right) \mathrm{d}\tau. \tag{7.102}$$

The biggest amount of data arrives at the node if all the sources are able to follow the controller command exactly, i.e., if for every p, function $f_p(\cdot)$ equals one. Then, the formula for the incoming data (7.98) reduces to

$$\Phi\left(\theta_i, \theta_i + \eta\right) = \sum_{p=1}^{m} \int_{\theta_i - \mathrm{RTT}_p}^{\theta_i + \eta - \mathrm{RTT}_p} v_p\left(\tau\right) \mathrm{d}\tau, \tag{7.103}$$

and the difference $\Phi(\theta_i, \theta_i + \eta) - \Omega(\theta_i)$ in (7.102) can be evaluated as

$$\Phi\left(\theta_i, \theta_i + \eta\right) - \Omega\left(\theta_i\right) = \sum_{p=1}^{m} \int_{\theta_i}^{\theta_i + \eta - \mathrm{RTT}_p} v_p\left(\tau\right) \mathrm{d}\tau. \tag{7.104}$$

Since $u_p(t) \leq \lambda_p u_{\max}$, we get $\Phi(\theta_i, \theta_i + \eta) - \Omega(\theta_i) \leq u_{\max}(\eta - \mathrm{RTT}_\mu)$. According to (7.96), the utilized bandwidth $h(t) \geq u_{\min}$. Therefore, since η is upper-bounded by T_{C} and $u_{\max}$ is greater than $u_{\min}$, we may estimate the buffer queue length as

$$\begin{aligned} y\left(\theta_i + \eta\right) &\leq y_{\mathrm{D}} - u_{\min}/\Gamma + u_{\max}\left(T_{\mathrm{C}} - \mathrm{RTT}_\mu\right) - u_{\min} T_{\mathrm{C}} \\ &= y_{\mathrm{D}} - u_{\min}\left(\mathrm{RTT}_\mu + \Gamma^{-1}\right) + \left(u_{\max} - u_{\min}\right)\left(T_{\mathrm{C}} - \mathrm{RTT}_\mu\right) < y_{\max}, \end{aligned} \tag{7.105}$$

which ends the first part of the proof.

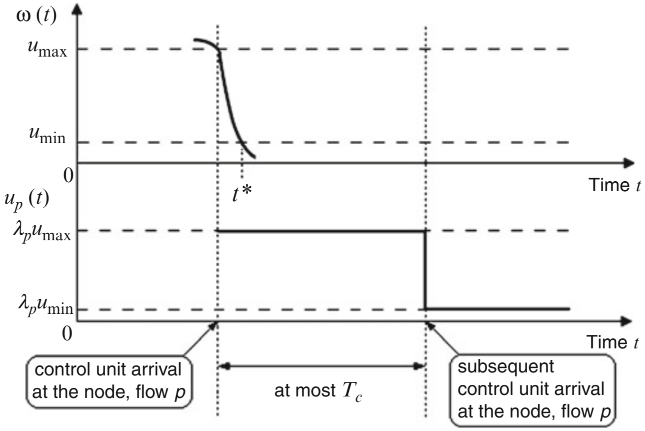


Fig. 7.14 Transmission rate assignment in the vicinity of t^* – rate decrease

Case 2. Now, let us examine the situation when $\omega(\theta_i) < u_{\min}$. We begin with finding the last moment $t^* < \theta_i$ when signal $\omega(\cdot)$ was greater than $u_{\min}$. From the algorithm definition, we get

$$\omega\left(t \le -T_{\mathrm{B}\max}\right) = \Gamma\left(y_{\mathrm{D}} - 0 - 0\right) > u_{\max} > u_{\min}, \tag{7.106}$$

where $T_{\mathrm{B}\max} = \max_{p=1,2,\ldots,m}(T_{\mathrm{B}}^{p})$. For $t \in (-T_{\mathrm{Bmax}}, T_{\mathrm{Fmin}})$, signal $\omega(\cdot)$ decreases with time and, at the upper limit of this interval, has a value greater than $u_{\min}$, i.e., $\omega(T_{\mathrm{Fmin}}) > u_{\min}$. This means that moment t^* actually exists and that $t^* > T_{\mathrm{Fmin}}$. Since $\omega(t^*) > u_{\min}$, we have $\omega(t^*) = \Gamma[y_{\mathrm{D}} - y(t^*) - \Omega(t^*)] > u_{\min}$, which implies

$$y\left(t^*\right) < y_{\mathrm{D}} - u_{\min}/\Gamma - \Omega\left(t^*\right). \tag{7.107}$$

The queue length at a time instant $t \in [\theta_i, \theta_{i+1})$ can be expressed as in (7.26). Substituting (7.107) into (7.26), we obtain

$$y(t) < y_{\mathrm{D}} - u_{\min}/\Gamma - \Omega\left(t^*\right) + \Phi\left(t^*, t\right) - \int_{t^*}^{t} h\left(\tau\right)\mathrm{d}\tau. \tag{7.108}$$

The difference between the maximum number of packets arriving at the node between t^* and t and the estimated amount of the in-flight packets, $\Phi(t^*, t) - \Omega(t^*)$, can be expressed as in (7.28). It is clear from (7.95) that $\sum_{p=1}^{m} \int_{t-\mathrm{RTT}_p}^{t} v_p(\tau)\mathrm{d}\tau \ge u_{\min}\mathrm{RTT}_{\mu}$. Moreover, since we consider the case when the transfer speed generated by the controller in the interval $(t^*, t]$ is equal to $u_{\min}$, and its assignment can be delayed by T_{C} (see Fig. 7.14), we have

$$\sum_{p=1}^{m} \int_{t^*}^{t} v_p(\tau)\,\mathrm{d}\tau \le u_{\max} T_{\mathrm{C}} + u_{\min}\left(t - t^* - T_{\mathrm{C}}\right). \tag{7.109}$$

Therefore, the difference $\Phi(t^*, t) - \Omega(t^*)$ can be estimated as

$$\Phi\left(t^*, t\right) - \Omega\left(t^*\right) \le u_{\max} T_{\mathrm{C}} + u_{\min}\left(t - t^* - T_{\mathrm{C}}\right) - u_{\min}\mathrm{RTT}_{\mu}. \tag{7.110}$$

Since the value of function $h(\cdot)$ does not drop below $u_{\min}$, using (7.110), we get

$$y(t) < y_{\mathrm{D}} - u_{\min}\left(\mathrm{RTT}_{\mu} + \Gamma^{-1}\right) + \left(u_{\max} - u_{\min}\right) T_{\mathrm{C}} = y_{\max}. \tag{7.111}$$

This concludes the proof. □

Full link utilization is guaranteed by the presence of data packets in the bottleneck node buffer at any instant of time. Since the sources may deliver fewer packets than expected by the controller, an appropriate condition ensuring full resource usage must be specified. The theorem presented below shows how the demand queue length y_{D} should be selected so that entire bandwidth at the output connection is utilized for data transfer.

Theorem 7.8. *If strategy (7.93) and (7.94) is applied to control the flow of data in the considered network, $f_{min}u_{max} > d_{max}$, and the demand value of the queue length satisfies the following inequality*:

$$y_{\mathrm{D}} > u_{\max}\left(\mathrm{RTT}_{\mu} + \Gamma^{-1}\right) + \left(f_{\min}u_{\max} - u_{\min}\right) T_{\mathrm{C}}, \tag{7.112}$$

then for any $t > T_{Fmax} + T_C + T_{max}$, where $T_{max} = y_{max}/(f_{min}u_{max} - d_{max})$, the queue length is always greater than zero.

Proof. The theorem assumption implies that we deal with time instants $t > T_{\mathrm{Fmax}} + T_{\mathrm{C}} + T_{\max} > \theta_1$. Considering some $i > 1$ and $\omega(\theta_i)$, we may distinguish two cases: the situation when $\omega(\theta_i) < u_{\max}$ and the circumstances when $\omega(\theta_i) \ge u_{\max}$.

Case 1. First, we consider the situation when $\omega(\theta_i) < u_{\max}$. Taking similar steps as in the proof of Theorem 7.2, namely, (7.34)–(7.36), we get $y(\theta_i) > y_{\mathrm{D}} - u_{\max}(\mathrm{RTT}_{\mu} + \Gamma^{-1})$. Inequality (7.112) and the fact that $f_{\min}$ $u_{\max} > d_{\max} \ge u_{\min}$ allow us to further estimate the queue length at instant θ_i as

$$\begin{aligned} y\left(\theta_i\right) &> y_{\mathrm{D}} - u_{\max}\left(\mathrm{RTT}_{\mu} + \Gamma^{-1}\right) \\ &> u_{\max}\left(\mathrm{RTT}_{\mu} + \Gamma^{-1}\right) + \left(f_{\min}\,u_{\max} - u_{\min}\right) T_{\mathrm{C}} - u_{\max}\left(\mathrm{RTT}_{\mu} + \Gamma^{-1}\right) \\ &= \left(f_{\min}\,u_{\max} - u_{\min}\right) T_{\mathrm{C}} > 0. \end{aligned} \tag{7.113}$$

Applying (7.113) to formula (7.13) for the queue length at a time instant $t \in [\theta_i, \theta_{i+1})$, we obtain

$$y\left(t = \theta_i + \eta\right) > \left(f_{\min} u_{\max} - u_{\min}\right) T_{\mathrm{C}} + \Phi\left(\theta_i, \theta_i + \eta\right) - \int\limits_{\theta_i}^{\theta_i+\eta} h\left(\tau\right) \mathrm{d}\tau. \quad (7.114)$$

The minimum incoming rate equals $u_{\min}$. From (7.99), we get $\Phi(\theta_i, \theta_i + \eta) \geq u_{\min}\eta$. The utilized bandwidth, in turn, can be as high as $d_{\max} < f_{\min} u_{\max}$. Therefore,

$$\begin{aligned} y\left(\theta_i + \eta\right) &> \left(f_{\min} u_{\max} - u_{\min}\right) T_{\mathrm{C}} + u_{\min}\eta - d_{\max}\eta \\ &= \left(f_{\min} u_{\max} - u_{\min}\right) T_{\mathrm{C}} - \left(d_{\max} - u_{\min}\right)\eta \\ &\geq \left(f_{\min} u_{\max} - u_{\min}\right)\left(T_{\mathrm{C}} - \eta\right). \end{aligned} \quad (7.115)$$

Since $T_{\mathrm{C}} > \eta$, $y(\theta_i + \eta) > 0$, which completes the first part of the proof.

Case 2. Now, let us investigate the situation when $\omega(\theta_i) \geq u_{\max}$. We look for the last moment $t^* < \theta_i$ when signal $\omega(\cdot)$ was smaller than $u_{\max}$. Note that according to Theorem 7.7, the queue length never exceeds $y_{\max}$. The buffer is never depleted faster than at rate $d_{\max}$. On the other hand, in the situation when the assigned rate equals $u_{\max}$, the incoming rate at the node (after taking into account the transfer limitations of the sources) will not be lower than $f_{\min} u_{\max}$. Thus, in order to preserve the buffer space indicated in Theorem 7.7, the controller may continuously set the biggest rate $u_{\max}$ for the maximum period $T_{\max} = y_{\max}/(f_{\min} u_{\max} - d_{\max})$, and instant t^* actually exists. Since t^* was the last instant, when $\omega(\cdot)$ was smaller than $u_{\max}$, and the time separation between any two consecutive aggregate input rate modifications does not exceed T_{C}, we conclude that $t^* \geq \theta_i - (T_{\max} + T_{\mathrm{C}})$. With analogy to (7.113), we obtain

$$y\left(t^*\right) > y_{\mathrm{D}} - u_{\max}/\Gamma - u_{\max}\,\mathrm{RTT}_{\mu} > \left(f_{\min} u_{\max} - u_{\min}\right) T_{\mathrm{C}} > 0. \quad (7.116)$$

The queue length at some time instant $t \in [\theta_i, \theta_{i+1})$ may be expressed as in (7.26). Then, applying (7.116) to (7.26), we get

$$y(t) > \left(f_{\min} u_{\max} - u_{\min}\right) T_{\mathrm{C}} + \Phi\left(t^*, t\right) - \int\limits_{t^*}^{t} h\left(\tau\right) \mathrm{d}\tau. \quad (7.117)$$

Recall that t^* was the last instant before t, when the controller calculated rate smaller than $u_{\max}$. This rate can be as low as $u_{\min}$. Afterwards, the algorithm sets the maximum rate value. Since control units appear at discrete time instants, the rate assignment can be delayed, yet not more than by T_{C} (see Fig. 7.15). Moreover, due to the source transfer limitations, the incoming rate at the node (although predicted

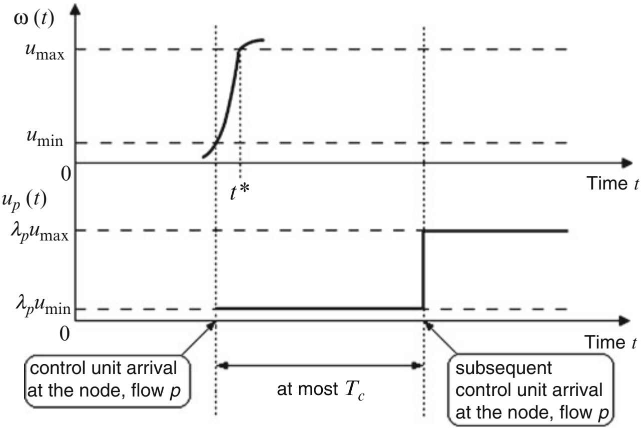


Fig. 7.15 Transmission rate assignment in the vicinity of t^* – rate increase

to be at the maximum) may be as low as $f_{\min}\, u_{\max}$. Thus, the amount of the incoming data $\Phi(\cdot,\cdot)$ will satisfy the following estimate:

$$\Phi\left(t^*, t\right) \geq u_{\min} T_{\mathrm{C}} + f_{\min}\, u_{\max}\left(t - t^* - T_{\mathrm{C}}\right). \tag{7.118}$$

For any t, the utilized bandwidth $h(t) \leq d_{\max}$, and $\int_{t^*}^{t} h(\tau)\mathrm{d}\tau \leq d_{\max}\,(t - t^*)$. Therefore, after substituting (7.118) into (7.117), we arrive at

$$\begin{aligned} y(t) &> (f_{\min}\, u_{\max} - u_{\min})\, T_{\mathrm{C}} + u_{\min} T_{\mathrm{C}} + f_{\min}\, u_{\max}\left(t - t^* - T_{\mathrm{C}}\right) - d_{\max}\left(t - t^*\right) \\ &= (f_{\min}\, u_{\max} - d_{\max})\left(t - t^*\right). \end{aligned} \tag{7.119}$$

The theorem assumptions imply $f_{\min} u_{\max} > d_{\max}$. Consequently, since $t > t^*$, we get $y(t) > 0$. This ends the proof. □

Theorems 7.7 and 7.8 define the buffer space which needs to be allocated for packet storage at the bottleneck node to guarantee the maximum throughput in the system despite possible shortage of the incoming data. In the next section, the performance of the developed flow control algorithm in the network with nonpersistent sources will be illustrated in simulation tests.

7.5.4 Simulation Results

The structure of the network model used in the simulations is identical to the one introduced in Sect. 7.2.3, i.e., four connections characterized by the delays:

Fig. 7.16 Available bandwidth

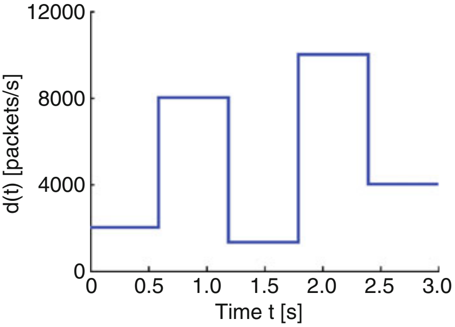
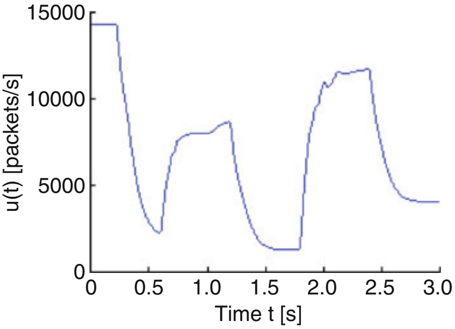


Fig. 7.17 Transmission rate generated by the controller



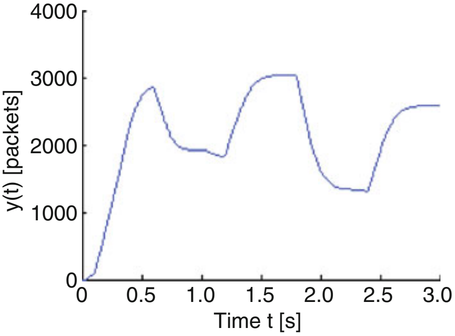
$\mathrm{RTT}_1 = 30$ ms, $\mathrm{RTT}_2 = 60$ ms, $\mathrm{RTT}_3 = 60$ ms, and $\mathrm{RTT}_4 = 90$ ms, pass through the bottleneck node. Similarly as in that section, we assume that each source sends a control unit every $N = 32$ packets.

The minimum available bandwidth $d_{\min}$ is adjusted to 1,280 packets/s and the maximum $d_{\max}$ to 10,000 packets/s. Function $d(\cdot)$ used in the simulations is illustrated in Fig. 7.16. Comparing the bandwidth patterns from Figs. 7.5 and 7.16, we can notice that they differ only in interval [1.2 s, 1.8 s]. The function shown in Fig. 7.5 is equal to zero in this interval, while $d(\cdot)$ from Fig. 7.16 does not fall below the minimum of 1,280 packets/s. The lower bound of the overall source rate was set as $u_{\min} = d_{\min} = 1{,}280$ packets/s, and the upper bound $u_{\max}$ as $14{,}300 > d_{\max}/f_{\min} = 10{,}000/0.7$ packets/s. This means that under the applied equal rate distribution, each source emits a control unit at least every $T_C = mN/u_{\min} = 4{\cdot}32/1{,}280 = 0.1$ s. For the sake of comparison with the results presented earlier in this chapter, the controller gain Γ is adjusted to 11.036 s^{-1}. Finally, in order to ensure full resource usage, the demand queue length calculated according to (7.112) is set as $3{,}245 > 3{,}241$ packets.

