NONBLOCKING ELECTRONIC AND PHOTONIC SWITCHING FABRICS

Wojciech Kabaciński

NONBLOCKING ELECTRONIC AND PHOTONIC SWITCHING FABRICS

NONBLOCKING ELECTRONIC AND PHOTONIC SWITCHING FABRICS

WOJCIECH KABACIŃSKI
Poznan University of Technology

Springer

Wojciech Kabaciński
Poznan University of Technology
Poland

Kabaciński, Wojciech.
Nonblocking electronic and photonic switching fabrics / Wojciech Kabaciński.
p. cm.
Includes bibliographical references and index.
ISBN 0-387-25431-5 (alk. paper) -- ISBN 0-387-26530-9 (e-book)
1. Telecommunication--Switching systems. 2. Photonics. I. Title.

TK5103.8.K33 2005
621.382'16--dc22

2005049798

ISBN 0-387-25431-5 e-ISBN 0-387-26530-9 Printed on acid-free paper.
ISBN 978-0387-25431-9

Printed in the United States of America.

9 8 7 6 5 4 3 2 1 SPIN 11342403

springeronline.com

To my parents,
Renata,
Rafał, and Przemysław

Contents

Preface

Switching fabrics first appeared in telephone exchanges, where there was a need to interconnect many pairs of telephones installed in the network. Due to the large number of connected subscribers and inter-exchange links, switching fabrics in telephone exchanges have to serve a great number of input and output ports. Because of the scale, large switching fabrics were constructed from smaller ones. The way of building switching fabrics from elements of smaller capacity and different characteristics of switching fabrics topologies has been for a long time a rich area of theoretical research. The seminal work is due to C. Clos [23], who first considered multistage strict-sense nonblocking switching networks, and V. E. Beneš [9], who first introduced the mathematical theory of switching networks. From that time, many research was conducted in this field. Switching fabrics found its application not only in telecommunication, starting from telephone exchanges through ATM switches and IP routers to optical cross-connect systems and optical packet switches, but also in other areas of knowledge like computation and control. Theory of switching fabrics becomes also a part of applied mathematics. Results of studies carried out by researchers from these area were published in numerous papers and some books.

This book is intended for people interested in the switching theory, and especially combinatorial characteristics of switching networks. It contains a considerable amount of already known results. Some of them are presented in more detailed form, other are only mentioned. The book contains many original results accrued by the author during his work at Poznan University of Technology on combinatorial properties of switching fabrics. The contents of the book is partially taken from the author's lecture notes given at Poznan University of Technology for post-graduate and doctoral students. I hope this book will be useful to not only post-graduate and PhD students but also to engineers and

switching fabrics designers who want to gain knowledge on switching theory and especially on combinatorial properties of switching fabrics.

WOJCIECH KABACIŃSKI
BRANAOWO, 2004

List of Symbols

β – normalized capacity of input and output terminals of two-sided switching fabrics or terminals in one-sided switching fabrics

β_1 – normalized capacity of an input terminal of two-sided switching fabrics

β_2 – normalized capacity of an output terminal of two-sided switching fabrics

b – minimum weight of a multirate connection

B – maximum weight of a multirate connection

$\mathbb{BW}_i$ – the i-th blocking window

$\mathbf{C}$ – The state matrix of a crossbar switch

$C_{SD}(n_1, r_1, m, n_2, r_2, v)$ – an asymmetrical three-stage space-division v-dilated Clos switching fabric

$C_{SD}(n, r, m, v)$ – a symmetrical three-stage space-division v-dilated Clos switching fabric

$C_{TD}(n_1, r_1, f_1, m, n_2, r_2, f_2, v, f_0)$ – an asymmetrical three-stage time-division v-dilated switching fabric

$C_{TD}(n, r, f, m, v, f_0)$ – a symmetrical three-stage time-division v-dilated switching fabric

$C_{MR}(n_1, r_1, \beta_1, m, n_2, r_2, \beta_2, v)$ – an asymmetrical three-stage multirate v-dilated switching fabric

$C_{MR}(n, r, \beta, m, v)$ – a symmetrical three-stage multirate v-dilated switching fabric

d – number of inputs and outputs in a crossbar switch used to construct a d-nary switching fabric

$\mathbf{E}_r$ – an elementary permutation matrix of size $r \times r$

$\mathbb{E}$ – a set of edges in $\mathcal{G}$

f_1 – number of time slots in an input terminal of a two-sided switching fabric or of a terminal of one-sided switching fabrics

f_2 – number of time slots in an output terminal of a two-sided switching fabric

f_0 – number of time slots in an inter-stage link

f – number of time slots in input and output terminals of two-sided switching fabrics

$\mathcal{G}$ – a graph

$\mathcal{G}(\mathbb{V}, \mathbb{E})$ – a graph with a set of $\mathbb{V}$ nodes and $\mathbb{E}$ edges

$\mathcal{G}(, \mathbb{O}_j)$ – a graph representing connecting paths between input terminals in $\mathbb{I}_i$ and output terminals in $\mathbb{O}_j$

$\mathbf{H}_r^{(n)}$ – a matrix of size $r \times r$ representing a maximal assignment

I_i – switch i of the first stage of a multistage switching fabric

$\mathbb{I}_i$ – the set of input terminals in a $\log_2 N$ switching fabric with the same values on bits $x_{n-1}x_{n-2}\ldots x_{\lfloor n/2 \rfloor}$ of their binary representation, where i is binary represented by $x_{n-1}x_{n-2}\ldots x_{\lfloor n/2 \rfloor}$

(I_i, O_j) – a connection between any input terminal of the first stage switch i and any output terminal of the last stage switch j in a two-sided switching fabric

(I_i, O_j, s) – an s-slot connection between any input terminal of the first stage switch i and any output terminal of the last stage switch j in a two-sided switching fabric

(I_i, O_j, ω) – a connection of weight ω between any input terminal of the first stage switch i and any output terminal of the last stage switch j in a two-sided switching fabric

(I_i, I_j) – a connection between any terminal of the first stage switch i and any terminal of the first stage switch j in a one-sided switching fabric

(I_i, I_j, s) – an s-slot connection between any terminal of the first stage switch i and any terminal of the first stage switch j in a one-sided switching fabric

(I_i, I_j, ω) – a connection of weight ω between any terminal of the first stage switch i and any terminal of the first stage switch j in a one-sided switching fabric

$(\mathbb{I}_i, \mathbb{O}_j)$ – a connection between any input terminal in $\mathbb{I}_i$ and any output terminal in $\mathbb{O}_j$