The rate determined by the controller is depicted in Fig. 7.17, the buffer occupancy in Fig. 7.18, and the transmission rate of the sources in Fig. 7.19.

As we can see from the graph in Fig. 7.18, the queue length never exceeds the value of 4,335 packets (the maximum calculated according to (7.97)) and does not drop to zero. These two properties imply no buffer overflow and full bottleneck

Fig. 7.18 Buffer occupancy



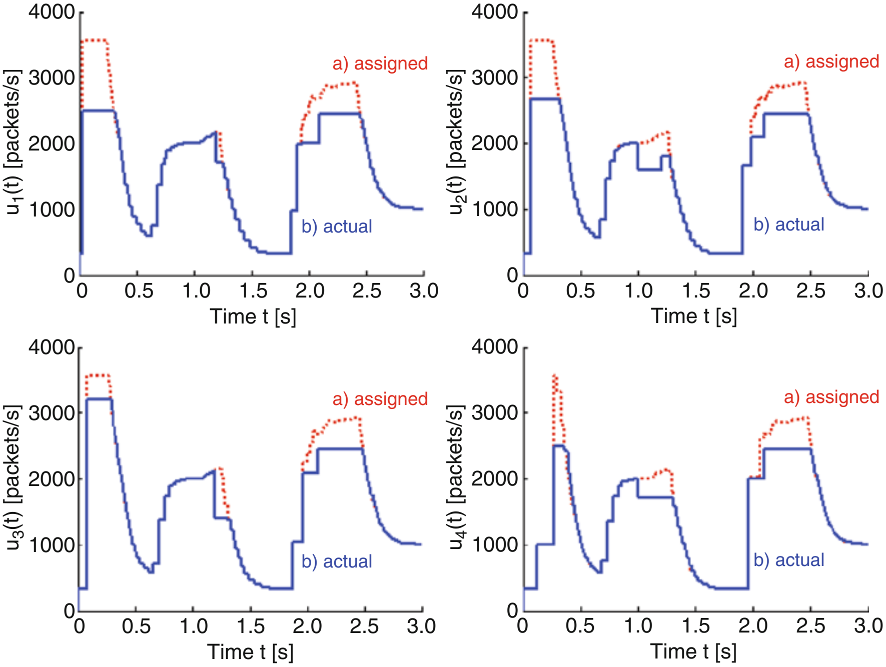


Fig. 7.19 Transmission rate of the sources: *a* assigned, *b* actual transfer speed

link utilization. The rates assigned for the sources (curve a) and the true transfer speeds of each transmitter (curve b) are shown in Fig. 7.19. It is clear from the plots that the sources cannot always deliver data at the rate established by the controller (the actual rate can be as low as 70% of the assigned one). Despite significant rate limitations experienced by the transmitters, the proposed strategy guarantees the maximum throughput in the network.

7.6 Chapter Summary

In this chapter, the flow control problem was addressed in the context of data transmission networks in which the feedback information about the current network state is provided irregularly in time. The networks were modeled as sampled data systems with variable, input-dependent sampling rate. A number of scenarios were considered, and appropriate control algorithms were developed for each case. The presented strategies ensure no data loss and full resource utilization in the analyzed multisource system even though the feedback information necessary for transfer speed adjustment is accessible at irregularly spaced time periods.

In Sect. 7.2, the network model was presented. Next, in Sect. 7.2, the fundamental control law for the system with aperiodic feedback was introduced, and its properties discussed. In Sect. 7.3, the robustness issues were investigated, and it was shown that the designed control strategy operates properly despite possible mismatch occurring between the real propagation delays and those estimated by the controller. Afterwards, in Sect. 7.4, a modified flow regulation scheme was proposed. The controller with an extra feed-forward term incorporated not only maintains the favorable features of the principal strategy but also helps reduce the queue length drift from the target value caused by positive available bandwidth. Thus, as the queuing time of the subsequent parts of data stream is unified, the delay jitter decreases, and the QoS with respect to delay-sensitive traffic may be improved. This facilitates handling the increasingly popular services related to multimedia transmission in the traditionally best-effort delivery networks. Finally, in Sect. 7.5, the situation when data sources cannot always obey the controller command and transmit data at a rate lower than the assigned one was investigated. It was shown that with appropriately modified algorithm parameters (the demand queue length and the rate upper saturation limit), the described control strategy continues achieving the maximum throughput in the communication system. Since the reduced rates may result from the source itself, or may be caused by the congestion occurring elsewhere in the network (and not at the node where the controller operates), the proposed algorithm may efficiently coexist with other, possibly different flow regulation schemes. The described control schemes need neither constant nor time-synchronized exchange of the feedback information. As a result, they are more scalable in the multisource network implementation than the similar schemes requiring continuous or periodic (with fixed sampling rate) feedback information delivery.

References

1. Cheng RG, Chang CJ (1996) Design of a fuzzy traffic controller for ATM networks. IEEE/ACM Trans Netw 4:460–469
2. Jagannathan S, Talluri J (2002) Predictive congestion control of ATM networks: multiple sources/single buffer scenario. Automatica 38:815–820

3. Kim HS, Shin SY, Kwon WH (2004) Feedback control for QoS of mixed traffic in communication networks. Control Eng Pract 12:527–536
4. Laberteaux KP, Rohrs ChE, Antsaklis PJ (2002) A practical controller for explicit rate congestion control. IEEE Trans Autom Control 47:960–978
5. Tan L, Pugh AC, Yin M (2003) Rate-based congestion control in ATM switching networks using a recursive digital filter. Control Eng Pract 11:1171–1181

Chapter 8
Discrete Sliding-Mode Congestion Control in TCP Networks

Before introducing the proposal of Van Jacobson [9], the TCP/IP-based networks suffered from severe congestion problems. The bursts of traffic intensity frequently led to a network breakdown called the congestion collapse, and the resulting throughput degradation by several orders of magnitude. The Jacobson's algorithm, implemented at the connection end points, ensured the basic control mechanism used to regulate the amount of data injected into the network. According to this algorithm, the transfer rate of a TCP source (or more specifically the window size) is increased until the congestion is detected at some link in the network. Initially, the window size at the source is enlarged by the number of packets acknowledged by the receiver. It is called the *slow-start*, or exponential-growth phase, and is used to quickly capture enough bandwidth to transmit the user's data at a sufficiently fast rate. When a certain threshold value is reached, *ssthresh*, the window size continues to grow, but at a slower rate. In this phase, called the *congestion avoidance*, the window is enlarged by one packet every RTT. The transmitter tries to reduce the risk of link buffer overflow at the remote node(s), and the window size increases approximately linearly in time. Usually, the bulk of the user's data is transmitted in this phase (see, e.g., the analysis performed in [19]).

The source rate cannot grow indefinitely. It increases until the maximum window size is reached, or the congestion is detected. When a packet is lost, and as a consequence the timer during which the acknowledgment for this packet should be received expires, the source infers that the congestion is taking place at some point in the network. In such case, the transmitter window size is reset to the initial value and the congestion avoidance threshold *ssthresh* is reduced. Afterwards, the slow-start and congestion avoidance phases repeat with the new *ssthresh* value. As a result of adopting this basic control mechanism, the transfer rate of a TCP source goes through the periods of *additive increase* (congestion avoidance phase) and *multiplicative decrease* (upon detection of packet loss). The fundamental TCP flow regulation scheme constitutes an implicit feedback control system. The sources test the network transfer capabilities until the congestion state is reached, and throttle

P. Ignaciuk and A. Bartoszewicz, *Congestion Control in Data Transmission Networks: Sliding Mode and Other Designs*, Communications and Control Engineering,
DOI 10.1007/978-1-4471-4147-1_8, © Springer-Verlag London 2013

the input rate according to the observations of the stream of acknowledgements obtained from the receiver. No direct action from the network is required to regulate the flow of data.

In time, the fundamental TCP flow control algorithm was augmented by a number of advantageous enhancements. The researchers proposed the use of different recovery schemes (e.g., TCP NewReno, SACK [3]), and other than the packet loss indications of congestion (e.g., TCP Vegas [2] applies RTT measurements to infer the current network state). It has also soon become apparent that the TCP flow control may benefit from additional mechanisms implemented closer to the place where the congestion actually occurs – at the network nodes [5]. By monitoring the queue length, the network node can detect the incipient congestion in a faster and more accurate way than the remote sources. Then, preemptive measures can be taken to throttle the incoming rate before the actual loss of packets takes place. This is the idea behind various AQM algorithms, which take appropriate actions to reduce the risk of congestion on the basis of the observed node state (buffer occupancy, queuing delay, link utilization, etc.). For instance, when the packet queue length in the output link buffer reaches certain value, then the node may selectively discard incoming packets to prevent further input rate increase. Instead of dropping packets according to an AQM scheme, the node can also signal the incipient congestion by marking appropriate bits in the packet headers. The receiver incorporates the congestion information read from the marked bits in the acknowledgments sent for the source. Upon the reception of a marked packet, the source reduces its window size, thus increasing the chance of avoiding packet loss at the bottleneck node. By applying an AQM scheme, and providing an explicit information about their state, the network nodes can actively participate in data flow control. In consequence, the packet losses and queuing delay can be significantly reduced. Typically, a single bit is used for the signaling purposes – the ECN bit in the TCP header. The network-assisted explicit congestion notification, promoted by S. Floyd [4], enforces minimum implementation effort in the existing TCP infrastructure. It requires only a modification in the way the end points interpret the ECN bit. Due to the vulnerability of the single-bit explicit congestion control (well studied in the context of ATM networks), various researchers also advocate the use of two bits [21] or multibit fields [10, 20] for achieving better accuracy and faster convergence rates in data flow control. The serious drawback of the multibit congestion notification is the necessity of making appropriate changes to the existing transfer solutions. However, in order to circumvent the implementation burden of introducing new protocols for multibit congestion notification, one can recur to marking single bits in multiple packets, for example, using the technique highlighted in [8]. For the discussion on the single- and multibit congestion notification, and relations between ATM/ABR and TCP marking schemes, one may refer to a recent publication by Almeida and Belo [1].

In this chapter, we address the problem of efficient data flow control in a TCP/IP network with routers supporting AQM. The network is treated as a feedback system in which the information about the current networking conditions is assumed to be relayed to the sources by means of the ECN bit in the TCP header. First, we briefly discuss the choice of the modeling framework. Then, the selected nonlinear model is treated in greater detail with the emphasis placed on the feedback nature of TCP/AQM transmission system. Following the description of the model structure, we proceed with the linearization and discretization of the associated dynamical equations. Two choices of system configuration are considered. In the first one, the influence of state delay is neglected, whereas in the second one, the information about both the input and state delay in the system is preserved. Consequently, two distinct discrete-time system representations are obtained, and two separate design procedures are conducted. In the control law derivation, the formal approach of discrete-time SM control is applied. As a result, two feedback control laws are developed, each specifying a particular marking scheme to be implemented in the network routers supporting AQM. The controllers are tested in a series of simulations and compared with the classical TCP/AQM marking schemes. The result of the numerical investigation is reported in the last part of the chapter.

8.1 Nonlinear TCP Dynamics

The importance of communication networks incited much research work on the existing, frequently heuristic data transfer solutions and underlying technologies. The key to the successful development of new types of networks and new protocols is good understanding of the already employed techniques. This goal seems to be best achieved by studying the protocol mathematical representation. However, due to the complexity of the Internet and the multitude of technologies involved, it is doubtful that a single framework could cover all the relevant aspects of data transfer in a consistent way. It is not surprising then that looking at the networking phenomena from different perspectives, the researchers in the past created quite distinct models describing the network dynamics. Among the outstanding proposals, one should certainly mention the utility-optimization approach of Kelly et al. [11, 12], studied in detail in the context of TCP/IP networks, for example, by Low et al. [15], the delay-based model of Mascolo [16], and the DiffServ framework, analyzed, for example, by Pitsillides et al. [18]. However, one of the most widely accepted models of the TCP dynamics was presented by Misra et al. in [17]. It was developed through a stochastic, fluid-flow analysis of packet traffic in the TCP networks with the emphasis placed on the dominant phase of the TCP data transfer – the congestion avoidance phase. In the conducted analysis, the authors took into account the key regulatory mechanisms provided by the TCP protocol and the network-assisted control enhancements implemented in the routers supporting AQM. Typically, in the investigation of the TCP dynamical characteristics, a simplified version of that model is applied, which ignores the TCP timeout mechanism.

According to [17], the mathematical description of the essential TCP dynamics (with the timeout effects neglected) can be represented by the pair of coupled, nonlinear differential equations:

$$\dot{W}(t) = \frac{1}{R(t)} - \beta W(t) \frac{W\,[t - R(t)]}{R\,[t - R(t)]} p\,[t - R(t)]\,, \tag{8.1}$$

$$\dot{q}(t) = \begin{cases} N(t)\frac{W(t)}{R(t)} - C(t), & q(t) > 0, \\ \max\left\{0, N(t)\frac{W(t)}{R(t)} - C(t)\right\}, & q(t) = 0, \end{cases} \tag{8.2}$$

where:

- $W(\cdot)$ is the expected window size (packets).
- $q(\cdot)$ is the expected packet queue length (packets); the notion of expected value in $W(\cdot)$ and $q(\cdot)$ refers to ensemble averaging.
- $C(\cdot)$ is the time-varying bandwidth utilized by the controlled flows (packets/s).
- $R(t) = T_p + q(t)/C(t)$ is the flow RTT which consists of two major terms – the propagation delay T_p that is usually assumed constant (the assumption is particularly well justified when the transfer route does not change during the exchange of data) and the time-varying queuing delay $q(t)/C(t)$.
- β is the window decrease parameter.
- $p(\cdot)$ is the probability of packet mark (or drop).
- $N(\cdot)$ is the load factor (the number of TCP connections contributing to the queue buildup in the output link buffer).

All the terms, $R(\cdot)$, $W(\cdot)$, $C(\cdot)$, $N(\cdot)$, $p(\cdot)$, and $q(\cdot)$, in (8.1) and (8.2) are varying functions of time. Moreover, $W(\cdot)$, $C(\cdot)$, $q(\cdot)$, and $p(\cdot)$ take only nonnegative, bounded values. The window size of a TCP source, $W(\cdot)$, is limited to the interval $[0, W_{\max}]$, where $W_{\max} > 0$ denotes the maximum window size, the packet queue length $q(\cdot) \in [0, B_{\text{size}}]$, where $B_{\text{size}} > 0$ is the link buffer capacity, and the packet marking probability is subject to the fundamental constraint $p(\cdot) \in [0, 1]$. The available bandwidth, $C(\cdot) \in [0, C_{\max}]$, constitutes the part of the overall link capacity $C_{\max}$ which is not consumed by the uncontrolled flows (e.g., inelastic traffic, or high priority transmission). The time-varying function $C(\cdot)$ also captures the link capacity fluctuations in the wireless environment (occurring, for instance, due to the fading and shadowing phenomena). Note that in the wireless networks, the channel bit rate (and thus the bandwidth available for data transfer) is adjusted according to the signal strength and bit error rate and typically undergoes changes even during a short-lived transmission.

The dynamics of the TCP flow control process represented by (8.1) and (8.2) is illustrated in a block diagram form in Fig. 8.1. The first equation in the analyzed set reflects the TCP window size adjustment according to the indications obtained from the network (the *TCP window control* structure in the graph shown in Fig. 8.1). The first term on the right-hand side in this equation, $1/R(\cdot)$, corresponds to the window *additive increase* part of the TCP mechanism, i.e., to increase the window size by one packet every RTT. The second term on the right-hand side in (8.1), $\beta W(\cdot)$, models the window *multiplicative decrease* mechanism. The window size is decreased by β for packets marked with probability $p(\cdot)$ arriving with intensity

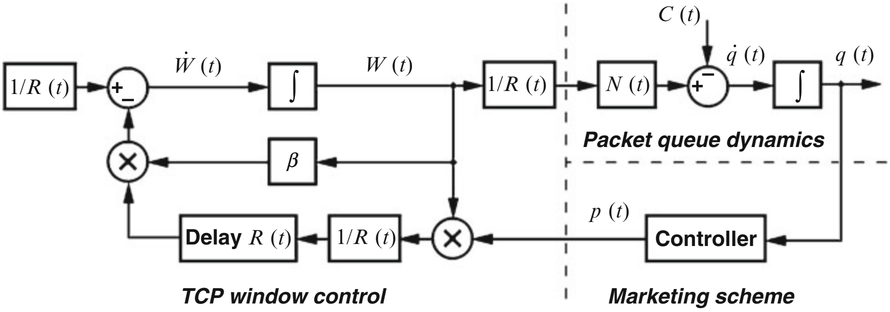


Fig. 8.1 TCP nonlinear dynamics

$W(\cdot)/R(\cdot)$. The window decrease parameter β is usually taken to be equal to 1/2, although as pointed out in [19], a more precise value for the continuous-time approximation of the TCP behavior suggests the value of ln(2) [13], or 2/3 [14]. The second equation in the considered set, (8.2), models the bottleneck queue length variations in the presence of the aggregated flow from $N(\cdot)$ connections and the time-varying bandwidth at the outgoing link. Consequently, the queue length is increased according to the incoming traffic intensity $W(\cdot)/R(\cdot)$ of $N(\cdot)$ flows, and it is decreased when the available bandwidth $C(\cdot) > 0$. For more insights on the TCP behavior described by (8.1) and (8.2), one can refer to [6, 7, 19].

In the next two sections, we investigate two distinct choices of small-signal linearization of the set of nonlinear equations (8.1) and (8.2) with the information about the delay retained. For each case, we discretize the linearized system dynamics and derive a discrete-time flow controller. The obtained controllers determine the packet marking rate at the network nodes supporting AQM.

8.2 System with Input Delay

By choosing the set of parameters subject to linearization, different approximations of system (8.1) and (8.2) can be obtained. The discussion provided so far in this monograph clearly indicates the benefits of explicit consideration of delay, which should be accounted for both in the system modeling and the controller design. Therefore, in contrast to the majority of earlier SM control proposals in the field (see Sect. 2.5 for a related discussion), while performing the linearization procedure, we intend to retain the information about the delay present in the system description. First, we analyze the system subject to the input delay only. We perform the linearization, discretize the obtained small-signal system model, and represent the discretized model in the state space. Next, we design a discrete-time controller to be implemented as a marking scheme in the router AQM procedures.