$\widehat{i}$ – complement to i, i.e., when i is binary represented by $i_{n-1}i_{n-2}\ldots i_1 0$ than $\widehat{i}$ is binary represented by $i_{n-1}i_{n-2}\ldots i_1 1$

M_k – switch k of the second stage of a three-stage two-sided switching fabric or a two-stage one-sided switching fabric

m – number of the second stage switches in two-stage switching fabrics composed of triangular switches or in three-stage Clos switching fabrics

m – number of additional stages in $\log_2 N$ switching fabrics

N – number of input terminals and output terminals in a symmetrical two-sided switching fabric or terminals in one-sided switching fabrics

N_1 – number of input terminals in an asymmetrical switching fabric

N_2 – number of output terminals in an asymmetrical switching fabric

n – number of inputs and outputs in a switch

n – number of stages in banyan-type switching fabrics

n_1 – number of inputs in a switch of the first stage in multistage switching fabrics

n_2 – number of outputs in a switch of the last stage in multistage switching fabrics

$\mathbb{NP}_i$ – a set of connecting paths which goes through the nodes of shell i of the $\log_2(N, m, p)$ switching fabric

O_j – switch j of the last stage of a multistage switching fabric

$\mathbb{O}_j$ – the set of input terminals in a $\log_2 N$ switching fabric with the same values on bits $y_{n-1}y_{n-2}\cdots y_{\lfloor n/2 \rfloor}$ of their binary representation, where j is binary represented by $y_{n-1}y_{n-2}\cdots y_{\lfloor n/2 \rfloor}$

p – a number of planes in $\log_2(N, m, p)$ switching fabrics

Π – a permutation; a maximal assignment

Π^{-1} – inverse permutation of permutation Π

$\pi(x)$ – an output terminal input terminal x is to be connected to in Π

$\pi^{-1}(y)$ – an input terminal output terminal y is to be connected with in Π

q – number of output terminals in multicast connection; q-cast connection

q_1 – maximum number of input terminals which may take part in a multiconnection

q_2 – maximum number of output terminals which may take part in q-cast connection

r_1 – number of switches at the first stage of a multi-stage two-sided switching fabric

r_2 – number of switches at the last stage of a multi-stage two-sided switching fabric

r – number of switches at first and last stages of a symmetrical multi-stage two-sided switching fabric or at the first stage of a one-sided switching fabric

$\mathbf{S}$ – matrix representing the state of a two-sided three-stage switching fabric

$\mathbb{SBW}_i$ – a subblocking window

$\mathbb{SI}_i$ – the set of input terminals accessible from stage i in a $\log_2 N$ switching fabric

$\mathbb{SI}_i^m$ – the set of input terminals accessible from stage i in a $\log_2 (N, m, p)$ switching fabric

$\mathbb{SO}_i$ – the set of output terminals accessible from stage i in a $\log_2 N$ switching fabric

$\mathbb{SO}_i^m$ – the set of output terminals accessible from stage i in a $\log_2 (N, m, p)$ switching fabric

s – number of stages in a multi-stage switching fabric

s – number of time slots used by a multi-slot connection; s-slot connection

$s_{\max}$ – maximum number of time slots used by a multi-slot connection

t – a size of a blocking window

$T_{SD}(n, m, r, v)$ – a v-dilated one-sided space-division switching fabric composed of triangular switches with r first stage switches, m second stage switches, and n terminal per each first stage switch

$T_{TD}(n, m, r, v, f_0)$ – a v-dilated one-sided time-division switching fabric composed of triangular switches with r first stage switches, m second stage switches, n terminal per each first stage switch, and f_0 time slots per each terminal

$T_{MR}(n, \beta, m, r, v)$ – a v-dilated one-sided multirate switching fabric composed of triangular switches with r first stage switches, m second stage switches, n terminal per each first stage switch, each terminal of capacity β

$\mathbb{V}_{\langle x,y \rangle}$ – a set of nodes on a connecting path in bipartite graph representation of $\log_2 (N, m, p)$ switching fabrics

v	–	number of links between two switches in successive stages
ω	–	a weight of multirate connection
$\lvert\mathbb{X}\rvert$	–	the cardinality of set $\mathbb{X}$
$\lfloor x \rfloor$	–	the greatest integer lower than or equal to x
$\lceil x \rceil$	–	the lowest integer greater than or equal to x
$x \oplus y$	–	the rest from dividing x by y
$\langle x, y \rangle$	–	a connection between input terminal x and output terminal y in a two-sided switching fabric or terminals x and y in a one-sided switching fabric
$\langle x, y, s \rangle$	–	an s-slot connection between input terminal x and output terminal y
$\langle x, y, \omega \rangle$	–	a multirate connection of weight ω between input terminal x and output terminal y
$\langle x, \mathbb{Y} \rangle$	–	a multicast connection between input terminal x and a set of output terminals $\mathbb{Y}$
$\langle x, \mathbb{Y}, \omega \rangle$	–	a multicast connection of weight ω between input terminal x and a set of output terminals $\mathbb{Y}$
$\mathbb{Y}$	–	a set $\{y_0, \ldots, y_k\}$ of output terminals taking part in a multicast connection

Chapter 1

INTRODUCTION

1.1 What Is Switching?

Telecommunication networks are designed to convey information between users. This information is provided by the user through appropriate terminal unit (telephone set, PC, for example). When two users want to exchange information, their terminals are to be connected by a transmission system. When there are more than two users connected to the network who want to exchange information, transmission systems are to be provided between each pair of them. The example of the network with six telephone users is shown in Fig. 1.1. In this approach $N(N-1)/2$ links are needed when N users are to be connected. This is not practical since links are rarely used by users, and for large number of users such realization is infeasible. Therefore, switching nodes were introduced in telecommunication networks (see Fig. 1.2). A switching node provides, on request, connecting path between a pair of users. Users are connected to the node by one transmission system, which is called a subscriber loop in case of the telephone network. Depending on the way information is conveyed in a telecommunication network, this connecting path may be provided for the duration of a connection (circuit switching), or only when information is really transmitted (packet switching).

Practical telecommunication networks contain many switching nodes, usually connected between themselves in the hierarchical order. One example is the telephone network with local exchanges and different classes of transit exchanges, as is shown in Fig. 1.3. Another example is the Internet network with many core and edge routers used to convey IP packets between users (see Fig. 1.4).

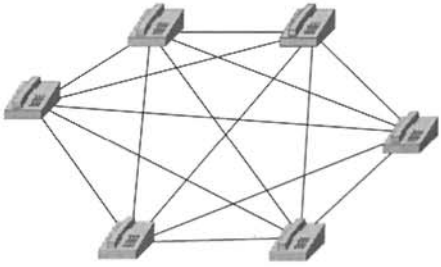


Figure 1.1. The example of the telephone network with six telephone subscribers

Figure 1.2. The example of the telephone network with six telephone subscribers and a switching node

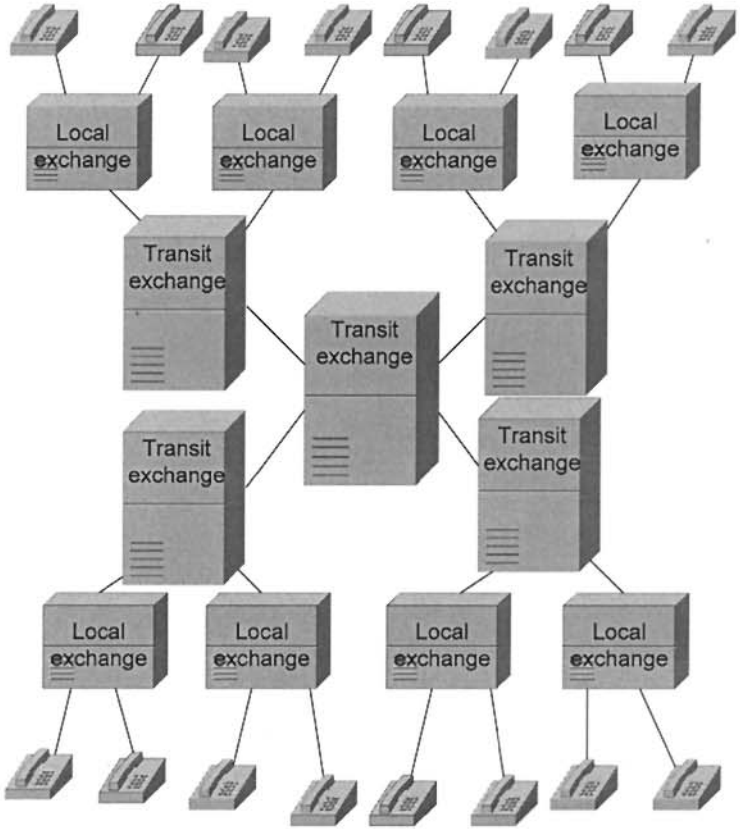


Figure 1.3. The telephone network

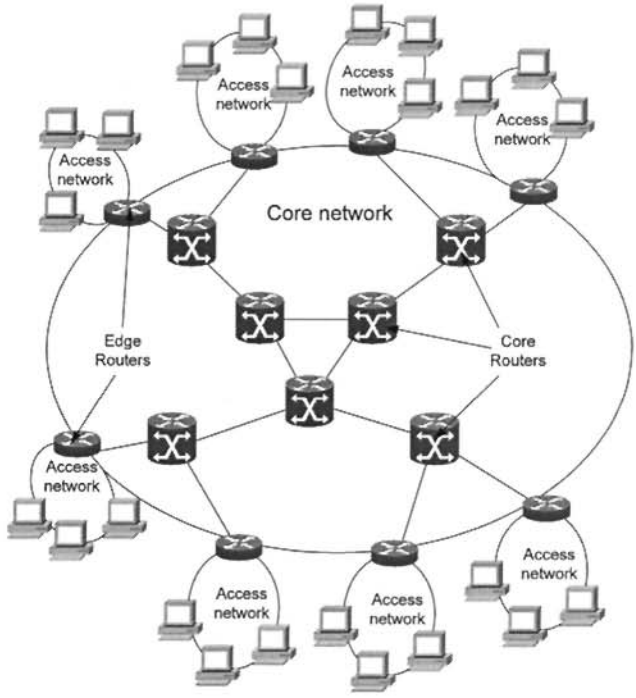


Figure 1.4. The Internet network

Telecommunication networks are only one example where switching fabrics are used. Another area is computer industry, where switching fabrics are implemented for instance in processor/memory interconnects for vector supercomputers, multicomputers and distributed shared-memory multiprocessors, clusters of personal computers or workstations, local, metropolitan or wide area networks (LANs, MANs, WANs) [39]. Switching fabrics are also used in other areas, for example in Viterbi decoders [3].

Different names are used for devices which perform switching functions. In telephone exchanges this part of exchange which performs switching was called a *switching network*. In multiprocessor systems the term *an interconnection network* or *a multistage interconnection network* (MIN) was used. When the speed rates of switched signals became greater and new transferring modes were introduced in the network (ATM networks, IP networks) the term *a switching fabric* or *a switch fabric* was introduced. In this book the term a switching fabric will be mostly used.

1.2 Evolution of Switching Technologies

The first switching technology implemented in telecommunication networks was *manual switching.* It was used in both telegraph and telephone networks. The first telephone exchange was installed in 1878, only two years after A. G. Bell invented the telephone. Telephone connections were set up by an operator on a manual switchboard by plugging a patch cord into respective jacks.

The next switching technology was *electromechanical switching.* Two kinds of electromechanical switching systems were installed in the telephone network. The first one was a step-by-step system invented by A. B. Strawger and implemented in the telephone network in 1892. This switching system was build from elements called Strawger switches. The other system is called the crossbar switching system. The first such system was installed in 1938. The crossbar switch used horizontal and vertical bars to select the contact. Such implementation of a switch is still used in switching, but the technology used has changed from electromechanical to electronic and photonic.

Invention of the vacuum tube and later the transistor started the electronic era not only in telecommunication industry. Binary devices in logic circuits, like gates and flip-flops, were used in different parts of switching systems for controlling and also for switching. First, *electronic switching* systems used analog electronic gates to switch analog signals from inputs to outputs. Implementation of digital transmission of voice signals in PCM systems made analog switching inconvenient, since digital signals had to be converted into analog form for switching, and then they had to be back digitized for transmission to the next switching exchange. Integrated circuits and electronic memories enabled to move from *analog switching* to *digital switching.* The first digital switching system was installed in 1976. Digital switching allowed substantial growth on the size of electronic switching systems. This switching technology is currently used in switching systems, not only in telephone exchanges but also in packet switches (IP routers) and digital cross-connect systems used in transport networks based on SDH/SONET systems.