A more complex configuration with both the input and state delay information preserved is considered in Sect. 8.3. In either of these two cases, we do not assume the bandwidth to be constant and incorporate $C(\cdot)$ variations explicitly in the linearization procedure. We also explicitly account for the effects of finite sampling rate, which is unavoidable in the practical network realization. This is in contrast to the majority of feedback controller designs for TCP/AQM networks which assume either constant bandwidth, continuous feedback information delivery, or neglect the influence of network latency.

8.2.1 *Discrete-Time Network Model*

In this section, we develop a discrete-time model reflecting the TCP behavior in the neighborhood of an equilibrium point. First, the linearization procedure of the nonlinear system equations (8.1) and (8.2) is performed. Next, the linearized system is discretized, and the relevant state-space description is provided. The model developed in this section incorporates the information related to the input delay only. The discussion of a more sophisticated case of the input and state delay information preserved will be covered in Sect. 8.3.

8.2.1.1 Linearization

Let us consider $W(\cdot)$ and $q(\cdot)$ in (8.1) and (8.2) as the state variables. The system is driven by two inputs: $p(\cdot)$ – the controller command – and $C(\cdot)$ – the exogenous signal of unknown form treated as a disturbance. Function $q(\cdot)$ is selected as the system output. The operating point is specified by the quadruple (W_0, q_0, p_0, C_0) so that

$$\dot{W} = 0 \text{ and } \dot{q} = 0\Big|_{(W,q,p,C)=(W_0,q_0,p_0,C_0)}. \tag{8.3}$$

Hence, equating the left-hand sides of (8.1) and (8.2) to zero, one obtains at the operating point (W_0, q_0, p_0, C_0) the following set of relations:

$$\beta p_0 W_0^2 = 1, NW_0 = C_0 R_0, \text{ and } R_0 = T_p + q_0/C_0, \tag{8.4}$$

where R_0 is the equilibrium RTT. For the network parameters N and T_p, the set of feasible operating points is defined as

$$\begin{aligned}\Omega_{(N,T_p)} = \{&(W_0, q_0, p_0, C_0) : W_0 \in (0, W_{\max}), q_0 \in (0, B_{\text{size}}), \\ &p_0 \in (0,1), C_0 \in (0, C_{\max}) \text{ and (8.4) is satisfied}\}.\end{aligned} \tag{8.5}$$

We assume that RTT is constant at the equilibrium point and ignore the nested dependence of the time-delay argument $t-R(t)$ on the queue length and the available bandwidth (see [15] for the relevant discussion). Thus, we set $t-R(t)\equiv t-R_0$ in the linearization procedure. Taking N and T_p as constants, and neglecting the state delay (according to [6], this is a reasonable assumption when $W(\cdot)\gg 1$), we obtain the simplified system dynamics

$$\dot{W}(t)=\frac{1}{T_p+q(t)/C(t)}-\frac{\beta W^2(t)}{T_p+q(t)/C(t)}p\left(t-R_0\right), \tag{8.6}$$

$$\dot{q}(t)=\frac{NW(t)}{T_p+q(t)/C(t)}-C(t). \tag{8.7}$$

Let us analyze the small-signal deviations of the state and input variables from their operating point:

$$\delta W(t)\triangleq W(t)-W_0, \delta q(t)\triangleq q(t)-q_0, \delta p(t)\triangleq p(t)-p_0, \delta C(t)\triangleq C(t)-C_0. \tag{8.8}$$

Defining the right-hand sides of (8.6) and (8.7) by

$$f\left(W,q,p_R,C\right)\triangleq\frac{1}{T_p+q/C}-\frac{\beta W^2}{T_p+q/C}p_R, \tag{8.9}$$

$$g\left(W,q,C\right)\triangleq\frac{NW}{T_p+q/C}-C, \tag{8.10}$$

where $p_R(t)\triangleq p(t-R_0)$, and taking the partial derivatives of $f(\cdot)$ and $g(\cdot)$ at the operating point, we obtain

$$\begin{aligned}
\frac{\partial f}{\partial C} &= \frac{q}{C^2}\frac{1-\beta W^2 p_R}{\left(T_p+q/C\right)^2}\bigg|_{(W_0,q_0,p_0,C_0)} = 0,\\
\frac{\partial f}{\partial W} &= -\frac{2\beta W p_R}{T_p+q/C}\bigg|_{(W_0,q_0,p_0,C_0)} = -\frac{2}{R_0W_0} = -\frac{2N}{R_0^2C_0},\\
\frac{\partial f}{\partial p_R} &= -\frac{\beta W^2}{T_p+q/C}\bigg|_{(W_0,q_0,p_0,C_0)} = -\frac{\beta R_0 C_0^2}{N^2},\\
\frac{\partial f}{\partial q} &= \frac{\beta W^2 p_R-1}{C\left(T_p+q/C\right)^2}\bigg|_{(W_0,q_0,p_0,C_0)} = 0,
\end{aligned} \tag{8.11}$$

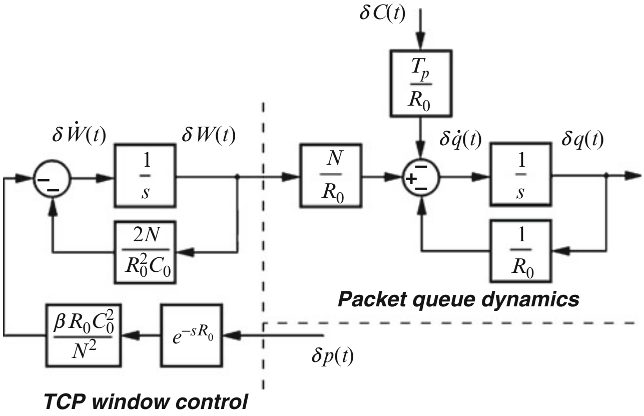


Fig. 8.2 Dynamics of deviation system with input delay

and

$$\begin{aligned}
\frac{\partial g}{\partial C} &= \frac{q}{C^2}\frac{NW}{\left(T_p+q/C\right)^2}-1\bigg|_{(W_0,q_0,p_0,C_0)} = \frac{q_0}{C_0R_0}-1=\frac{R_0-T_p}{R_0}-1=-\frac{T_p}{R_0},\\
\frac{\partial g}{\partial W} &= \frac{N}{T_p+q/C}\bigg|_{(W_0,q_0,p_0,C_0)} = \frac{N}{R_0},\\
\frac{\partial g}{\partial q} &= -\frac{NW}{C\left(T_p+q/C\right)^2}\bigg|_{(W_0,q_0,p_0,C_0)} = -\frac{1}{R_0}.
\end{aligned} \tag{8.12}$$

Consequently, the deviation dynamics can be expressed as

$$\delta \dot{W}(t) = -\frac{2N}{R_0^2C_0}\delta W(t) - \frac{\beta R_0 C_0^2}{N^2}\delta p\left(t-R_0\right), \tag{8.13}$$

$$\delta \dot{q}(t) = -\frac{1}{R_0}\delta q(t) + \frac{N}{R_0}\delta W(t) - \frac{T_p}{R_0}\delta C(t). \tag{8.14}$$

The deviation system dynamics is illustrated in a block diagram form in Fig. 8.2. Since R_0, C_0, and N are all positive quantities, the open-loop poles, $-2N/R_0^2C_0$ and $-1/R_0$, lie in the left-half plane, and the open-loop system is locally asymptotically stable. Further remarks on the stability of the considered system can be found in [6] and [7].

8.2.1.2 Discretization

Before proceeding with the discretization of system (8.13) and (8.14), let us first write it in the state-space form

$$\begin{aligned}\dot{\mathbf{x}}_{\mathbf{c}}(t) &= \mathbf{A}_{\mathbf{c}}\mathbf{x}_{\mathbf{c}}(t) + \mathbf{b}_{\mathbf{c}}\delta p\,(t - R_0) + \mathbf{d}_{\mathbf{c}}\delta C(t),\\ y(t) &= {\mathbf{q}_{\mathbf{c}}}^{\mathrm{T}}\mathbf{x}_{\mathbf{c}}(t),\end{aligned} \tag{8.15}$$

where $\mathbf{x}_{\mathbf{c}}(t) = [\delta q(t)\ \delta W(t)]^{\mathrm{T}}$ is the state vector, $\mathbf{A}_{\mathbf{c}}$ is 2×2 state matrix, and $\mathbf{b}_{\mathbf{c}}$, $\mathbf{d}_{\mathbf{c}}$, and $\mathbf{q}_{\mathbf{c}}$ are 2×1 vectors:

$$\mathbf{A}_{\mathbf{c}} = \begin{bmatrix} -\dfrac{1}{R_0} & \dfrac{N}{R_0} \\ 0 & -\dfrac{2N}{R_0^2 C_0} \end{bmatrix}, \mathbf{b}_{\mathbf{c}} = \begin{bmatrix} 0 \\ -\dfrac{\beta R_0 C_0^2}{N^2} \end{bmatrix}, \mathbf{d}_{\mathbf{c}} = \begin{bmatrix} -\dfrac{T_p}{R_0} \\ 0 \end{bmatrix}, \mathbf{q}_{\mathbf{c}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{8.16}$$

We assume that the system is sampled with period T, and R_0 constitutes an integer multiple of T, i.e., $R_0 = n_0 T$, where $n_0 > 0$. Using the inverse Laplace transform method, we obtain for $C_0 R_0 \neq 2N$ the following discrete-time form of the state matrix $\mathbf{A}_{\mathbf{d}} = \mathrm{e}^{\mathbf{A}_{\mathbf{c}}T} = \mathscr{L}^{-1}\{s\mathbf{I} - \mathbf{A}_{\mathbf{c}}\}|_{t=T}$ as

$$\mathbf{A}_{\mathbf{d}} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} e^{-\frac{T}{R_0}} & \dfrac{C_0 N R_0}{C_0 R_0 - 2N}\left(e^{-\frac{2NT}{R_0^2 C_0}} - e^{-\frac{T}{R_0}}\right) \\ 0 & e^{-\frac{2NT}{R_0^2 C_0}} \end{bmatrix}. \tag{8.17}$$

The discrete-time representation of the input and disturbance vectors is obtained from $\mathbf{b}_{\mathbf{d}} = \left(\int_0^T e^{\mathbf{A}_{\mathbf{c}}\tau}\mathrm{d}\tau\right)\mathbf{b}_{\mathbf{c}}$ and $\mathbf{d}_{\mathbf{d}} = \left(\int_0^T e^{\mathbf{A}_{\mathbf{c}}\tau}\mathrm{d}\tau\right)\mathbf{d}_{\mathbf{c}}$. Since $\mathbf{A}_{\mathbf{c}}$ is nonsingular, we may apply simplified formulas to calculate $\mathbf{b}_{\mathbf{d}}$ and $\mathbf{d}_{\mathbf{d}}$: $\mathbf{b}_{\mathbf{d}} = \mathbf{A}_{\mathbf{c}}^{-1}\left(e^{\mathbf{A}_{\mathbf{c}}T} - \mathbf{I}_2\right)\mathbf{b}_{\mathbf{c}}$ and $\mathbf{d}_{\mathbf{d}} = \mathbf{A}_{\mathbf{c}}^{-1}\left(e^{\mathbf{A}_{\mathbf{c}}T} - \mathbf{I}_2\right)\mathbf{d}_{\mathbf{c}}$, where $\mathbf{I}_2$ is 2×2 identity matrix. Thus, we get

$$\mathbf{b}_{\mathbf{d}} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \beta \begin{bmatrix} \dfrac{C_0^4 R_0^4}{2N^2(C_0 R_0 - 2N)} e^{-\frac{2NT}{R_0^2 C_0}} - \dfrac{C_0^3 R_0^3}{N(C_0 R_0 - 2N)} e^{-\frac{T}{R_0}} - \dfrac{C_0^3 R_0^3}{2N^2} \\ \dfrac{C_0^3 R_0^3}{2N^3} e^{-\frac{2NT}{R_0^2 C_0}} - \dfrac{C_0^3 R_0^3}{2N^3} \end{bmatrix}, \tag{8.18}$$

and

$$\mathbf{d_d} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} T_p \left[\exp\left(-T/R_0\right) - 1\right] \\ 0 \end{bmatrix}. \tag{8.19}$$

The output vector in discrete-time system $\mathbf{q_d} = \mathbf{q_c}$.

In the special case $C_0 R_0 = 2N$, we obtain

$$\mathbf{A_d} = \begin{bmatrix} e^{-\frac{T}{R_0}} & \frac{NT}{R_0} e^{-\frac{T}{R_0}} \\ 0 & e^{-\frac{T}{R_0}} \end{bmatrix}, \mathbf{b_d} = 4\beta N \begin{bmatrix} \left(1 + \frac{T}{R_0}\right) e^{-\frac{T}{R_0}} - 1 \\ e^{-\frac{T}{R_0}} - 1 \end{bmatrix}, \tag{8.20}$$

$\mathbf{d_d}$ given by (8.19), and $\mathbf{q_d} = \mathbf{q_c}$.

8.2.1.3 State-Space Representation

In order to explicitly consider the influence of input delay, for convenience of the controller derivation, the discrete-time system (8.17)–(8.19) is represented in the extended state space

$$\begin{aligned} \mathbf{x}\left[(k+1)T\right] &= \mathbf{A}\mathbf{x}(kT) + \mathbf{b}u(kT) + \mathbf{d}d(kT), \\ y(kT) &= \mathbf{q}^{\mathrm{T}}\mathbf{x}(kT), \end{aligned} \tag{8.21}$$

where:

- $u(kT) = \delta p(kT)$ denotes the controller command, i.e., the deviation of the packet marking probability from the equilibrium value p_0 at instant kT.
- $d(kT) = \delta C(kT)$ represents the deviation of the available bandwidth from its equilibrium value C_0 at instant kT.
- $\mathbf{x}(kT) = [x_1(kT)\ x_2(kT) \ldots\ x_n(kT)]^{\mathrm{T}}$ is the state vector with:
  - $x_1(kT) = y(kT) = \delta q(kT)$ representing the difference between the packet queue length and its equilibrium value q_0 at instant kT
  - $x_2(kT) = \delta W(kT)$ the current difference between the window size and its equilibrium value W_0
  - The remaining state variables

$$x_i(kT) = \delta p[k - n + i - 1)T] \tag{8.22}$$

  for $i = 3, \ldots, n$ representing the controller command history.

- $\mathbf{A}$ is $n \times n$ state matrix, $\mathbf{b}$, $\mathbf{d}$, and $\mathbf{q}$ are $n \times 1$ input, disturbance, and output vectors, respectively,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & b_1 & 0 & \dots & 0 \\ 0 & a_{22} & b_2 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \mathbf{d} = \begin{bmatrix} d_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \mathbf{q} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \tag{8.23}$$

and the system order $n = n_0 + 2 = (R_0/T) + 2$ depends on the discretization period and RTT of the flows. Note that since $R_0 > 0$, the system is at least of order 3. The coefficients a_{11}, a_{12}, a_{22}, b_1, and b_2 are taken from (8.17), and (8.18) if $C_0 R_0 \neq 2N$, and from (8.20) otherwise. The constant d_1 is determined from (8.19).

The deviation dynamics should be kept in the vicinity of the equilibrium point. Hence, the desired state vector in the analyzed case is the origin of the deviation state space

$$\mathbf{x_d} = \begin{bmatrix} 0\ 0\ 0 \dots 0 \end{bmatrix}^{\mathrm{T}}.$$

8.2.2 *Flow Control Strategy*

The purpose of the control action can be formulated as to reach a desired operating point (for instance, to drive the queue length to a set-point level) and to maintain the system dynamics near the given operating point despite modeling inaccuracies, parameter variations, and external disturbances. In particular, in the presence of bandwidth (or link capacity) variations, and the fluctuating number of connections, the instantaneous queue length drift from the equilibrium value should not cause buffer overflow. On the other hand, a bandwidth surge or load change should not lead to depleting the buffer, as it usually implies decreased utilization of the network resources.

8.2.2.1 Sliding-Mode Controller Design

Let us introduce the sliding hyperplane described by the following equation:

$$s\,(kT) = \mathbf{c}^{\mathrm{T}} \left[\mathbf{x_d} - \mathbf{x}\,(kT)\right] = -\mathbf{c}^{\mathrm{T}} \mathbf{x}\,(kT) = 0, \tag{8.24}$$

where for the feasibility purposes the vector describing the plane parameters,

$$\mathbf{c}^{\mathrm{T}} = \begin{bmatrix} c_1\ c_2\ c_3 \dots c_{n-1}\ c_n \end{bmatrix}, \tag{8.25}$$

should be chosen such that $\mathbf{c}^{\mathrm{T}}\mathbf{b} \neq 0$. The selection of this vector will be analyzed further in this section. Substituting (8.21) into the equation $\mathbf{c}^{\mathrm{T}}\mathbf{x}[(k+1)T] = 0$ (with $d(\cdot) \equiv 0$), we get

$$\mathbf{c}^{\mathrm{T}}\mathbf{x}\left[(k+1)T\right] = \mathbf{c}^{\mathrm{T}}\left[\mathbf{A}\mathbf{x}(kT) + \mathbf{b}u(kT)\right] = 0, \tag{8.26}$$

which leads to the following feedback control law:

$$u(kT) = -\left(\mathbf{c}^{\mathrm{T}}\mathbf{b}\right)^{-1}\mathbf{c}^{\mathrm{T}}\mathbf{A}\mathbf{x}(kT). \tag{8.27}$$

Applying (8.23), we can rewrite (8.27) in the following form

$$\begin{aligned} u(kT) = & -c_n{}^{-1}\left[a_{11}c_1x_1(kT) + (a_{12}c_1 + a_{22}c_2)\,x_2(kT)\right] \\ & - c_n{}^{-1}\left[(b_1c_1 + b_2c_2)\,x_3(kT) + \sum_{j=4}^{n} c_{j-1}x_j(kT)\right]. \end{aligned} \tag{8.28}$$

Since the properties of SM controllers are mainly determined by an appropriate choice of the sliding plane parameters, we devote the rest of this section to the selection of a suitable vector **c** for controller (8.28).