Optical fibers introduced in transmission systems offer a huge transmission bandwidth unavailable for copper cables. Transmission bit rates of 2.5, 10 and 40 Gbps are now available and soon rates of 160 Gbps will be available commercially. Electronic switching cannot be used at such high rates, so incoming signals have to be not only converted from optical to electrical form but also have to be demultiplexed to lower bit rates. To omit this inconvenient and expensive signal conversion and demultiplexing, switching systems based on the optical technology have been elaborated in research laboratories and industry. *Optical*

Transfer Modes	STM	PTM	ATM
Switching Techinques	Circuit switching Multirate circuit switching Fast circuit switching	Packet switching Frame switching Frame relaying	ATM switching

Table 1.1. Transfer modes and switching techniques

switching, called also *photonic switching*, enables optical signals to be switched directly from inputs to outputs without conversion to electronic form. Optical switching technology is used in optical cross-connect systems installed in emerging automated switched optical transport networks (AOTN). Much research is also carried out to implement optical switching in switching nodes using packet and ATM switching.

1.3 Transfer Modes and Switching

The term transfer mode is used by ITU-T to describe a technique which is used in a telecommunication network, covering aspects related to transmission, multiplexing, and switching [33]. Three main transfer modes are called *Synchronous Transfer Mode* (STM), *Packet Transfer Mode* (PTM), and *Asynchronous Transfer Mode* (ATM). Different switching techniques are connected with these transfer modes. They are summarized in Table 1.1 and will be described in following sections.

1.3.1 Synchronous Transfer Mode

In STM transmission bandwidth is organized into periodic frames. Each frame consists of certain number of bits grouped into time slots, each of the same number of bits. The general architecture of the STM frame is shown in Fig. 1.5. The first time slot, denoted by TS0, usually contains synchronization pattern, which enables determination of the beginning of each frame at the receiver. Remaining time slots of each frame are used for conveying data.

Two types of STM systems are used in telecommunication networks. Duration of the frame is 125 μs, since 8 kHz sampling frequency was normalized for PCM systems used for voice circuits. Two formats of PCM systems are used worldwide. One format uses frames composed of 24 time slots (used in North America), the other uses 32 8-bit time slots (Europe). The latter system, called also PCM 30/32 or E1, uses TS0 for synchronization (called also synchronization channel), one time slot (usually TS16) for signalling (signalling channel), and the remaining

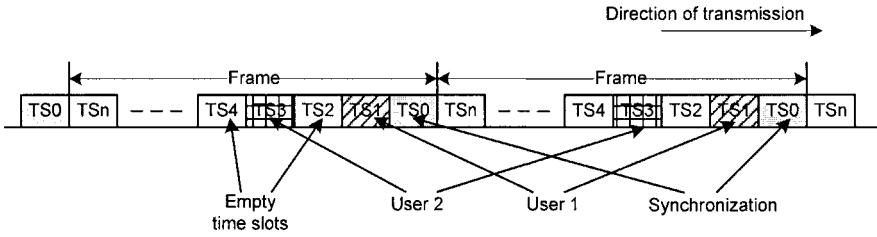


Figure 1.5. Synchronous Transfer Mode

time slots (called also data channels) are used for conveying user data. Another frame is STM-1 frame used in SDH systems. Both, E1 and STM-1 systems, are further multiplexed and form so cold digital hierarchies, called Plesiochronous Digital Hierarchy (PDH) and Synchronous Digital Hierarchy (SDH), respectively.

Circuit switching technique is used in STM mode. In circuit switching time slots are assigned to users on the call-by-call basis. When a time slot is not assigned, it is free but it is always present in the frame. Assigned time slots carries only user information. Source and destination addresses are determined by the number of time slot which is assigned to the connection during a call set-up phase. In other words, the time slot number is a label which is used to route data in this time slot, since each connection always use the same time slot in the frame during the complete duration of the connection.

Circuit switching provides a fixed bit rate for transmission. For instance, in E1 system this rate is 64 kbps (8 bits per frame, 8000 frames per second). *Multirate circuit switching* was proposed to overcome the inflexibility of this single bit rate when services with different bit rate requirements were introduced. In multirate circuit switching, called also *multi-channel* or *multi-slot switching*, one connection may occupy more than one channel of basic bit rate. This type of switching is used in N-ISDN (Narrowband Integrated Services Digital Networks) for example, for videotelephony.

Another inconvenience in circuit switching is that occupied channels cannot be used by other connections when no information is being sent by users. To use channels more efficiently, fast circuit switching has been proposed [133]. In *fast circuit switching* information on the required bandwidth, destination, and label identifying the connection are assigned during the connection set-up phase, but channels are allocated dynamically only when information is being sent from the source to the

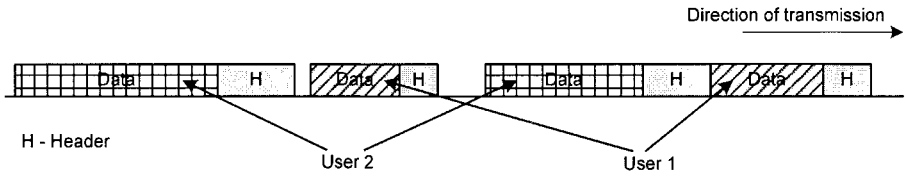


Figure 1.6. Packet Transfer Mode

destination. This approach, however, has not found practical implementation in the telecommunication network.

1.3.2 Packet Transfer Mode

In PTM user data are transferred by structured sequences of bits called packets. Each packet, apart from user data, also contains additional information which is used inside the network for routing, error control, flow control, etc. This information is placed at the front of packets and is called a header (some additional bits, like error correcting codes, are located at the end of packets). Packets have a variable length and may be transmitted at any time, provided that no other packet is transmitted in a transmission link. The concept of PTM is shown in Fig. 1.6. Unlike in STM mode, a new packet may appear on the link at any time, and when there is no information to send, no packet is transmitted in a link.

Packet switching technique is used with PTM. Different types of telecommunication networks were based on PTM, starting from telegraph network which may be considered as a packet switching network. Other networks like X.25 and its alternative solutions (i.e., *frame relaying* and *frame switching*) or Internet also use PTM.

1.3.3 Asynchronous Transfer Mode

Asynchronous Transfer Mode combines features of STM and PTM. In ATM data are transferred in fixed length packets called cells. Similarly, as in STM mode, cells are transmitted synchronously one after another, and a time for transmitting one cell is called a time slot. When no information is to be sent an empty cell is transmitted. However, transmitted cells are not structured into frames, like in STM mode. Users can insert data into any empty cell, so cells transmitted from a user may appear at any time when new cell (time slot) is started, similarly as in PTM, and time slot number cannot be in this mode used as the label (address information). Therefore, each cell contains a header

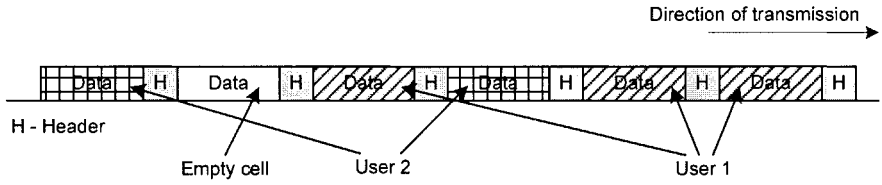


Figure 1.7. Asynchronous Transfer Mode

with a label, which is used to direct a cell to a destination user. A label is assigned to the connection at the connection set-up phase and is released after the connection is terminated. When no information is to be sent when a connection is set up, empty cells are sent. This approach enables the use of transmission bandwidth more efficiently than in STM, where time slots occupied by connections cannot be used by other connection even if no data is being transmitted. The concept of ATM is shown in Fig. 1.7.

1.4 Architectures of Switching Nodes

1.4.1 General Switch Architecture

The general architecture of a switching node is presented in Fig. 1.8. It contains N input modules, N output modules, a switching fabric, a control unit, and a management unit. Each input link is connected to one input module. Functions realized in the input module depends on the transmission method used in the link. In general, it converts line signals to signals suitable for processing and transmitting in the switching node. It also synchronizes frames, extracts signalling information and passes it to the control unit, and prepares signals for transmission through the switching fabric. The switching fabric transfers input signals to requested output modules through connecting paths, which are set up for this purpose. At the output module, signals received from the switching fabric are prepared for further transmission through the output link. This includes framing, insertion of signalling information if required, and transcoding signals to the line code and form appropriate for the output link. The control unit processes connections, sets up connecting paths through the switching fabric, processes signalling information, handles errors in call processing, and performs traffic routing and management functions. The management unit manages the configuration of the switching node, performs testing, billing and security management. Some other units, not shown in Fig. 1.8, like main distribution frame and powering, are also parts of the switching node.

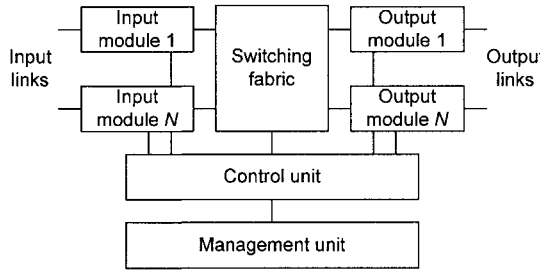


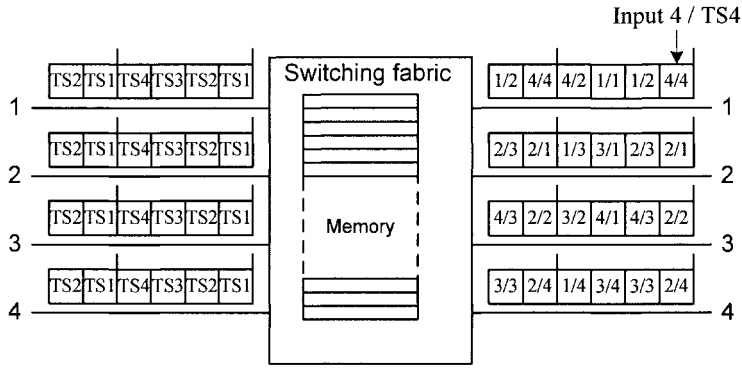
Figure 1.8. The general switching node architecture

We will now describe architectures and functions of switching nodes in several telecommunication networks, and give also some other examples where switching fabrics are used.

1.4.2 Telephone Exchange

Telephone subscribers used to be connected to the telephone exchange through subscriber line interface circuits (SLIC) placed in remote units called concentrators. Concentrators were connected to the main exchange by means of the set of E1 links. This type of links was also used for connections with other exchanges. Switching fabrics used mainly circuit switching with time-division multiplexing TDM (called also *TDM switching*). The principle of TDM switching is shown in Fig. 1.9. The switching of a time slot from incoming link to outgoing link is controlled by a translation table, which contains the relation between the slot number of an incoming link and the associated slot number in an outgoing link [33]. For instance time slot TS1 of input link 1 will always be switched to TS3 of output link 1. This relation is established during connection set-up phase and is maintained until the connection is terminated. The contents of the translation table is modified when a connection is set up or released. In the switching fabric switching can be done in switching elements which performs this function only in space (i.e., they move a time slot form one input link to the same time slot of any output link), time (a time slot form the input link is switched to any time slot of the output link), or in a combination of both. The switching fabric of Fig. 1.9 performs the switching function in both domains. The same approach is used in mobile switching centers (MSC) of GSM networks.

Currently, subscribers are more often connected to the remote subscribers modules which are connected to a main exchange by means of access networks.



Translation Table

Input	Slot	Output	Slot
1	1	1	3
	2	1	2
	3	2	4
	4	4	4
2	1	2	1
	2	3	1
	3	2	2
	4	4	1
3	1	2	3
	2	3	4
	3	4	2
	4	4	3
4	1	3	3
	2	1	4
	3	3	2
	4	1	1

Figure 1.9. TDM switching - principle of operation

1.4.3 ATM Switches

Functions of the input and output modules of ATM switch, called input and output port controllers, respectively, are shown in Fig. 1.10. Input and output port controllers may be fabricated on one circuit board called the line interface card. Input line is connected to the physical interface which converts incoming bit stream from line code to binary code (optical to electronic conversion is also performed when optical fiber is used), synchronizes bits and frames (when cells are transmitted in the frame, for instance SDH/SONET or E1), end process information in the header of the frame. Extracted cells are passed to the cell synchronization units, which delimits cells' boundaries, checks correctness of received cells, discards cells with errors in header and empty cells, and passes remaining cells to the cell processing unit. In the cell processing unit cells are prepared for transmission through the switching fabric.

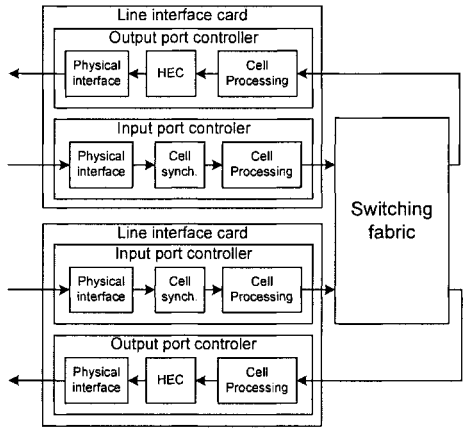
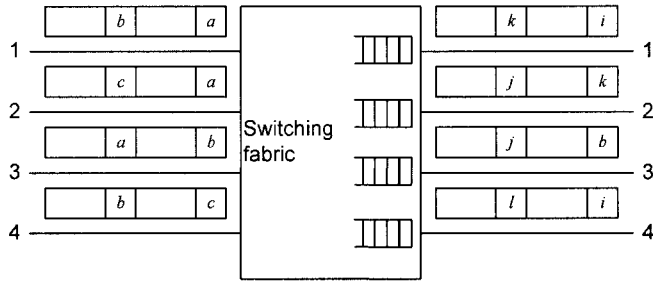


Figure 1.10. ATM switch

The destination output port for each cell is determined using routing information table and an internal header is added. Depending on the switch organization buffers may be also placed in input module to overcome output contention problems.