8.2.2.2 Dead-Beat Controller

As discussed in the previous chapters, a good control scheme for data transmission networks should provide appropriately fast responsiveness to the changes of networking conditions. Therefore, parameters of the sliding plane will be selected so that a dead-beat controller for the deviation dynamics (8.21)–(8.23) is obtained.

For dead-beat control, all the poles of the closed-loop state matrix should be at the origin. The closed-loop state matrix $\mathbf{A}_{\mathbf{cl}} = [\mathbf{I}_n - \mathbf{b}(\mathbf{c}^{\mathrm{T}}\mathbf{b})^{-1}\mathbf{c}^{T}]\mathbf{A}$ with control (8.28) applied is determined as

$$\mathbf{A}_{\mathbf{cl}} = \begin{bmatrix} a_{11} & a_{12} & b_1 & 0 & \dots & 0 \\ 0 & a_{22} & b_2 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ -\dfrac{a_{11}c_1}{c_n} & -\dfrac{a_{12}c_1 + a_{22}c_2}{c_n} & -\dfrac{b_1c_1 + b_2c_2}{c_n} & -\dfrac{c_3}{c_n} & \dots & -\dfrac{c_{n-1}}{c_n} \end{bmatrix}, \tag{8.29}$$

and its characteristic polynomial $\det(z\mathbf{I}_n - \mathbf{A_{cl}})$ as

$$z^n + \frac{c_{n-1} - (a_{11} + a_{22})\, c_n}{c_n} z^{n-1} + \frac{a_{11}\,(a_{22}c_n - c_{n-1}) - a_{22}c_{n-1} + c_{n-2}}{c_n} z^{n-2} + \cdots$$
$$\cdots + \frac{a_{11}\,(a_{22}c_5 - c_4) - a_{22}c_4 + c_3}{c_n} z^3 + \frac{a_{11}\,(a_{22}c_4 - c_3) - a_{22}c_3 + b_1c_1 + b_2c_2}{c_n} z^2$$
$$+ \frac{a_{11}\,(a_{22}c_3 - b_2c_2) + a_{12}b_2c_1 - a_{22}b_1c_1}{c_n} z. \tag{8.30}$$

Therefore, the roots of the characteristic equation $\det(z\mathbf{I}_n - \mathbf{A_{cl}}) = 0$ (the closed-loop poles) are all zero, if the following set of conditions is satisfied:

$$\begin{aligned}
&c_{n-1} - (a_{11} + a_{22})\, c_n = 0, \\
&a_{11}\,(a_{22}c_n - c_{n-1}) - a_{22}c_{n-1} + c_{n-2} = 0, \\
&\qquad\qquad\vdots \\
&a_{11}\,(a_{22}c_5 - c_4) - a_{22}c_4 + c_3 = 0, \\
&a_{11}\,(a_{22}c_4 - c_3) - a_{22}c_3 + b_1c_1 + b_2c_2 = 0, \\
&a_{11}\,(a_{22}c_3 - b_2c_2) + a_{12}b_2c_1 - a_{22}b_1c_1 = 0.
\end{aligned} \tag{8.31}$$

Equation set (8.31) is solved recursively. First, we determine c_{n-1} from the first equation in (8.31):

$$c_{n-1} = (a_{11} + a_{22})\, c_n. \tag{8.32}$$

Next, we substitute c_{n-1} given by (8.32) into the second equation in (8.31). In this way, c_{n-2} may be expressed in terms of the system parameters and the last element of vector $\mathbf{c}$. We get

$$c_{n-2} = c_n \left(a_{11}^2 + a_{11}a_{22} + a_{22}^2\right). \tag{8.33}$$

Having determined c_{n-1} and c_{n-2}, we solve for c_{n-3}, obtaining

$$c_{n-3} = c_n \left(a_{11}^3 + a_{11}^2 a_{22} + a_{11}a_{22}^2 + a_{22}^3\right) = c_n \sum_{j=0}^{3} a_{11}^{3-j} a_{22}^{j}. \tag{8.34}$$

If one continues the substitutions, the following relation may be observed:

$$\begin{aligned}
c_{n-j} &= c_n \left(a_{11}^j + a_{11}^{j-1}a_{22} + \cdots + a_{11}a_{22}^2 + a_{22}^j\right) \\
&= c_n \sum_{i=0}^{j} a_{11}^{j-i} a_{22}^{i}, \quad \text{for } j = 1, \ldots, n-3.
\end{aligned} \tag{8.35}$$

And the last two elements of vector **c** are determined as

$$\begin{aligned} c_2 &= \left(c_n \sum_{i=0}^{n-2} a_{11}^{n-2-i} a_{22}^{i} - b_1 c_1 \right) / b_2, \\ c_1 &= c_n a_{11}^{n-1} / \left(a_{11} b_1 + a_{12} b_2 - a_{22} b_1 \right). \end{aligned} \tag{8.36}$$

Substituting (8.35) and (8.36) into (8.28), we obtain

$$\begin{aligned} & c_n{}^{-1} \left[\left(b_1 c_1 + b_2 c_2 \right) x_3 (kT) + \sum_{j=4}^{n} c_{j-1} x_j (kT) \right] \\ &= \left[\left(b_1 c_1 + c_n \sum_{i=0}^{n-2} a_{11}^{n-2-i} a_{22}^{i} - b_1 c_1 \right) x_3 (kT) + \sum_{j=4}^{n} \left(\sum_{i=0}^{n-j+1} a_{11}^{n-j+1-i} a_{22}^{i} \right) x_j (kT) \right] \\ &= \sum_{j=3}^{n} \left(\sum_{i=0}^{n-j+1} a_{11}^{n-j+1-i} a_{22}^{i} \right) x_j (kT), \end{aligned} \tag{8.37}$$

and after applying (8.22), the sum reduces to

$$\sum_{j=3}^{n} \left(\sum_{i=0}^{n-j+1} a_{11}^{n-j+1-i} a_{22}^{i} \right) \delta p \left[(k-n+j-1) T \right] = \sum_{j=k-n_0}^{k-1} \left(\sum_{i=0}^{k-j} a_{11}^{k-j-i} a_{22}^{i} \right) \delta p (jT). \tag{8.38}$$

Consequently, using $x_1(kT) = \delta q(kT)$, $x_2(kT) = \delta W(kT)$, and (8.38) in (8.28), we get the following closed-form expression for dead-beat control law for the considered system:

$$\begin{aligned} u(kT) = \delta p(kT) = -g_1 \delta q(kT) - g_2 \delta W(kT) \\ - \sum_{j=k-n_0}^{k-1} \left(\sum_{i=0}^{k-j} a_{11}^{k-j-i} a_{22}^{i} \right) \delta p (jT), \end{aligned} \tag{8.39}$$

where

$$g_1 = \frac{a_{11}^{n}}{a_{11} b_1 + a_{12} b_2 - a_{22} b_1}, g_2 = \frac{\left(a_{12} b_2 - a_{22} b_1 \right) a_{11}^{n-1}}{b_2 \left(a_{11} b_1 + a_{12} b_2 - a_{22} b_1 \right)} + \frac{a_{22}}{b_2} \sum_{i=0}^{n-2} a_{11}^{n-2-i} a_{22}^{i}. \tag{8.40}$$

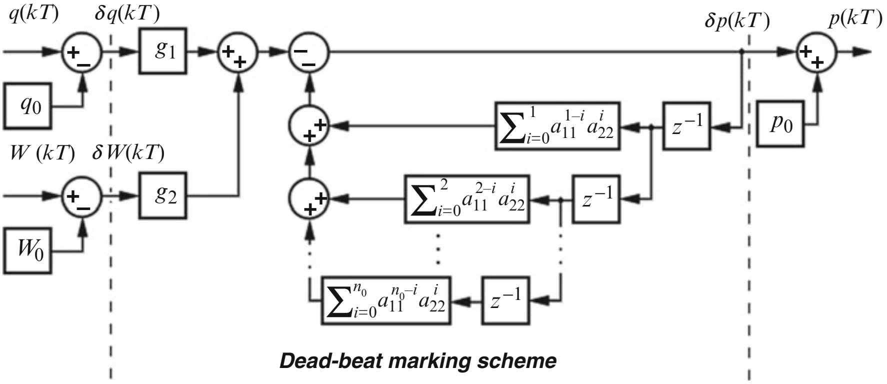


Fig. 8.3 Dead-beat controller for system with input delay

Equation (8.39) represents a fixed-gain variable-state feedback control law. In order to determine the current packet marking rate $\delta p(kT)$, the controller uses the measurement of the instantaneous queue length $\delta q(kT)$, the window size estimate (obtained from observing the packet incoming rate) $\delta W(kT)$, and the marking rate history recorded within the last RTT. Since all the controller parameters can be computed off-line, good operational efficiency in the node implementation is ensured. The structure of the designed dead-beat controller is illustrated in Fig. 8.3.

8.2.2.3 Modified Control Law

The developed dead-beat scheme provides fast reaction to the changes of networking conditions. However, when the system departs from the desired operating point, for instance, due to bandwidth variations, the controller performance degrades. This is mainly attributed to the changes in the round-trip time $R(t) = T_p + q(t)/C(t)$, which in the case of large $C(\cdot)$ variations may significantly depart from the operating-point value R_0. As analyzed in Sects. 5.2 and 6.2, when the delay differs from the nominal value taken into account in the design procedure, the dead-beat controller performance degrades, leading even to instability. Also, in the case of the system analyzed in this chapter, the modeling inaccuracy and high-frequency parasitics will adversely influence the operation of dead-beat control law in the actual network implementation. Nevertheless, as it was indicated in Sect. 5.2.2, the controller robustness can be enhanced by introducing a scaling factor in the sliding plane. By incorporating the factor $\gamma \in [0, 1]$ into the sliding hyperplane (8.25), we obtain the vector of plane parameters

$$\mathbf{c}^{\mathrm{T}} = \begin{bmatrix} \gamma c_1 & \gamma c_2 & \gamma c_3 \ldots & \gamma c_{n-1} & c_n \end{bmatrix}, \tag{8.41}$$

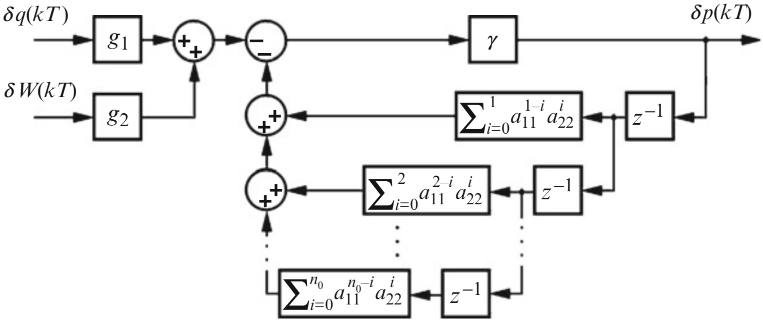

$\delta q(kT)$
$\delta W(kT)$
g_1
g_2
γ
$\delta p(kT)$
$\sum_{i=0}^{1} a_{11}^{1-i} a_{22}^{i}$
$\sum_{i=0}^{2} a_{11}^{2-i} a_{22}^{i}$
$\sum_{i=0}^{n_0} a_{11}^{n_0-i} a_{22}^{i}$
z^{-1}
z^{-1}
z^{-1}

Fig. 8.4 Proportional controller for system with input delay

where $c_1, c_2, \ldots, c_{n-1}$ are given by (8.35) and (8.36). Using (8.41) in (8.27), we arrive at the proportional control law

$$\delta p\,(kT) = -\gamma\left[g_1\delta q\,(kT) + g_2\delta W\,(kT) + \sum_{j=k-n_0}^{k-1}\left(\sum_{i=0}^{k-j} a_{11}^{k-j-i}a_{22}^{i}\right)\delta p\,(jT)\right] \tag{8.42}$$

with g_1 and g_2 given by (8.40). The new controller eliminates error at the output slower than the dead-beat scheme, but it is less fragile to inconsistencies in the system modeling and changes in the operating conditions.

The block diagram of the proportional controller is illustrated in Fig. 8.4. As we can see from the graph, the dead-beat structure is modified by adding a gain element to the primary signal path. The remaining elements and their configuration remain the same as depicted in Fig. 8.3. In particular, the structure responsible for delay compensation is preserved.

8.3 System with Input and State Delay

In the linearization procedure conducted in the previous section, an explicit consideration was given to the latency in the control signal path – the input delay influencing the effect of packet marking rate $p[t-R(t)]$. The state delay affecting the changes in the window size $W[t-R(t)]$ and the queue length $q[t-R(t)]$ was neglected. In this section, we extend the ideas presented so far in this chapter to the system with the information about both the input and state delay retained in the system model. The linearization procedure leads to more complex coupled differential equations in the model description, which call for more sophisticated controller design procedures. We show, however, that with an appropriate selection of the discrete-time state space, the controller derivation yields a similar structure as obtained in Sect. 8.2.

8.3.1 Discrete-Time Network Model

We begin with linearizing the system equations (8.1) and (8.2), which represent the fundamental TCP dynamics (in the congestion avoidance phase). Contrary to the procedure presented in the previous section, here the information about the state delay in addition to the input latency is retained.

8.3.1.1 Linearization

We assume that RTT is constant at the equilibrium point and ignore the dependence of the time-delay argument $t - R(t)$ on the queue length, setting $t - R(t) \equiv t - R_0$ in the linearization procedure. Taking N and T_p as constants, and explicitly considering the delay in $W(\cdot)$, $q(\cdot)$, and $C(\cdot)$, we obtain the system dynamics in the following form:

$$\dot{W}(t) = \frac{1}{T_p + q(t)/C(t)} - \frac{\beta W(t) W(t - R_0)}{T_p + q(t - R_0)/C(t - R_0)} p(t - R_0), \tag{8.43}$$

$$\dot{q}(t) = \frac{N W(t)}{T_p + q(t)/C(t)} - C(t). \tag{8.44}$$

Next, we analyze small-signal deviations (8.8) of the state and input variables from their operating point values. Defining the right-hand sides of (8.43) and (8.44) by

$$f(W, W_R, q, q_R, p_R, C, C_R) \triangleq \frac{1}{T_p + q/C} - \frac{\beta W W_R}{T_p + q_R/C_R} p_R, \tag{8.45}$$

$$g(W, q, C) \triangleq \frac{N W}{T_p + q/C} - C, \tag{8.46}$$

where

$$W_R(t) \triangleq W(t - R_0), q_R(t) \triangleq q(t - R_0), p_R(t) \triangleq p(t - R_0), C_R(t) \triangleq C(t - R_0), \tag{8.47}$$

and taking the partial derivatives at the operating point, we obtain

$$
\begin{aligned}
\frac{\partial f}{\partial C} &= \frac{q}{C^2}\frac{1}{\left(T_p+q/C\right)^2}\bigg|_{\left(W_0,q_0,p_0,C_0\right)} = \frac{C_0\left(R_0-T_p\right)}{C_0^2R_0^2} = \frac{R_0-T_p}{R_0^2C_0},\\
\frac{\partial f}{\partial C_R} &= -\frac{\beta W W_R p_R q_R}{C_R^2\left(T_p+q_R/C_R\right)^2}\bigg|_{\left(W_0,q_0,p_0,C_0\right)} = -\frac{q_0}{C_0^2}\frac{1}{\left(T_p+q_0/C_0\right)^2} = -\frac{R_0-T_p}{R_0^2C_0},\\
\frac{\partial f}{\partial W} &= -\frac{\beta W_R p_R}{q_R/C_R+T_p}\bigg|_{\left(C_0,W_0,p_0,q_0\right)} = -\frac{\beta W_0}{\left(q_0/C_0+T_p\right)}\frac{1}{\beta W_0^2} = -\frac{1}{R_0W_0} = -\frac{N}{R_0^2C_0},\\
\frac{\partial f}{\partial W_R} &= \frac{\partial f}{\partial W},\\
\frac{\partial f}{\partial p_R} &= -\frac{\beta W W_R}{q_R/C_R+T_p}\bigg|_{\left(C_0,W_0,p_0,q_0\right)} = -\frac{\beta W_0^2}{R_0} = -\frac{\beta R_0^2C_0^2/N^2}{R_0} = -\frac{\beta R_0C_0^2}{N^2},\\
\frac{\partial f}{\partial q} &= -\frac{1}{C\left(q/C+T_p\right)^2}\bigg|_{\left(C_0,W_0,p_0,q_0\right)} = -\frac{1}{C_0R_0^2},\\
\frac{\partial f}{\partial q_R} &= \frac{\beta W W_R p_R}{C_R\left(q_R/C_R+T_p\right)^2}\bigg|_{\left(C_0,W_0,p_0,q_0\right)} = \frac{1}{C_0R_0^2},
\end{aligned}
\tag{8.48}
$$

and

$$
\begin{aligned}
\frac{\partial g}{\partial C} &= \frac{q}{C^2}\frac{NW}{\left(T_p+q/C\right)^2}-1\bigg|_{(W_0,q_0,p_0,C_0)} = \frac{C_0\left(R_0-T_p\right)C_0R_0}{C_0^2R_0^2}-1 = -\frac{T_p}{R_0},\\
\frac{\partial g}{\partial W} &= \frac{N}{T_p+q/C}\bigg|_{(W_0,q_0,p_0,C_0)} = \frac{N}{R_0},\\
\frac{\partial g}{\partial q} &= -\frac{NW}{C\left(T_p+q/C\right)^2}\bigg|_{(W_0,q_0,p_0,C_0)} = -\frac{C_0R_0}{C_0R_0^2} = -\frac{1}{R_0}.
\end{aligned}
\tag{8.49}
$$

Consequently, the deviation dynamics can be expressed by

$$
\begin{aligned}
\delta\dot{W}(t) = &-\frac{N}{R_0^2C_0}\left[\delta W(t)+\delta W\left(t-R_0\right)\right]-\frac{1}{R_0^2C_0}\left[\delta q(t)-\delta q\left(t-R_0\right)\right]\\
&-\frac{\beta R_0C_0^2}{N^2}\delta p\left(t-R_0\right)+\frac{R_0-T_p}{R_0^2C_0}\left[\delta C(t)-\delta C\left(t-R_0\right)\right],
\end{aligned}
\tag{8.50}
$$

$$
\delta\dot{q}(t) = -\frac{1}{R_0}\delta q(t)+\frac{N}{R_0}\delta W(t)-\frac{T_p}{R_0}\delta C(t). \tag{8.51}
$$

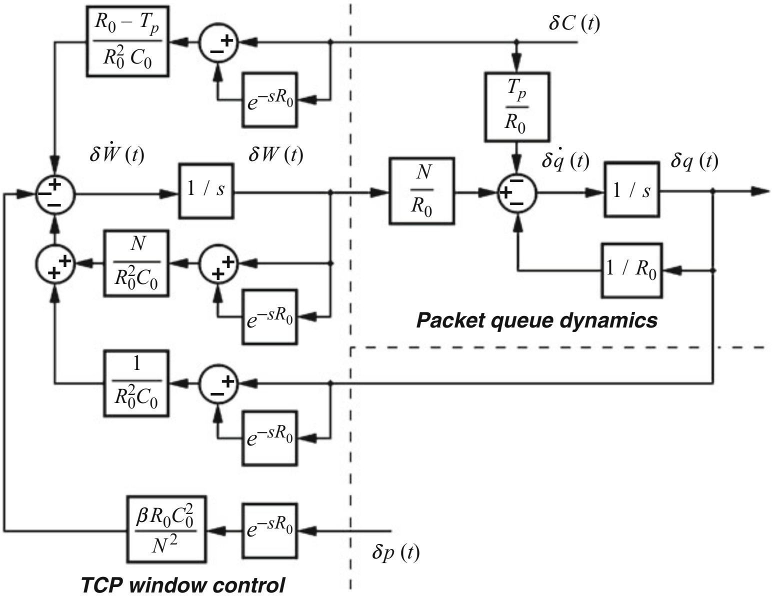


Fig. 8.5 Dynamics of deviation system with input and state delay

The deviation system dynamics in a block diagram form is illustrated in Fig. 8.5. The open-loop stability, taking into account the high-frequency parasitics and coupling effects, can be analyzed following the ideas presented in [7].