At the output port controller, cells from the switching fabric are passed to the cell processing unit, which removes internal header and inserts new header with the label assigned to the cell on the outgoing link. In HEC unit header error check code is calculated and inserted in the respective field of cell's header, and empty cells are generated if necessary. Then cells are put into frames (if framed transmission is used) and converted to the signal appropriate for transmission in the output link. Output buffers are also located at the port controller.

The principle of ATM switch operation is shown in Fig. 1.11 [18, 33]. Each cell is directed to the requested output according to the label located in its header. This label contains VCI (*virtual channel identifier*) and VPI (*virtual path identifier*). The label has a local meaning, is assigned during connection set-up, and is placed in the routing information table together with output port number and the new label which is to be used in the output link. The example of such routing table is also shown in Fig. 1.11. In each time slot, destination ports for cells from each input port are taken from the routing table, routes through the switching fabric are determined, connecting paths are set up (in case the switching fabric is self-routing, connecting paths are determined in the switching fabric switch-by-switch from the output port address placed in the header at the cell processing unit of the input port controller), and cells



Routing Information Table

Input	Header	Output	Header
1	*a*	1	*i*
	b	2	*j*
2	*a*	1	*k*
	b	2	*i*
	c	3	*j*
3	*a*	4	*l*
	b	2	*k*
4	*a*	1	*k*
	b	3	*l*
	c	4	*i*

Figure 1.11. ATM switching - principle of operation

are transferred to output ports. In ATM switching it may happen that in a given time slot, two or more cells should be directed to the same output port. This phenomenon is called output port contention and is solved by buffering cells. Buffers may be placed in different units of the switch, for instance at inputs, at outputs, or inside the switching fabric.

1.4.4 IP Routers

IP routers can be categorized depending on their size into low-end routers, middle-size routers, and high-end routers [18]. The first two classes performs switching functions in software. Line cards are connected to the central processing unit through a shared bus. Processing units can be also placed on line cards for packet forwarding, in order to reduce the central processing unit load. The capacity of the processing unit and central bus speed limits the capacity of such types of routers. In high-end routers of large capacity a switching fabric is used to switch packets between inputs and outputs. The general architecture of such a router is shown in Fig. 1.12 [16, 18]. Ingress line card contains physical interface and packet processor. The physical interface performs optical-to-electrical and serial to parallel conversions. It also synchronizes incoming bits, processes frame overhead and delineates packets. Packets are then processed by the packet processor, which performs table lookup and packet classification. The packet processor also performs

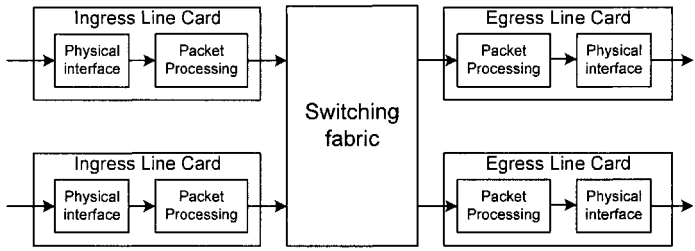


Figure 1.12. High-end IP router

various control functions like traffic access control, buffer management, and packet scheduling. All these functions may be implemented in more than one unit. Most switching fabrics use synchronized packet switching, i.e., variable length input packets are segmented into fixed length packets for transferring through the switching fabric. This function is also realized in the packet processor.

After switching, fixed length packets are again processed by packet processor at the egress line card. They are reassembled to original variable length packets, buffered and scheduled for transmission. At the physical interface packets are placed in appropriate frame (SDH/SONET), frame header is generated, and then the bit stream is converted from parallel to serial and from electrical to optical form.

Fixed length packets are called cells, but they do not have the same length as ATM cells. The principle of switching is the same as for ATM switching.

1.4.5 Cross-connect Systems

Cross-connect systems are switching nodes used in transport networks. When transport networks is based on SDH/SONET systems and switching is made in electronic form, the switching nodes are called digital cross-connect systems (DXCs) and digital add/drop multiplexers (ADMs). The principle of switching in these systems is similar to TDM switching. Currently optical transmission is used in transport networks and DXCs are being replaced with optical cross-connect systems (OXCs) and optical ADMs (OADMs). The optical transport network is shown in Fig. 1.13. It provides connecting paths between users (telephone exchanges, IP routers). This connecting path is also called the lightpath, and it uses one optical channel (one wavelength) in an optical fiber. The optical transport network may use ring or mesh topology. OADMs are placed on the network edge, and provide access to the optical transport

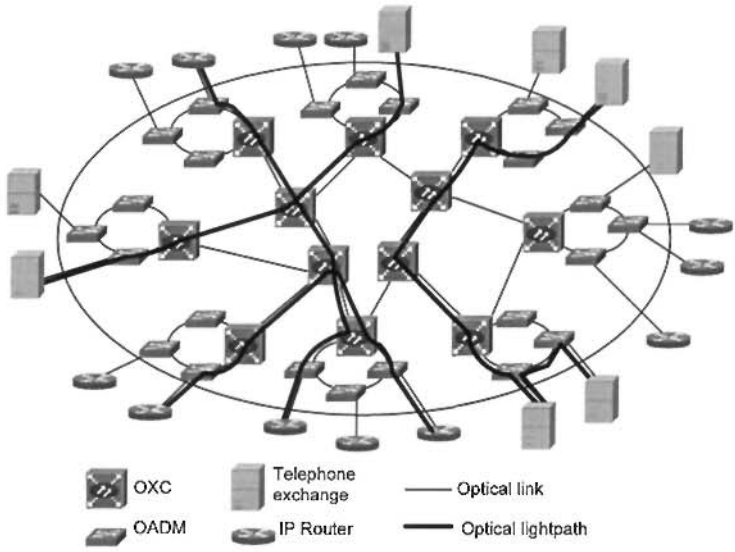


Figure 1.13. An Optical Transport Network

network. OADMs are also useful in building simple optical networks with small numbers of nodes and wavelengths.

Different architectures of OADMs were proposed in literature and implemented in practice [142]. Some of them use optical switching fabric to switch wavelengths. Two examples of such architectures are shown in Fig. 1.14. In the first example simple 2×2 switching elements are used. After wavelength demultiplexing each wavelength passes through the switching element and depending on the state of this element the wavelength is either switched through or dropped. When the wavelength is dropped, the same wavelength is also added through the switch. After switching, wavelengths are back multiplexed to the output fiber. The same function can be realized using the switching fabric of greater capacity, as is shown in Fig. 1.14b.

Several architectures were also considered for OXCs. One of these architectures with several switching fabrics, each for switching signals on different wavelengths, is shown in Fig. 1.15. Wavelengths from each input fiber are firstly demultiplexed and then each wavelength is switched by the different switching fabric. Each switching fabric switches only the same wavelength. After switching, wavelengths are back multiplexed to optical fiber. Instead of n switching fabrics (when n wavelengths are multiplexed in one optical fiber), one switching fabric of greater capacity can be used. Optical wavelength converters may be also used at outputs

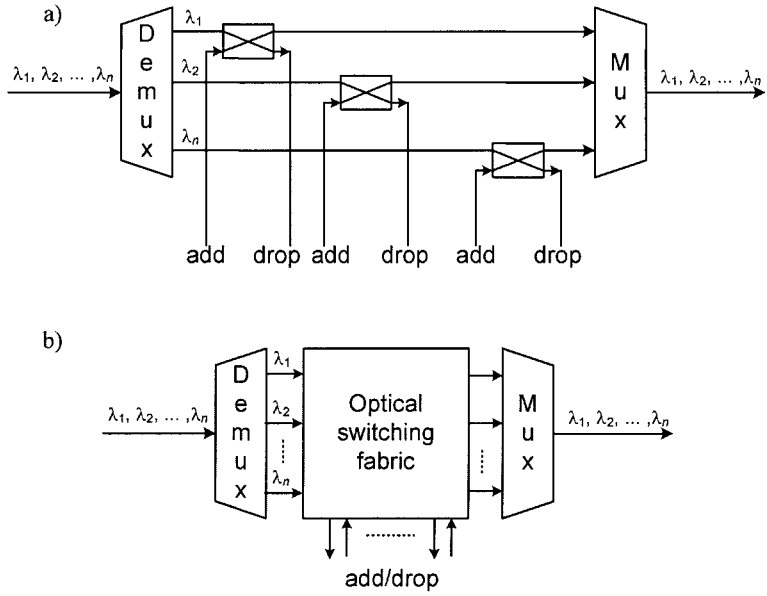


Figure 1.14. OADMs architectures with switching elements (a) and with the switching fabric (b)

or inputs of the switching fabric to convert incoming or outgoing signals from one wavelength to another.

OXCs uses mostly circuit switching technique, however, optical packet switching and optical burst packet switching are considered for use in core routers [176, 131, 171]. Optical packet switching, however, will require faster optical switching elements and optical buffers.

1.4.6 Switching in Multiprocessor Systems

The growing need for fast computing and the limited speed of processors lead to the proposition of the new architecture of computer systems. Parallel computers were designed to increase processing power. Such a computer contains multiple processors connected to memory and other input/output devices. Processors cooperate to solve a large problem and memory components are distributed among processors. In such architecture, some kind of communication subsystem is required to connect processors with memories and other peripherals. It may be done using system buses, but when the number of processors and devices connected to the bus increases, then the bus becomes the communication bottleneck that degrades the performance of the system. The number of devices connected to the bus is also limited by electrical load character-

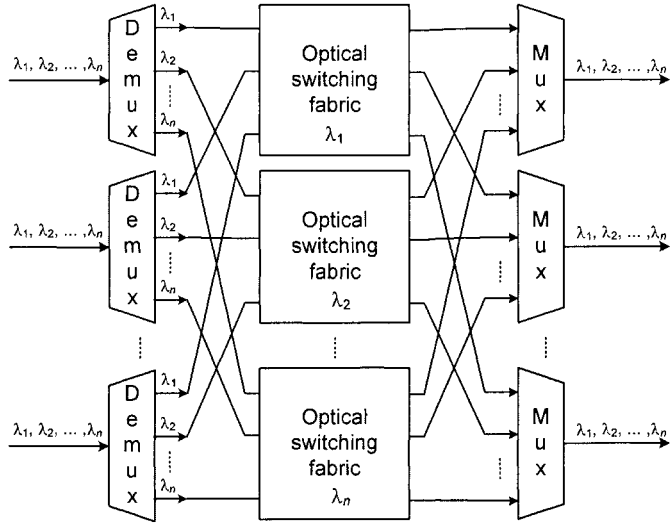


Figure 1.15. OXC with separate switching fabrics for each wavelength

istics. Another solution is to use switching fabric in which connections between processors and memory components will be done on request. Such a multiprocessor system is shown in Fig. 1.16. Communications between devices connected to the switching fabric is realized by means of messages. The switching may use circuit switching technique, where circuit connection is established between devices before a message is sent. Another option is packet switching of fixed length packets. Different variations of switching packets in the switching fabrics where proposed, like *virtual-cut through switching*, *wormhole switching* or *mad postman switching* [39]. Their purpose is to reduce the latency and increase the performance of the switching fabric. For instance, in virtual-cut through switching the decision were the packet is to be directed is made up when the packet header with routing information is received (the whole packet does not need to be in the node when this decision is made up). In wormhole switching fixed length packets are further divided into so called flits, where *flit* is the unit of message flow control.

1.4.7 Switching in Storage Area Networks

A Storage Area Network (SAN) is a network whose primary purpose is the transfer of data between computer systems and storage elements. A SAN consists of a communication infrastructure, which provides physical connections, and a management layer, which organizes the connec-

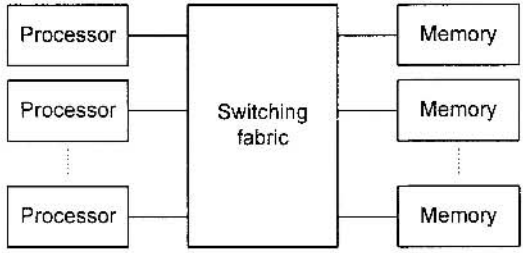


Figure 1.16. A multiprocessor system with shared memory

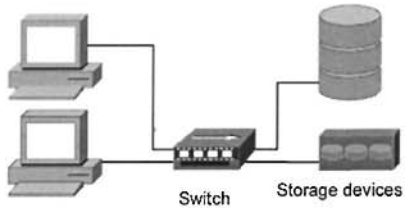


Figure 1.17. A small SAN

tions, storage elements, and computer systems so that data transfer is secure and robust [157]. A SAN allows any-to-any connections across the network using network elements such as routers, gateways, hubs, and switches. These networks are built by enterprises having medium to large data centers. They eliminate the traditional dedicated connection between a server and DAS (direct access storage). Servers have no longer their own storage devices but are connected by a switch to various types of peripheral and storage devices. This architecture is shown in Fig. 1.17. Different standards were developed for this purpose by companies, like ESCON (enterprise serial connection), Fiber Channel, or HIPPI (high performance parallel interface). First SAN operated within a building or campus, but today SANs operate over a wider area like metropolitan or even long-haul networks and operate at bit rates ranging from 200 Mbps to 1 Gbps. This architecture spread over a wider area is shown in Fig. 1.18.