8.3.1.2 Discretization

In order to obtain a convenient state-space description, we need to use a first-order approximation in the discretization procedure. Assuming sufficiently small sampling period T, and RTT being a multiple of T, i.e., $R_0 = n_0 T$, where n_0 is a positive integer, we obtain the following discrete-time representation of (8.50):

$$\begin{aligned}\delta W[(k+1)T] = {} & \left(1 - \frac{NT}{R_0^2 C_0}\right)\delta W(kT) - \frac{NT}{R_0^2 C_0}\delta W[(k-n_0)T] \\ & - \frac{T}{R_0^2 C_0}\delta q(kT) + \frac{T}{R_0^2 C_0}\delta q[(k-n_0)T] - \frac{\beta R_0 C_0^2 T}{N^2}\delta p[(k-n_0)T] \\ & + \frac{(R_0 - T_p)T}{R_0^2 C_0}\delta C(kT) - \frac{(R_0 - T_p)T}{R_0^2 C_0}\delta C[(k-n_0)T]. \qquad (8.52)\end{aligned}$$

Similarly, the first-order discrete-time approximation of (8.51) is determined as

$$\delta q\left[(k+1)T\right] = \left(1 - \frac{T}{R_0}\right)\delta q\,(kT) + \frac{NT}{R_0}\delta W\,(kT) - \frac{T_pT}{R_0}\delta C\,(kT)\,. \quad (8.53)$$

8.3.1.3 State-Space Representation

In order to efficiently conduct the controller design procedure using formal approach, the discrete-time system (8.52) and (8.53) is represented in the state space. Similarly as in the previous section, the choice of the state variables reflects the influence of delay on the system dynamics. Here, the particular selection of elements of the state vector and their interactions explicitly accounts for the effects of latency in the input channel (δp, δC) and in the system state (δq, δW). The overall state equations are given by (8.21), where:

- $\mathbf{x}(kT) = [x_1(kT)\; x_2(kT) \ldots\; x_n(kT)]^{\mathrm{T}}$ is the state vector with:
  - $x_1(kT) = y(kT) = \delta q(kT)$ representing the difference between the packet queue length and its equilibrium value q_0 at instant kT,
  - $x_2(kT) = \delta W(kT)$ the current difference between the window size and its equilibrium value W_0,
  - the remaining state variables

$$\begin{aligned} x_i\,(kT) = {} & a_{n1}\delta q\left[(k-n+i-1)\,T\right] + a_{n2}\delta W\left[(k-n+i-1)\,T\right] \\ & + b_n\delta p\left[(k-n+i-1)\,T\right] + d_n\delta C\left[(k-n+i-1)\,T\right] \quad (8.54) \end{aligned}$$

  for $i = 3, \ldots, n$ representing the state and input signal history.

- $\mathbf{A}$ is $n \times n$ state matrix, $\mathbf{b}$, $\mathbf{d}$, and $\mathbf{q}$ are $n \times 1$ input, disturbance, and output vectors,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & \ldots & 0 \\ a_{21} & a_{22} & 1 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 1 \\ a_{n1} & a_{n2} & 0 & 0 & \ldots & 0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ b_n \end{bmatrix}, \mathbf{d} = \begin{bmatrix} d_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ d_n \end{bmatrix}, \mathbf{q} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad (8.55)$$

  where

$$\begin{aligned} a_{11} &= 1 - T/R_0, a_{12} = NT/R_0, \\ a_{21} &= -T/(R_0^2C_0), a_{22} = 1 - NT/(R_0^2C_0), \\ a_{n1} &= T/(R_0^2C_0), a_{n2} = -NT/(R_0^2C_0), \\ b_n &= -\beta R_0C_0^2T/N^2, \\ d_1 &= -T_pT/R_0, d_n = -\left(R_0 - T_p\right)T/\left(R_0^2C_0\right), \quad (8.56) \end{aligned}$$

and the system order $n = n_0 + 2 = (R_0/T) + 2$ depends on the discretization period T and the nominal RTT of the flows R_0. Since $R_0 > 0$, the system is at least of order 3. The desired system state is defined as $\mathbf{x_d} = [0\ 0\ 0 \ldots\ 0]^{\mathrm{T}}$ so that the deviation dynamics is kept in the vicinity of the equilibrium point.

8.3.2 *Flow Control Strategy*

We wish to get a controller which will drive the system to a desired operating point, and maintain the state in the vicinity of that point despite modeling inaccuracies and the presence of disturbances (bandwidth and load factor fluctuations). The controller should quickly react to the changes of networking conditions and avoid overshoots and oscillations if possible. The design procedure follows similar steps as discussed in Sect. 8.2.2. First, the sliding hyperplane is introduced, and a general form of the control law is provided. Then, the closed-loop characteristic polynomial is analyzed, and parameters of the plane are selected so that all the closed-loop poles are at the origin. In this way, a dead-beat control law is obtained. Finally, the designed dead-beat controller is modified to improve robustness with respect to parametric uncertainties and external perturbations.

8.3.2.1 Sliding-Mode Controller Design

We introduce a sliding hyperplane described by the equation $s(kT) = -\mathbf{c}^{\mathrm{T}}\mathbf{x}(kT) = 0$, where the vector describing the plane parameters (8.25) is chosen such that $\mathbf{c}^{\mathrm{T}}\mathbf{b} \neq 0$. Substituting (8.21) into the equation $\mathbf{c}^{\mathrm{T}}\mathbf{x}[(k+1)T] = 0$, we arrive at

$$u(kT) = -\left(\mathbf{c}^{\mathrm{T}}\mathbf{b}\right)^{-1}\mathbf{c}^{\mathrm{T}}\mathbf{A}\mathbf{x}(kT)\,.$$

Applying (8.55), we can rewrite this equation in the following form:

$$\begin{aligned} u(kT) &= -(b_n c_n)^{-1} \\ &\times \left[(a_{11}c_1 + a_{21}c_2 + a_{n1}c_n)\,x_1(kT) + (a_{12}c_1 + a_{22}c_2 + a_{n2}c_n)\,x_2(kT) + \sum_{j=3}^{n} c_{j-1}x_j(kT)\right]. \end{aligned} \tag{8.57}$$

Next, we show how parameters of the sliding plane should be selected so that a dead-beat controller is obtained.

8.3.2.2 Dead-Beat Controller

Dead-beat control requires all the poles of the closed-loop state matrix to be placed at the origin. The closed-loop state matrix $\mathbf{A_{cl}} = [\mathbf{I}_n - \mathbf{b}(\mathbf{c}^T\mathbf{b})^{-1}\mathbf{c}^\mathrm{T}]\mathbf{A}$ with control (8.57) applied is determined as

$$\mathbf{A_{cl}} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & \dots & 0 \\ a_{21} & a_{22} & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ -\frac{a_{11}c_1 + a_{21}c_2}{c_n} & -\frac{a_{12}c_1 + a_{22}c_2}{c_n} & -\frac{c_2}{c_n} & -\frac{c_3}{c_n} & \dots & -\frac{c_{n-1}}{c_n} \end{bmatrix},$$

and its characteristic polynomial $\det(z\mathbf{I}_n - \mathbf{A_{cl}})$ as

$$z^n + \frac{c_{n-1} - (a_{11} + a_{22})\,c_n}{c_n} z^{n-1} + \frac{a_{11}\,(a_{22}c_n - c_{n-1}) - a_{12}a_{21}c_n - a_{22}c_{n-1} + c_{n-2}}{c_n} z^{n-2} + \cdots$$
$$\cdots + \frac{a_{11}\,(a_{22}c_4 - c_3) - a_{12}a_{21}c_4 - a_{22}c_3 + c_2}{c_n} z^2 + \frac{a_{11}\,(a_{22}c_3 - c_2) + a_{12}\,(c_1 - a_{21}c_3)}{c_n} z.$$

Denoting the first principal minor of $\mathbf{A}$ by

$$M_1 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}, \tag{8.58}$$

the characteristic polynomial may be presented in a more compact form as

$$z^n + \frac{c_{n-1} - (a_{11} + a_{22})\,c_n}{c_n} z^{n-1} + \frac{M_1c_n - (a_{11} + a_{22})\,c_{n-1} + c_{n-2}}{c_n} z^{n-2} + \cdots$$
$$\cdots + \frac{M_1c_4 - (a_{11} + a_{22})\,c_3 + c_2}{c_n} z^2 + \frac{M_1c_3 - a_{11}c_2 + a_{12}c_1}{c_n} z. \tag{8.59}$$

For dead-beat control, $\det(z\mathbf{I}_n - \mathbf{A_c})$ should be equal to z^n, which is satisfied when the following set of conditions is simultaneously satisfied:

$$\begin{aligned} &c_{n-1} - (a_{11} + a_{22})\,c_n = 0, \\ &M_1c_n - (a_{11} + a_{22})\,c_{n-1} + c_{n-2} = 0, \\ &\qquad\vdots \\ &M_1c_4 - (a_{11} + a_{22})\,c_3 + c_2 = 0, \\ &M_1c_3 - a_{11}c_2 + a_{12}c_1 = 0. \end{aligned} \tag{8.60}$$

Set (8.60) is solved by recursion. First, we substitute $c_{n-1} = (a_{11} + a_{22})c_n$ in the second equation in the set, and obtain c_{n-2} as a function of c_n. Then, we use the outcome of this operation to determine c_{n-3} from the third equation. Continuing with the substitutions, the following result is obtained:

$$\begin{aligned}
c_{n-1} &= (a_{11} + a_{22})\,c_n,\\
c_{n-2} &= c_n\left[-M_1 + (a_{11} + a_{22})^2\right],\\
c_{n-3} &= c_n\,(a_{11} + a_{22})\left[-2M_1 + (a_{11} + a_{22})^2\right],\\
c_{n-4} &= c_n\left[M_1^2 - 3M_1(a_{11} + a_{22})^2 + (a_{11} + a_{22})^4\right],\\
c_{n-5} &= c_n\,(a_{11} + a_{22})\left[3M_1^2 - 4M_1(a_{11} + a_{22})^2 + (a_{11} + a_{22})^4\right],\\
c_{n-6} &= c_n\left[-M_1^3 + 6M_1^2(a_{11} + a_{22})^2 - 5M_1(a_{11} + a_{22})^4 + (a_{11} + a_{22})^6\right],\\
&\vdots
\end{aligned}\tag{8.61}$$

Note that the coefficients in the powers of M_1 in subsequent rows in (8.61) form the diagonal elements of Pascal's triangle. Hence, for $j = 2, \ldots, n-1$, we may write

$$c_{n-j} = \begin{cases} c_n\,(a_{11} + a_{22}) \sum\limits_{i=0}^{p=(j-1)/2} \begin{pmatrix} j-p+i \\ j-p-1-i \end{pmatrix} (-M_1)^{p-i}(a_{11} + a_{22})^{2i}, & \text{for} \quad c_{n-1}, c_{n-3}, \ldots \\ c_n \sum\limits_{i=0}^{p=j/2} \begin{pmatrix} j-p+i \\ j-p-i \end{pmatrix} (-M_1)^{p-i}(a_{11} + a_{22})^{2i}, & \text{for} \quad c_{n-2}, c_{n-4}, \ldots \end{cases}\tag{8.62}$$

Finally, c_1 is determined from the last equation in set (8.60) as

$$c_1 = (a_{11}c_2 - M_1c_3)\,/a_{12}.\tag{8.63}$$

Applying (8.54) to the last term in (8.57), we obtain

$$\begin{aligned}
&(b_nc_n)^{-1}\sum_{j=3}^{n} c_{j-1}x_j\,(kT) = \sum_{j=3}^{n} g_jx_j\,(kT)\\
&= \sum_{j=k-n_0}^{k-1} g_{j-k+n_0+3}\left[a_{n1}\delta q\,(jT) + a_{n2}\delta W\,(jT) + b_n\delta p\,(jT) + d_n\delta C\,(jT)\right].
\end{aligned}\tag{8.64}$$

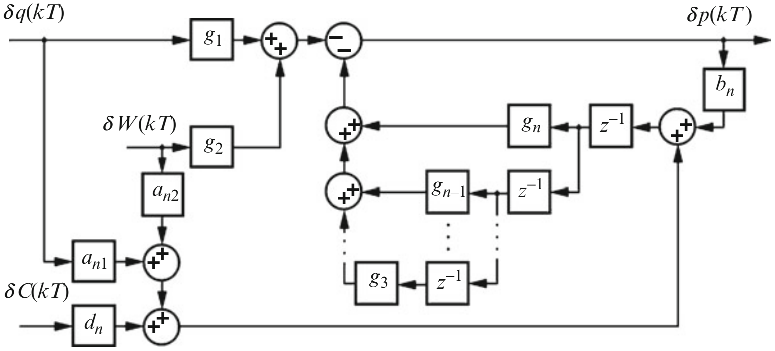


Fig. 8.6 Dead-beat controller for system with input and state delay

Consequently, using $x_1(kT) = \delta q(kT)$, $x_2(kT) = \delta W(kT)$, and (8.64), we can represent the dead-beat control law in the following form:

$$u(kT) = \delta p(kT) = -g_1 \delta q(kT) - g_2 \delta W(kT) - \sum_{j=k-n_0}^{k-1} g_{j-k+n_0+3} \left[a_{n1} \delta q(jT) \right.$$
$$\left. + a_{n2} \delta W(jT) + b_n \delta p(jT) + d_n \delta C(jT) \right], \tag{8.65}$$

where

$$g_1 = \frac{a_{11}c_1 + a_{21}c_2 + a_{n1}c_n}{b_n c_n},\ g_2 = \frac{a_{12}c_1 + a_{22}c_2 + a_{n2}c_n}{b_n c_n},\ g_j = \frac{c_{j-1}}{b_n c_n} \text{ for } j \geq 3. \tag{8.66}$$

Similarly as (8.39), equation (8.65) represents a fixed-gain variable-state feedback control law. All the controller coefficients can be computed off-line, which guarantees good operational efficiency. In order to establish the current packet marking rate $\delta p(kT)$, the controller uses the measurement of the instantaneous queue length $\delta q(kT)$ and the window size estimate $\delta W(kT)$. However, in contrast to the input-delay-aware (IDA) controller (8.39), the state-delay-aware (SDA) one (8.65) uses the past values of the queue length, window size, and bandwidth measurement in addition to the marking rate history recorded within the last RTT. The structure of the designed dead-beat controller is illustrated in Fig. 8.6.

8.3.2.3 Modified Control Law

The additional measurements used by the SDA controller (8.65) allow for determining a more exact marking rate value when the state remains in close neighborhood of the operating point. However, once the system departs from the

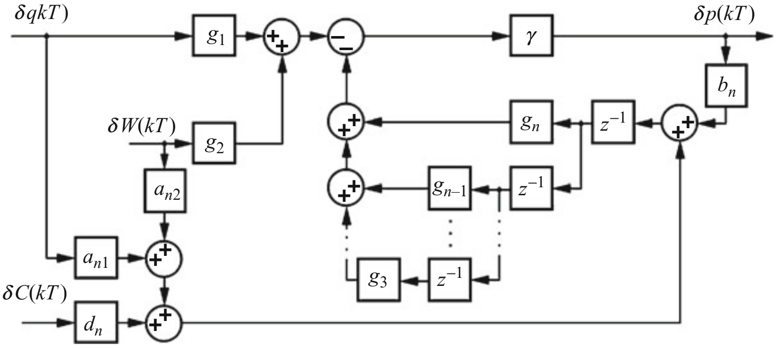


Fig. 8.7 Proportional controller for system with input and state delay

desired operating point, especially, when RTT undergoes substantial change, then these extra, noisy measurements may lead to decreased performance in terms of overshoots and oscillations, as compared to controller (8.39). Nevertheless, the controller robustness can be enhanced using a similar technique as discussed in the modified IDA controller (8.39). Consequently, in order to improve the controller performance in the presence of increased bandwidth variations, one may incorporate a scaling factor, $\gamma \in [0, 1]$, into the sliding plane as in (8.41). Then, after employing the modified plane (8.41) in (8.57), one obtains a proportional control law

$$\begin{aligned} u(kT) &= \delta p(kT) = -\gamma \left[g_1 \delta q(kT) + g_2 \delta W(kT) \right] \\ &\quad - \gamma \sum_{j=k-n_0}^{k-1} g_{j-k+n_0+3} \left[a_{n1} \delta q(jT) + a_{n2} \delta W(jT) + b_n \delta p(jT) + d_n \delta C(jT) \right] \end{aligned} \tag{8.67}$$

with g_j given by (8.66).

The block diagram of the proportional controller is illustrated in Fig. 8.7. Similarly as in the case of IDA controller, the dead-beat SDA scheme is changed by introducing additional gain element in the main signal path. The remaining elements and their configuration are depicted in Fig. 8.6. In particular, the delay compensator structure is retained.

8.4 Simulation Results

In this section, we test the performance of the designed discrete-time controllers regulating the flow of data in a TCP/IP network. The controllers are implemented as AQM packet marking schemes in routers to provide network-assisted flow control. The feedback information is conveyed to the sources by means of ECN bits.

Table 8.1 System parameters

Parameter	Value
N	50
β	0.5
T_p	0.2 s
R_0	0.5 s
T	0.1 s
q_0	100 packets
B_{size}	200 packets
C_0	333.333 packets/s
$C_{\max}$	550 packets/s
W_0	3.333 packets
$W_{\max}$	20 packets
p_0	0.18

8.4.1 System Parameters

We assume $N = 50$ TCP Reno sources passing through the router supporting AQM. The packet marking scheme is implemented in the router software according to (8.39) for the IDA controller, and according to (8.65) for the SDA controller. The propagation delay in the network $T_p = 0.2$ s. The nominal packet queue length is assumed $q_0 = 100$ packets, and the nominal link capacity $C_0 = 333.333$ packets/s which corresponds to 1.333 Mb/s with mean packet size of 500 bytes. Hence, the nominal round-trip time $R_0 = T_p + q_0/C_0 = 0.2 + 100/333.333 = 0.5$ s. With the sampling period $T = 0.1$ s, we get $R_0 = n_0 T = 5T$, and the system order $n = n_0 + 2 = 7$. Consequently, with $\beta = 0.5$, the window size and marking rate at the operating point

$$\begin{aligned} W_0 &= C_0 R_0 / N = 333.333 \cdot 0.5/50 = 3.333 \text{ packets}, \\ p_0 &= 1/\left(\beta W_0^2\right) = 1/\left(0.5 \cdot 3.333^2\right) = 0.18. \end{aligned} \tag{8.68}$$

The maximum window size $W_{\max}$ is assumed equal to 20 packets, the link capacity is set as $C_{\max} = 550$ packets/s, and the buffer size as $B_{\text{size}} = 200$ packets. The key system parameters are grouped in Table 8.1.

8.4.2 Controller Parameters

In the test, we verify performance of the designed dead-beat and proportional controllers and compare their operation with the classical RED and PI-AQM algorithms.

Applying the values listed in Table 8.1, we get the following sliding plane parameters for dead-beat controller (8.39):

$$\mathbf{c}^{\mathrm{T}} = \begin{bmatrix} -0.035 & -2.431 & 2.653 & 2.485 & 2.183 & 1.706 & 1 \end{bmatrix}.$$

Table 8.2 RED parameters

Parameter	Value
$p_{\max}$	0.1
$\min_{\text{th}}$	80 packets
$\max_{\text{th}}$	150 packets
α	0.0001
L_{RED}	0.0014
K	0.033
ω_g	0.12 rad/s

With this plane applied, the controller calculates the packet marking rate as $p(kT) = p_0 + \delta p(kT)$, where $\delta p(kT)$ is obtained from (8.39) and (8.40) as

$$\begin{aligned}\delta p\,(kT) = &-0.028\delta q\,(kT) - 2.456\delta W\,(kT) + 1.706\delta p\,[(k-1)\,T] \\ &+ 2.183\delta p\,[(k-2)\,T] + 2.485\delta p\,[(k-3)\,T] + 2.653\delta p\,[(k-4)\,T] \\ &+ 2.721\delta p\,[(k-5)\,T]\,.\end{aligned}$$

On the other hand, the sliding plane for controller (8.65) is determined as

$$\mathbf{c}^{\mathrm{T}} = \begin{bmatrix} 0.017 & 2.897 & 2.811 & 2.609 & 2.264 & 1.740 & 1 \end{bmatrix},$$

and the gain vector (8.66) as

$$\mathbf{g}^{\mathrm{T}} = [g_j]_{1\times 7} = \begin{bmatrix} 0.010 & 2.550 & 2.608 & 2.530 & 2.348 & 2.037 & 1.566 \end{bmatrix}.$$

The gain constant of proportional controllers (8.42) and (8.67) is set as $\gamma = 0.04$.

For the RED controller, we assume the following parameter values: $\min_{\text{th}}$ and $\max_{\text{th}}$ in the packet marking profile are set as 80 packets and 150 packets, respectively, which corresponds to the queuing delay in the range 0.24–0.45 s, $p_{\max}$ is adjusted as 0.1, and the averaging weight as $\alpha = 0.0001$. Hence, the AQM-RED controller gain $L_{\text{RED}} = p_{\max}/(\max_{\text{th}} - \min_{\text{th}}) = 0.0014$ and the low-pass filter pole $K = -\ln(1-\alpha)\cdot C_0 = 0.033$ (see [6]). The RED unity-gain crossover frequency is determined as $\omega_g = 0.1\min\{2N/R_0^2C_0,\ 1/R_0\} = 0.12$ rad/s. Parameters set in the tests are grouped in Table 8.2.

The transfer function of PI controller used for TCP/AQM flow regulation [7],

$$C_{\text{PI}}(s) = K_{\text{PI}}\frac{s/z+1}{s}, \tag{8.69}$$

is parameterized by the position of zero z, and the controller gain K_{PI}. According to [7], $z = 2N/R_0^2C_0$, which in our case results in $z = 1.2$. The gain constant K_{PI} is determined from

$$K_{\text{PI}} = \omega_g z\left|\frac{j\omega_g + 1/R_0}{C_0^2/2N}\right|, \tag{8.70}$$

Table 8.3 PI controller parameters

Parameter	Value
z	1.2
(a) K_{PI}	0.003
(b) K_{PI}	6.552×10^{-4}

where ω_g is the unity-gain crossover frequency (for the PI controller). In the tests, two K_{PI} settings are applied: the first value a, as suggested in [7], is calculated for ω_g equal ten times the RED crossover frequency, i.e., $K_{\mathrm{PI}} = 0.003$, and the second value b is adjusted to obtain increased robustness as $K_{\mathrm{PI}} = 6.552{\cdot}10^{-4}$ (it corresponds to the crossover frequency $\omega_g = 0.3$ rad/s). The applied parameters are listed in Table 8.3.

8.4.3 Test Results

A number of simulation scenarios are considered whose results are discussed in sections Test 1 and Test 2. In section Test 1, the controller set-point tracking capabilities are evaluated. It is verified if the controllers are able to stabilize the queue length at level q_0, and what are the transient characteristics. In particular, the controllers are compared with respect to the level of overshoots and oscillations. All the simulations in Test 1 are conducted in the disturbance-free environment. Next, in section Test 2, we verify the controller robustness to (unpredictable) bandwidth fluctuations and uncertain, variable number of TCP sessions participating in the flow control process.

Test 1. We assume that initially the buffer is empty, i.e., $q(0) = 0$, and test the controller ability to reach the queue length set-point value of $q_0 = 100$ packets. The number of TCP connections $N(\cdot)$ and the available bandwidth $C(\cdot)$ are assumed constant, equal to their nominal values N and C_0 given in Table 8.1.

The buffer occupancy resulting from the operation of the designed controllers is shown in Fig. 8.8 – curve a represents the dead-beat schemes (both input- and state-delay-aware controllers), whereas curve b reflects the queue length evolution in the system regulated by proportional controllers (8.42) and (8.67). For comparison, in Fig. 8.9, we illustrate the simulation results obtained for the RED and PI marking schemes. We can see from the graphs that each controller allows the buffer to be filled with data, and the nominal queue length is reached with similar rise times. It is also apparent from the figures that all the controllers generate an overshoot and, with the exception of RED, oscillations. The developed controllers provide faster convergence of the queue length to the set-point value than the classical RED and PI schemes. However, the dead-beat controllers exhibit small-amplitude oscillations around the set-point level also in steady state. This is attributed to high-frequency switching of the control signal in the interval [0, 1] which is caused by high sensitivity of dead-beat control to modeling inaccuracies. The proportional

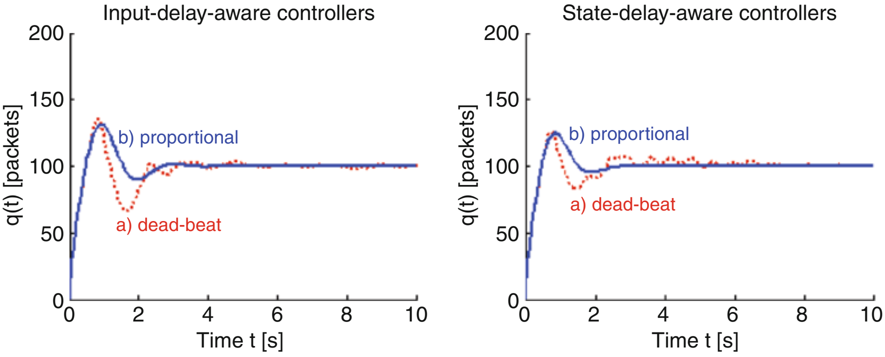


Fig. 8.8 Buffer occupancy: *a* dead-beat, *b* proportional controller with $\gamma = 0.04$

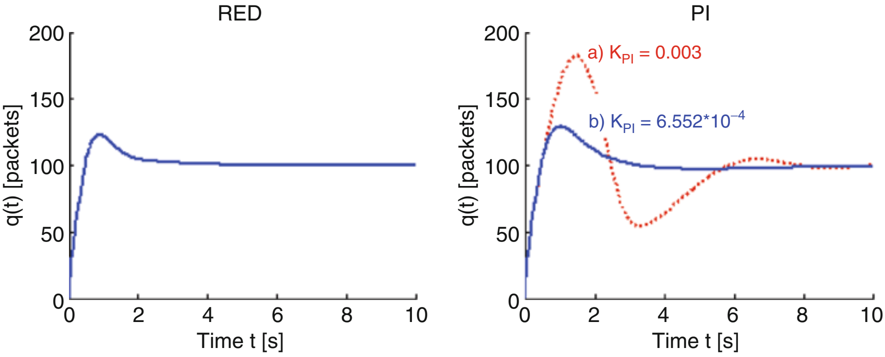
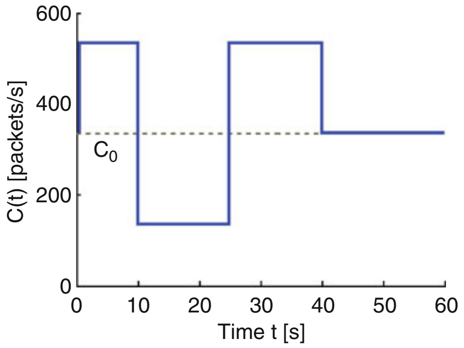


Fig. 8.9 Buffer occupancy RED and PI controllers: *a* $K_{\mathrm{PI}} = 0.003$, *b* $K_{\mathrm{PI}} = 6.552 \times 10^{-4}$

controllers never enter the (input) saturation region, thus providing smoother unit-step response. It is also clear from the plots presented in the figure that the designed SDA controllers ensure faster convergence of the queue length to the set-point value with smaller overshoot and reduced oscillations than the IDA ones. The steady-state queue length is achieved after 8 and 6 nominal RTTs in the case of proportional controllers (8.42) and (8.67), respectively, after 9 in the case of the RED marking scheme, and after 20 (setting a) and 19 (setting b) nominal RTTs in the case of the PI controller. The best set-point tracking is demonstrated by the SDA proportional controller. It generates small overshoot (similar to the RED scheme) and provides the shortest settling time. The worst performance, in turn, is demonstrated by the PI controller in setting a. The PI controller a generates unacceptably large overshoot (which in a perturbed environment may lead to buffer overflow) and very slowly decaying oscillations of large amplitude.

Test 8.2. The objective of the second series of simulations is to verify the controller robustness to changes in the available bandwidth and network load. In the graphs

Fig. 8.10 Available bandwidth



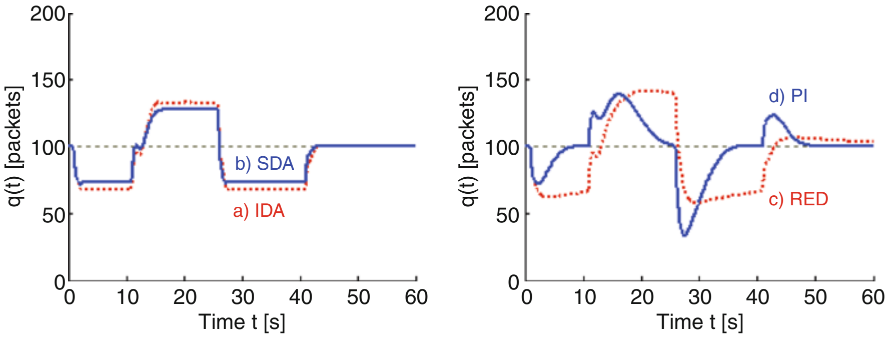


Fig. 8.11 Buffer occupancy ($C(\cdot)$ variable, $N(\cdot) = \text{const}$): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

which follow, curve a represents the operation of controller (8.42), curve b the operation of controller (8.67), both with the gain setting $\gamma = 0.04$, curve c reflects the action of the RED marking scheme, and curve d represents the test results of the PI controller with K_{PI} adjusted as $6.552 \cdot 10^{-4}$ (improved robustness setting).

Scenario 2.1. First, we test the controllers in the presence of bandwidth fluctuations illustrated in Fig. 8.10. The pattern presented in the graph reflects abrupt changes of large amplitude in the range of 60% from the nominal value of 333.333 packets/s. The system is assumed to be in equilibrium until $t = 1$ s when a rise in the available bandwidth is experienced according to the $C(\cdot)$ evolution depicted in Fig. 8.10. In the first simulation, the number of connections is assumed constant and equal to the nominal value $N(t) = 50$ TCP sessions.

The results of the simulations are illustrated in Figs. 8.11–8.14: the buffer occupancy in Fig. 8.11, the aggregate flow RTT in Fig. 8.12, the packet marking probability in Fig. 8.13, and the aggregate flow window size in Fig. 8.14.

We can see from the plots in Fig. 8.11 that despite large bandwidth differences from the nominal value, the queue length remains within the assigned buffer space and is positive. This implies that packet losses are avoided, and all of the available

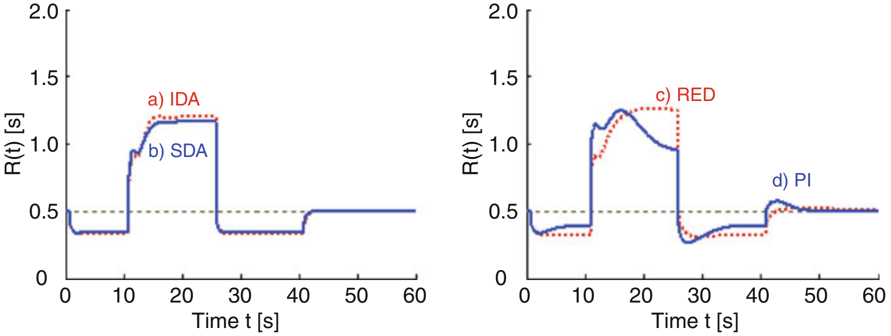


Fig. 8.12 Round-trip time ($C(\cdot)$ variable, $N(\cdot) = \text{const}$): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

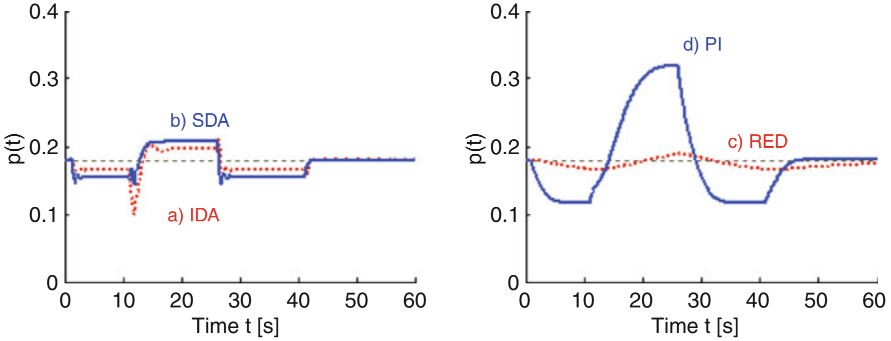


Fig. 8.13 Packet marking rate ($C(\cdot)$ variable, $N(\cdot) = \text{const}$): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

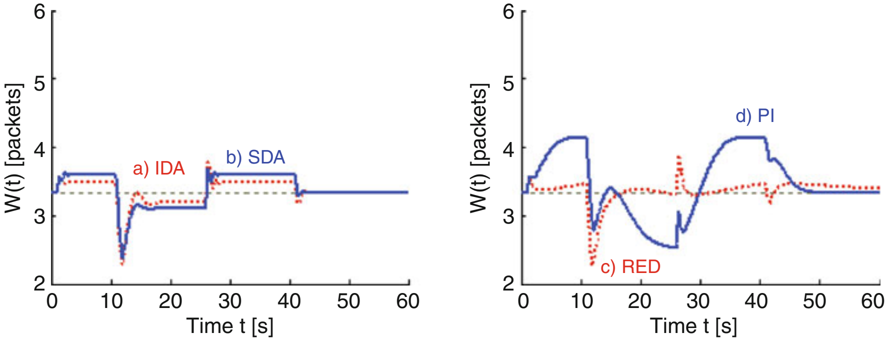


Fig. 8.14 Window size ($C(\cdot)$ variable, $N(\cdot) = \text{const}$): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

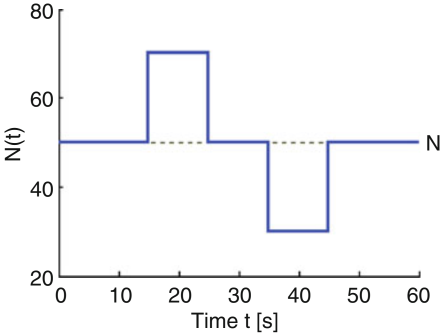


Fig. 8.15 Number of connections

bandwidth is efficiently used for data transfer. The designed SM controllers quickly follow the trend in the bandwidth variations. They provide smaller queue length drift from the nominal value than the RED and PI controllers. Although, in contrast to the PI scheme, the queue length converges to the level different from the operating point, the overall RTT follows a similar, yet less oscillatory, pattern as in the case of the PI controller (see Fig. 8.12). This means that the designed SM controllers ensure the same delay for subsequent packets in a data stream in periods of constant bandwidth even though the bandwidth actually available for data transfer differs from the nominal value C_0. Note also in Fig. 8.11 that once $C(\cdot)$ returns to C_0 (at $t = 40$ s), the output variable $q(\cdot)$ and the round-trip time $R(\cdot)$ rapidly converge to their equilibrium values. In this situation, RED and PI controllers exhibit slower convergence rate, with additional overshoot generated by the PI scheme.

It follows from Figs. 8.13 and 8.14 that packet marking rate remains in the linear range (0, 1) and window size does not grow excessively towards the maximum of 20 packets. The slowest reaction to the bandwidth variations is demonstrated by the RED controller, and the PI one generates the largest discrepancy of $p(\cdot)$ and $W(\cdot)$ from equilibrium. The designed SM controllers quickly bring signals $p(\cdot)$ and $W(\cdot)$ to new steady-state values (with a small overshoot and rapidly decaying oscillations), which are retained until further bandwidth change. Once $C(\cdot)$ returns to the level C_0, the packet marking rate a and b is quickly brought back to $p_0 = 0.18$ and the window size to $W_0 = 3.333$ packets.

Scenario 2.2. Similar observations as in Scenario 2.1 can be made when the number of TCP sessions varies with time. The results of the second series of simulations when $C(\cdot) = C_0 = const$, and $N(\cdot)$ progresses according to the pattern illustrated in Fig. 8.15, are shown in Figs. 8.16–8.19. Again, the designed SM controllers demonstrate higher degree of robustness than the classical schemes. The queue length shown in Fig. 8.16, similarly as RTT shown in Fig. 8.17, quickly converges to steady-state level without overshoots or oscillations after sudden, significant changes in the TCP load.

In addition to the excellent queue-length stabilization, the SDA controller also provides smooth, oscillation-free evolution of the packet marking rate and minute

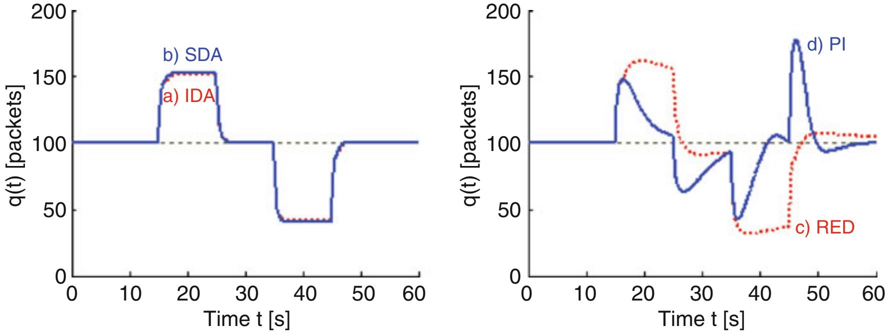


Fig. 8.16 Buffer occupancy ($C(\cdot) = \text{const}$, $N(\cdot)$ variable): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

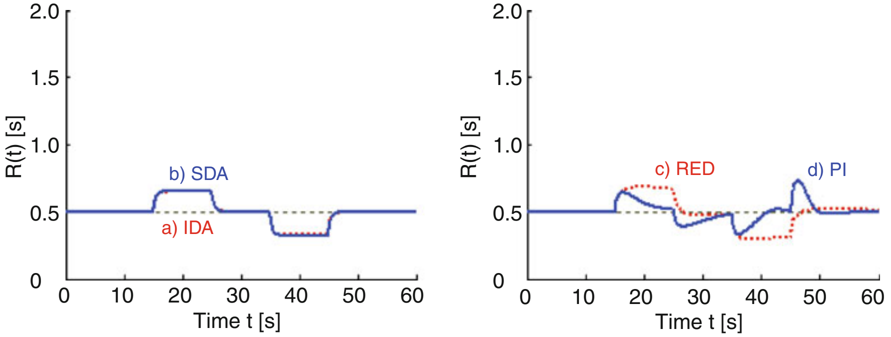


Fig. 8.17 Round-trip time ($C(\cdot) = \text{const}$, $N(\cdot)$ variable): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

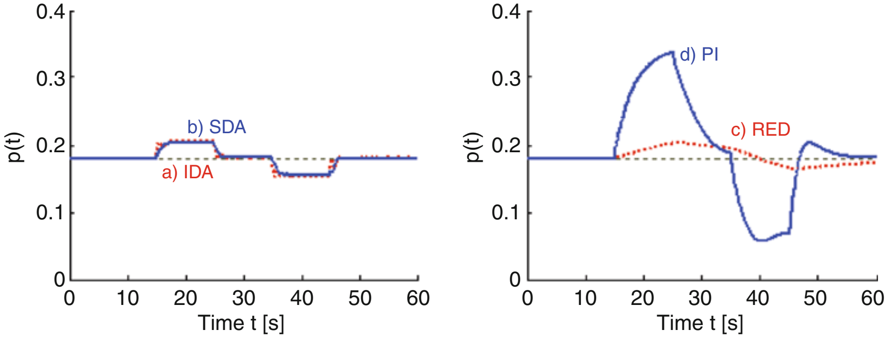


Fig. 8.18 Packet marking rate ($C(\cdot) = \text{const}$, $N(\cdot)$ variable): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

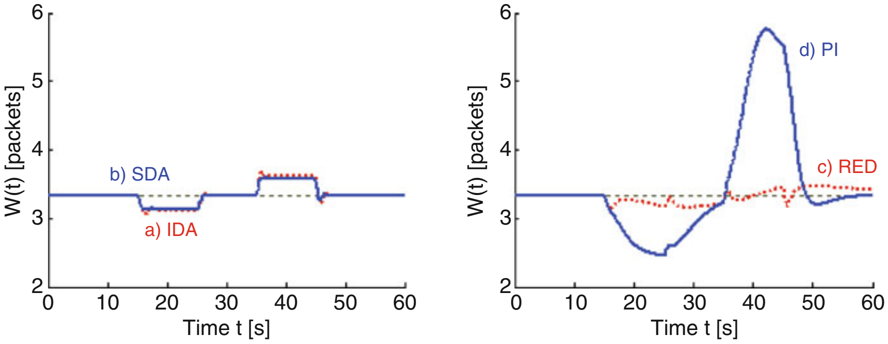


Fig. 8.19 Window size ($C(\cdot) = \text{const}$, $N(\cdot)$ variable): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

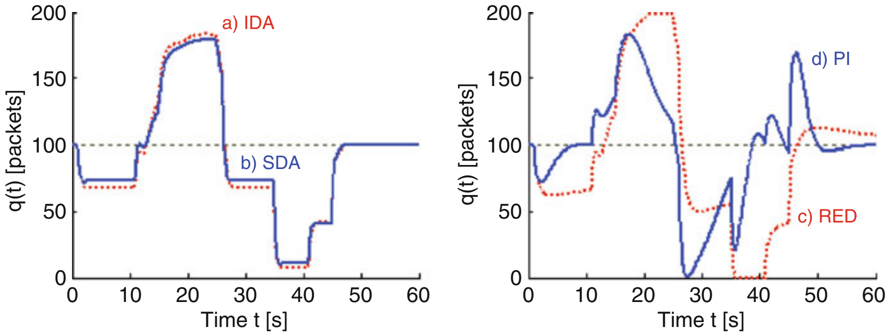


Fig. 8.20 Buffer occupancy ($C(\cdot)$, $N(\cdot)$ variable): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

oscillations in the window size, as depicted in Figs. 8.18 and 8.19, respectively (the curves obtained from the IDA controller show small-amplitude, decaying oscillations both in $p(\cdot)$ and $W(\cdot)$). The RED and PI schemes are more sensitive to the changes of the number of active connections than the SM controllers. In particular, the PI controller generates large overshoot both in $p(\cdot)$ and $W(\cdot)$ profile.

Scenario 2.3. In the third simulation scenario, we test the controller performance when both the link capacity and the network load vary with time. The $C(\cdot)$ evolves as depicted in Fig. 8.10, and the number of active connections changes according to the pattern shown in Fig. 8.15. The plots in Fig. 8.20 indicate that the queue length resulting from the application of SM controllers is positive and never exceeds the level of 200 packets. Consequently, the buffer is not overflowed, and the entire available bandwidth is used for transmission of data. The SM controllers quickly follow the capacity and load variations, maintaining constant values between the changes of networking conditions. The SDA controller provides slightly smaller queue length drift from the equilibrium value of 100 packets than the IDA one. The

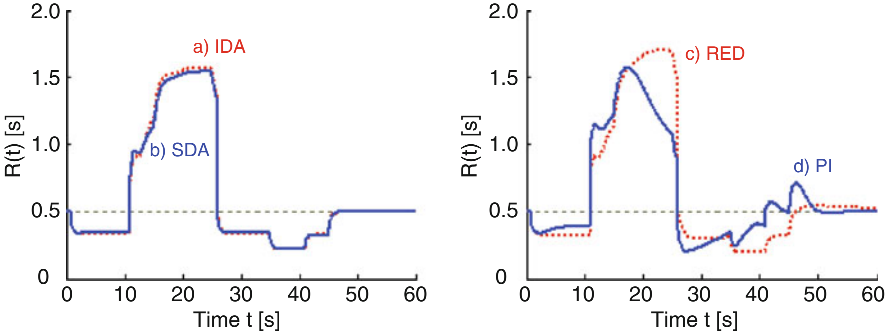


Fig. 8.21 Round-trip time ($C(\cdot)$, $N(\cdot)$ variable): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

RED controller is less robust than the SM ones to $C(\cdot)$ and $N(\cdot)$ fluctuations, whose combined effect leads to packet losses in the interval [20 s, 23 s], and decreased bandwidth utilization in the interval [35 s, 41 s]. The PI marking scheme maintains the queue length in the interval (0, 200 packets), but it exhibits overshoots and fails to converge to steady state between the changes of networking conditions due to long settling time. This also affects RTT, which in the case of the PI controller exhibits variations in nearly the entire simulation interval.

We can see from Figs. 8.22 and 8.23 that packet marking rate established by all the controllers always remains in the linear range (0, 1). The window size is kept below the maximum of 20 packets. Similarly as in the first scenario, the slowest reaction to bandwidth variations is demonstrated by the RED controller, and the PI one generates the largest overshoot in $p(\cdot)$ and $W(\cdot)$. In the case of the SM and RED controllers, the window size remains in the interval (2 packets, 4 packets) without large fluctuations, which helps in reducing the traffic burstiness. An important observation which follows from Figs. 8.20–8.23 is that once the perturbations vanish ($C(\cdot)$ and $N(\cdot)$ return to their equilibrium values), the SM controllers immediately bring the system back to the desired operating point. This is not the case for the classical schemes, which require much longer time to reach equilibrium.

Scenario 2.4. The fourth scenario deals with assessing the controller performance in a stochastic setting. Due to the noisy channel and the presence of uncontrolled traffic, $C(\cdot)$ may exhibit high-frequency oscillations. In the test, we assume that the available bandwidth follows the normal distribution with mean equal to $C_0 = 333.333$ packets/s and standard deviation 100 packets/s. Similarly, a large number of short-lived flows may cause frequent variations of the TCP load. In the simulation, we assume that the number of connections follows the normal distribution with mean equal to $N = 50$, and standard deviation 10. The actual $C(\cdot)$ and $N(\cdot)$ applied in the test are illustrated in Fig. 8.24. We can see from the plots that the available bandwidth varies in the range 80–550 packets/s, and the number of active TCP sessions in the range 25–70.

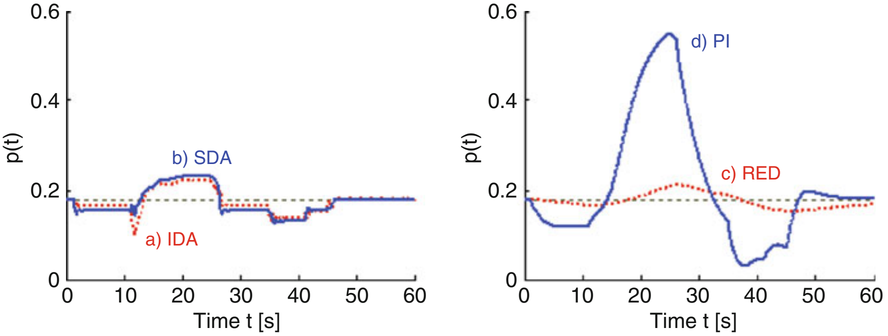


Fig. 8.22 Packet marking rate ($C(\cdot)$, $N(\cdot)$ variable): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

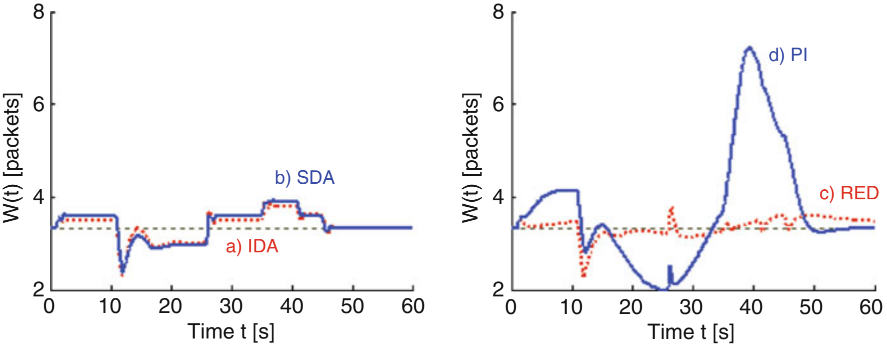


Fig. 8.23 Window size ($C(\cdot)$, $N(\cdot)$ variable): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

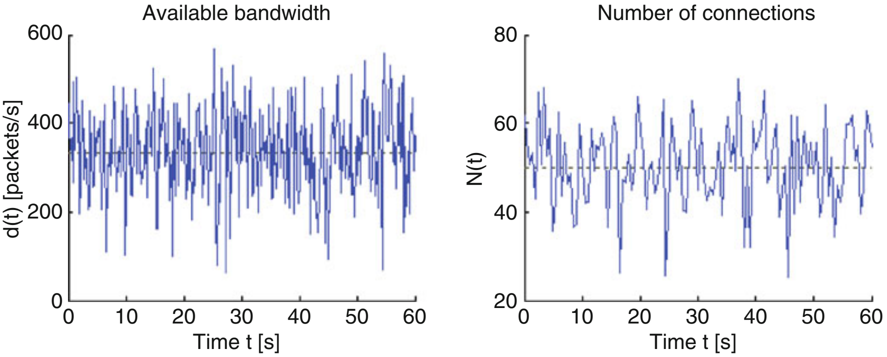


Fig. 8.24 Available bandwidth $C(\cdot)$ and number of active connections $N(\cdot)$ in stochastic setting

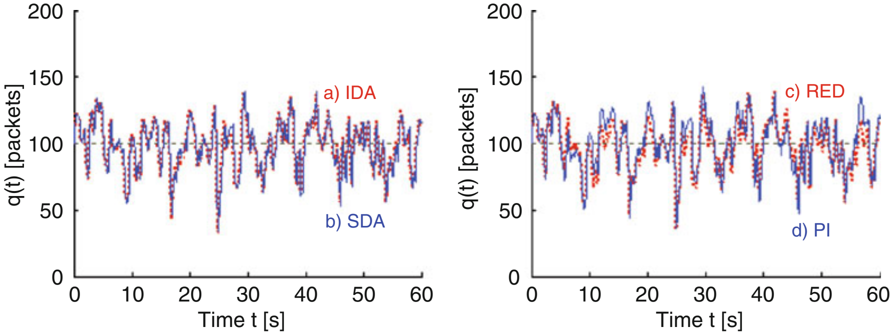


Fig. 8.25 Buffer occupancy (stochastic setting): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

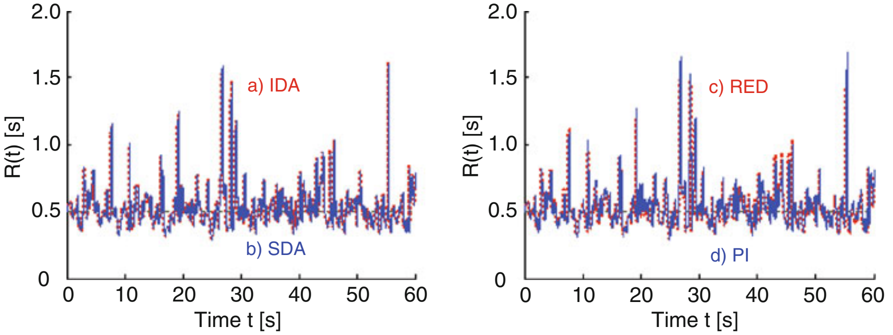


Fig. 8.26 Round-trip time (stochastic setting): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

The results of the simulation are depicted in Figs. 8.25–8.28: the buffer occupancy in Fig. 8.25, the flow RTT in Fig. 8.26, the packet marking rate in Fig. 8.27, and the window size in Fig. 8.28. The presented plots demonstrate that the closed-loop stability is maintained despite highly variable perturbations affecting the flow control process. The packet queue length evolves similarly for all the controllers and is positive during the whole simulation interval which implies full bandwidth usage. Moreover, the buffer is not overflowed, which implies that no packet needs to be rejected by the router, and the maximum throughput is ensured.

The low-pass nature of RED originating from the queue averaging filter (see the discussion in [6] and [7]) is clearly visible in Fig. 8.27, which depicts the marking rate established by the controllers. The $p(\cdot)$ signal generated by other controllers is affected by perturbations, with the smallest degree of oscillations shown by the proportional SDA one. Despite the apparent differences in the controller command, the output variable (the packet queue length) evolves similarly in the case of all four controllers – it fluctuates with nonincreasing amplitude around the set-point value

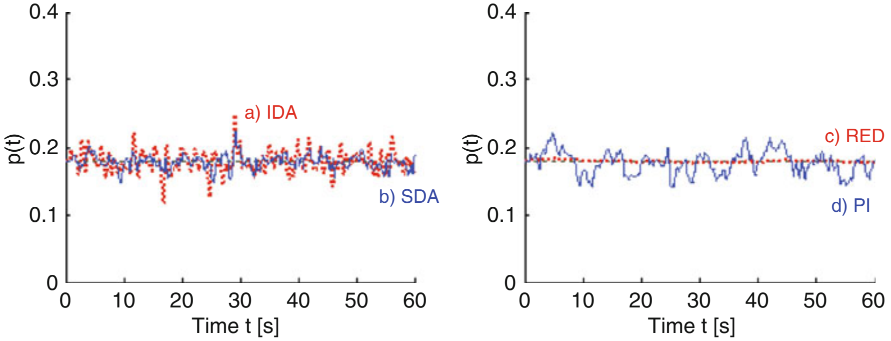


Fig. 8.27 Packet marking rate (stochastic setting): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

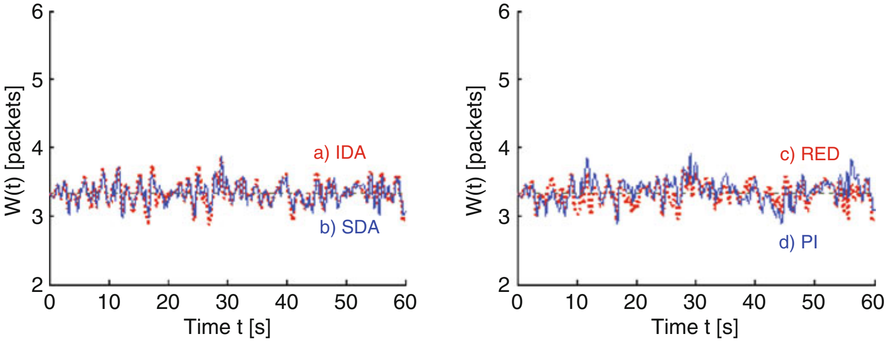


Fig. 8.28 Window size (stochastic setting): *a* proportional IDA, *b* proportional SDA, *c* RED, and *d* PI controller

of 100 packets. This shows that the external disturbances have major influence on the instantaneous queue length and the average is due to the properly established control signal with mean equal to the equilibrium value $p_0 = 0.18$.

8.5 Chapter Summary

In this chapter, we addressed the problem of data flow control in TCP networks in which the routers support AQM. Using the popular model describing the TCP congestion avoidance behavior [17], a few control algorithms were developed. Applying different linearization assumptions, two models approximating the nonlinear TCP dynamics around equilibrium point were obtained. The presented approach differs from the typical ones discussed in the literature in three primary aspects. First of all, explicit consideration is given to the effects related to finite sampling rate, and the controller derivation is conducted directly in discrete-time domain. Secondly, various kinds of network latency are included in the model representation with novel state-space representation. Finally, in contrast to the majority of previous solutions, the effects related to link capacity variations, which occur, for instance, due to the presence of uncontrolled traffic, or bit rate changes in the wireless environments, are directly considered in the system modeling and controller derivation. As a result, the designed controllers are more robust to the bandwidth variations than the classical AQM schemes proposed before.

In the first model, the information about the state delay was neglected, leaving only the input delay for explicit consideration. The obtained small-signal system representation was discretized and written in a state-space form. Extending the concepts discussed in Chaps. 5 and 6, the state variables were chosen so that the information about the input signal history could be preserved. The state-space system description was employed in the formal controller derivation. We applied the approach based on discrete-time SM control with the sliding plane selected for a dead-beat scheme. In this way, an input-delay-aware discrete-time SM controller was obtained. Next, a more sophisticated linearization procedure was considered so that, in addition to the input delay, also the information about the state delay could be explicitly incorporated in the control law derivation. The linear model was discretized and represented in the extended state space. The state variables were chosen in a special way so that the history of input and state variables could be preserved. The discrete-time state-space model was applied in the formal procedure of SM controller design. Again, the sliding plane was selected for a dead-beat scheme. The resulting control laws were provided in a closed-form with a constant gain vector precomputable ahead of the actual control process.

In order to improve the controller robustness, the structure of dead-beat controllers was modified by introducing a gain element used to perform scaling of the input signal. The improved, proportional controllers ensure high robustness to modeling uncertainty and external perturbations (such as capacity, or load variations), while maintaining much of the excellent dynamics of dead-beat schemes. The robustness was shown to be preserved even for a significant drift of the capacity, number of connections, and RTT from their equilibrium values (which were used for the controller derivation). Since all the controller parameters can be calculated offline, good operational efficiency in the network node implementation is guaranteed.

As demonstrated in extensive numerical studies, the designed SM controllers outperform the classical solutions, RED and PI packet marking schemes, both in the queue length set-point tracking, and in the resistance to disturbances and modeling inaccuracies. A possible drawback of the developed delay-aware controllers is that for persistent disturbance other than the nominal one, the queue length differs from the equilibrium value. However, this drawback, which originates from the predictor-like controller structure, can be leveraged by introducing extra feedforward disturbance compensation following the lines discussed in Sect. 7.4. An important point to notice in the operation of the presented controllers is that when the perturbations vanish ($C(\cdot)$ and $N(\cdot)$ return to their equilibrium values), the system state quickly reaches equilibrium, with the convergence rate several times faster than in the case of the classical AQM control techniques.

The results presented in this chapter may be further extended by incorporating a formal control-theoretic analysis of the system dynamics. For example, a valuable extension would be to establish the global stability bounds for different ranges of network parameter variation (as it was done in Chaps. 4, 5, 6, and 7). A closer look at the parameters of dead-beat and LQ optimal controllers presented in Chaps. 5 and 6 suggests that the proportional controller proposed in this chapter might result in the LQ optimal one for the linearized TCP dynamics. Therefore, another research direction would be to conduct the optimization procedure and verify whether the proportional controllers truly optimize the small-signal system dynamics in the LQ sense.

References

1. Almeida A, Belo C (2010) Explicit congestion control based on 1-bit probabilistic marking. Comput Commun 33:S30–S40
2. Brakmo LS, Peterson LL (1995) TCP Vegas: end-to-end congestion avoidance on a global Internet. IEEE J Sel Areas Commun 13:1465–1480
3. Davie BS, Peterson LL (2000) Computer networks: a systems approach. Morgan-Kaufmann, San Francisco
4. Floyd S (1994) TCP and explicit congestion notification. ACM SIGCOMM Comput Commun Rev 24:10–23
5. Floyd S, Jacobson V (1993) Random Early Detection gateways for congestion avoidance. IEEE/ACM Trans Netw 1:397–413
6. Hollot CV, Misra V, Towsley D, Gong WB (2001) A control theoretic analysis of RED. Proc IEEE INFOCOM, Anchorage, USA, 3:1510–1519
7. Hollot CV, Misra V, Towsley D, Gong WB (2002) Analysis and design of controllers for AQM routers supporting TCP flows. IEEE Trans Autom Control 47:945–959
8. Imer OC, Basar T (2001) Control of congestion in high-speed networks. Eur J Control 7:132–144
9. Jacobson V (1988) Congestion avoidance and control. In: Proceedings of the ACM SIGCOMM, Stanford, USA, pp 314–329
10. Katabi D, Handley M, Rohrs ChE (2002) Congestion control for high bandwidth-delay product networks. In: Proceedings of the ACM SIGCOMM, Pittsburgh, USA, pp 89–102
11. Kelly FP (2001) Mathematical modelling of the Internet. In: Engquist B, Schmid W (eds) Mathematics Unlimited – 2001 and Beyond. Springer, Berlin
12. Kelly FP, Maulloo AK, Tan DKH (1998) Rate control for communication networks: shadow prices, proportional fairness and stability. J Oper Res Soc 49:237–252
13. Kunniyur SS, Srikant R (2003) End-to-end congestion control schemes: utility functions, random losses and ECN marks. IEEE/ACM Trans Netw 11:689–702
14. Kunniyur SS, Srikant S (2004) An Adaptive Virtual Queue (AVQ) algorithm for Active Queue Management. IEEE/ACM Trans Netw 12:286–299
15. Low SH, Paganini F, Doyle JC (2002) Internet congestion control. IEEE Control Syst Mag 22:28–43
16. Mascolo S (2006) Modeling the Internet congestion control using a Smith controller with input shaping. Control Eng Pract 14:425–435
17. Misra V, Gong WB, Towsley (2000) Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In: Proceedings of the ACM SIGCOMM, Stockholm, Sweden, pp 151–160

18. Pitsillides A, Şekercioğlu Y, Ramamurthy G (1997) Effective control of traffic flow in ATM networks using Fuzzy logic based Explicit Rate Marking (FERM). IEEE J Sel Areas Commun 15:209–225
19. Srikant R (2004) The mathematics of Internet congestion control. Birkhäuser, Boston
20. Wu H, Ren F, Muc D, Gong X (2009) An efficient and fair explicit congestion control protocol for high bandwidth-delay product networks. Comput Commun 32:1138–1147
21. Xia Y, Subramanian L, Stoica I, Kalyanaraman S (2005) One more bit is enough. In: Proc ACM SIGCOMM, Philadelphia, USA, pp 37–48

Chapter 9
Summary and Conclusions

Watching how ubiquitous communication networks have become in recent years, there is no question about their importance for social life and modern economy. They provide means of information sharing and data exchange basically anywhere, anytime. However, the numerous advantages of being able to quickly communicate thoughts and ideas over large distances, and gain access to remote resources, can only be achieved if the communication system is administered in a proper way. A major impact on the condition of a network is attributed to the mechanisms of control. They should allow efficient resource sharing according to the set of constraints specified for a given network (such as the channel capacity). More importantly, however, the implemented control methods should provide appropriate reaction to the dynamically changing networking conditions. In this way, one may avoid bottlenecks, loss of data, and guarantee adequate service level.

This monograph was devoted to the problem of efficient data flow control in communication networks. We looked at the network as a dynamical system and addressed the traffic regulation problem from the control-theoretic perspective. From this point of view, the transport of data in the network can be perceived as a dynamical process with delay, subject to disturbances. Therefore, the primary attention in this work was directed to the effects of nonnegligible latency on the control system performance. We considered the traffic regulation problem in the context of a single-connection aggregated flow and in the case of multiple flows characterized by different delays controlled simultaneously. Depending on the level of details one would like to incorporate in the system description, the network can be modeled as continuous-time, discrete-time, or a sampled-data system. We covered all three cases: fluid-flow traffic approximation in continuous-time domain, discrete-time process with constant sampling period, and sampled-data system with variable discretization period. In the presented framework, one should notice in particular the novel state-space representation of network dynamics. In the adopted state space, the effects related to delay are projected into a set of state variables recording the delayed signal history. In addition to efficient modeling of the fundamental networking phenomena, such as storing packets in the node buffers, we showed

P. Ignaciuk and A. Bartoszewicz, *Congestion Control in Data Transmission Networks: Sliding Mode and Other Designs*, Communications and Control Engineering,
DOI 10.1007/978-1-4471-4147-1_9, © Springer-Verlag London 2013

how, in a consistent way, one can introduce more advanced characteristics of the data flow process. We indicated how one can analyze and express mathematically the effects related to bandwidth fluctuations, delay variations, packet reordering, source nonidealities, rate saturation, multiple bottlenecks, etc.

For each model, and each traffic scenario, a control algorithm was designed using formal, control-theoretic approach. Since high level of robustness is of paramount importance in the considered class of uncertain, perturbed systems, we applied the robust control method – sliding-mode control. In order to address the issues related to delay, a particular attention was given to the crucial step in the SM controller design, which is selection of the sliding plane. We proposed several useful choices of switching functions and sliding hyperplanes, such as the ones selected for dead-beat or LQ optimal control. In each case, the hyperplane (or the switching function) explicitly incorporates delay compensating features. As a result, the obtained controllers retain stability for arbitrary delays. A simple form of the designed control algorithms allows for detailed analytical study of the communication system properties. In each scenario, a set of conditions was formulated and strictly proved, showing how to avoid the congestion and, at the same time, ensure full bandwidth utilization and maximum throughput. These favorable properties are achieved with the proposed controllers implemented despite no prior knowledge on the nature of the bandwidth, or delay variations, which are treated as external disturbances in the system.

The presented methodology of modeling discrete time-delay systems, and the controller design using the principles of SMC and dynamical optimization [1–3, 6], can be effectively applied to other systems with nonnegligible latency. For example, the described methodology can be used to create regulation schemes for such important classes of problems as optimization of logistic processes, or remote plant control in networked control systems. Recently, the methodology described in this monograph has been employed in the modeling and control of the goods flow process in supply chain [4, 5, 7–9]. We considered both the traditional inventory systems [4, 5, 7], which can be modeled as integrators with delay, and the processes with deteriorating goods [8, 9], which are typically represented as first-order systems with nonnegligible latency.

References

1. Ignaciuk P, Bartoszewicz A (2008) Linear quadratic optimal discrete time sliding mode controller for connection oriented communication networks. IEEE Trans Ind Electron 55: 4013–4021
2. Ignaciuk P, Bartoszewicz A (2009) Linear quadratic optimal sliding mode flow control for connection-oriented communication networks. Int J Robust Nonlinear Control 19:442–461
3. Ignaciuk P, Bartoszewicz A (2009) Discrete-time linear-quadratic (LQ) optimal and nonlinear flow control in multi-source connection-oriented communication networks. Eur Trans Telecommun 20:679–688

4. Ignaciuk P, Bartoszewicz A (2010) LQ optimal sliding mode supply policy for periodic review inventory systems. IEEE Trans Autom Control 55:269–274
5. Ignaciuk P, Bartoszewicz A (2010) Linear-quadratic optimal control strategy for periodic-review inventory systems. Automatica 46:1982–1993
6. Ignaciuk P, Bartoszewicz A (2011) Discrete-time sliding-mode congestion control in multi-source communication networks with time-varying delay. IEEE Trans Control Syst Technol 19:852–867
7. Ignaciuk P, Bartoszewicz A (2012) LQ optimal and reaching law based sliding modes for inventory management systems. Int J Syst Sci 43:105–116
8. Ignaciuk P, Bartoszewicz A (2012) LQ optimal sliding-mode supply policy for periodic-review perishable inventory systems. J Franklin Inst 439:1561–1582
9. Ignaciuk P, Bartoszewicz A (2012) Linear-quadratic optimal control of periodic-review perishable inventory systems. IEEE Trans Control Syst Technol. doi:10.1109/TCST.2011.2161086

Appendix

In the Appendix, we discuss the tools applied for model validation and performing tests of the designed control algorithms.

In the control society, probably the most widespread tool used to verify control system designs is Matlab-Simulink. It allows for scalable modeling of interactions even in complex dynamical systems through the reuse of basic blocks and connectors (e.g., delays, integrators, and summers) and built-in library functions (e.g., routines for obtaining numerical solutions to Riccati equations). The simulation engine of Matlab-Simulink provides sufficient mathematical fidelity to handle high signal dynamics and to check the system stability and exactness of the boundary conditions. These characteristics make Simulink a baseline testing facility in the design of control systems prior to assembling the actual devices and constructing physical prototypes. However, certain phenomena associated with the application considered in this work – traffic flow regulation in communication networks – such as finite information processing time, simultaneous reception of multiple control units, packet reordering, etc., are difficult to emulate in Simulink. Therefore, in addition to the mathematical tools such as Matlab-Simulink, the telecommunication society would typically verify the performance of new protocols in discrete event simulators, such as ns2 [1] or OPNET Modeler [2]. The discrete event simulators concentrate on the aspects related to the actual packet handling in a data transfer system. In effect, they create a virtual environment that attempts to mimic the behavior of the real network. Unfortunately, the added complexity associated with the specifics of a particular transmission technology (and the applied testing framework) may lead to unintended side effects and bigger computational errors. As a result, the analysis of the algorithm principal properties and validation of the boundary cases may become obscure and inconclusive.

In order to enable detailed, accurate study of signal evolution in the analyzed transmission system, the numerical results presented in this monograph were mostly obtained from appropriate models constructed in Matlab-Simulink. However, the simulation under more strenuous networking conditions when issues related to finite processing time and computational imperfections introduce stochastic

P. Ignaciuk and A. Bartoszewicz, *Congestion Control in Data Transmission Networks: Sliding Mode and Other Designs*, Communications and Control Engineering,
DOI 10.1007/978-1-4471-4147-1, © Springer-Verlag London 2013

per-turbations, such as multiple flow control scenario reported in Sect. 6.1.2, Test 4, was performed in ns2. For this purpose, the ns2 framework has been extended to incorporate the proposed control algorithms and the principles of data and signaling traffic interchange. The ns2 modules developed in C++, detailed installation instructions, and sample Tcl (Tool Command Language) scripts for ns2 are available online at the following address:

http://www.zsk.p.lodz.pl/~ignaciuk/research/SMCC/book/SMC_com_netw.htm

In addition, the web page gives access to sample Simulink models to experiment with. The models are prepared with basic graphical user interface for easy parameter adjustment. On the indicated web page, one may also find extra figures and simulation results illustrating the algorithm performance when tested in Matlab-Simulink in comparison with the data obtained from ns2.

References

1. ns2 Network Simulator. http://www.isi.edu/nsnam/ns/
2. OPNET Modeler. http://www.opnet.com/solutions/network_rd/modeler.html

Index

P. Ignaciuk and A. Bartoszewicz, *Congestion Control in Data Transmission Networks: Sliding Mode and Other Designs*, Communications and Control Engineering,
DOI 10.1007/978-1-4471-4147-1, © Springer-Verlag London 2013