The channel set up through a SAN makes that applications see storage devices attached to the SAN as if they are locally attached storage [47]. A SAN supports direct, high speed transfers between servers and storage devices in the following ways: server to storage, server to server, and storage to storage [158]. Switches used in SANs are usually divided into two types, depending on their port number: directors and fabric

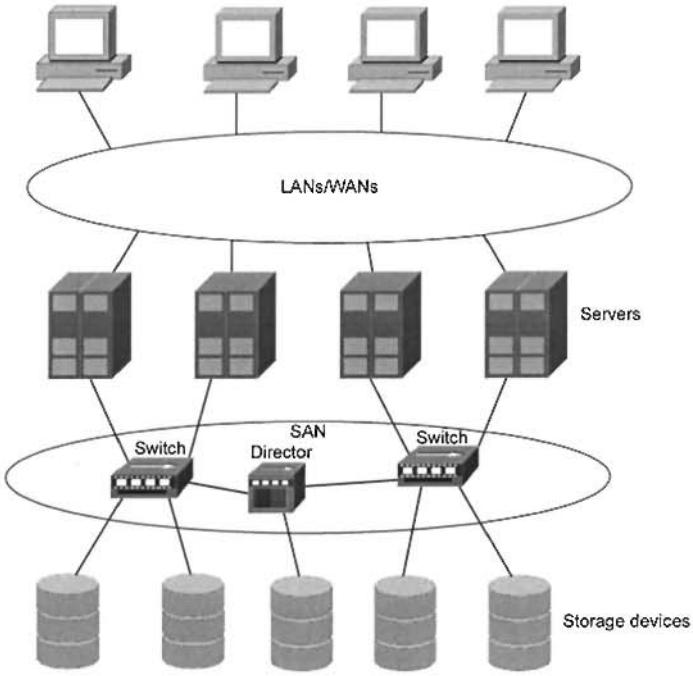


Figure 1.18. A wide SAN

switches. Directors are usually devices of greater capacity, with 100 or more ports, and they're really designed to run in the core of the data center. The important features of directors are their high availability and high scalability. Fabric switches have lower port counts than directors, and their main design point is cost. Fabric switches are used in smaller SANs, in small or medium-sized enterprises, or at the edge of the data center.

1.5 The Contents of the Book

Many architectures of switching fabrics were proposed in the literature. Description of all these architectures, their principles of operation and characteristics, are far beyond the scope of one book. When designing a switching fabric, many designing criteria should be considered. Some of these criteria are common to all switching fabrics, other depends on the technology used or switching fabric's application.

One measure which is used to compare the switching fabric is their cost. This cost depends on the number of required switches, but other factors like packaging, ease of fabrication and control should also be

taken into account. Scalability, expandability, and reliability are also important characteristics which are considered when designing switching fabrics. An important characteristic in photonic switching is losses introduced by connecting paths and difference between maximal and minimal losses of different connecting paths. This difference is often expressed in the difference between the maximum and minimum number of switching elements in the optical path of different input terminal and output terminal pairs. When switch fabric is integrated on a single substrate, waveguides connecting switching elements may intersect between themselves. The number of such waveguide crossovers is also the measure considered in the switch fabric design and should be minimized since they introduce additional power losses and crosstalk. In packet switching a switching fabric should provide bounded delay and small cell or packet loss probability while achieving a maximum throughput close to 100%. The switching fabric should also provide a correct packet sequence at the output ports.

This book covers only several switching fabric architectures and concentrates on their combinatorial properties. Properties like rearrangebility or nonblockingness are connected with control algorithms used for finding a connecting path through a switching fabric. Some control algorithms used in architectures discussed in this book are also described. Only few books, to the author's knowledge, were devoted entirely to mathematical theory of switching fabrics. The first book, published in 1965, is written by V. E. Beneš [9]. Other books covering this subject were written by F.K. Hwang [56], A. Pattavina [138], and S.-Y. R. Li [99], and were published in 1998, 1998 and 2001, respectively. Some other books contains collections of papers concerning combinatorial properties of different switching fabrics [37, 38].

In this book some known results concerning nonblockingness and rearrangeability of switching fabrics are surveyed, but it also contains new results published recently in scientific journals and conference proceedings. Many of these results were obtained by the author during his own research studies and when supervising PhD students.

The organization of the book is as follows. In Chapter 2 the main terminology used in the switching theory and some classifications of switching fabrics are introduced. This classification is done using different criteria. Many other criteria, not covered in this Chapter, may be also proposed and considered, but those used in this Chapter introduce the main terminology which is later on used in this book. Connection types and connection models are also introduced. Graph representations of switching fabrics and connections sets are described at the end of this Chapter.

Chapter 3 is devoted to architectures and control of single path and standard path switching fabrics. First the crossbar architecture is described. Then triangular switches composed of crosspoints and 2×2 switching elements are discussed. Control algorithms for these architectures are also given. Tree-type and one-stage switching fabrics are considered next. Finally, banyan-type switching fabrics are discussed. The architecture, its control and graph representation are presented. Some examples of practical implementations of switching fabrics considered in this Chapter are given at the end.

In Chapter 4 two-stage switching fabrics are covered. Short remarks about two-stage two-sided switching fabrics are first given. The main part of this Chapter is devoted to one-sided two-stage switching fabrics composed of triangular switches. Space-division, time-division and multirate switching fabrics are considered. For each of them strict-sense nonblocking, wide-sense nonblocking, and rearrangeable conditions are discussed.

Organization of Chapter 5 is similar to Chapter 3, but two-sided three-stage switching fabrics are considered. First, different path searching algorithms are introduced for both unicast and multicast connections. Then strict-sense and wide-sense nonblocking conditions are discussed. Rearrangeability of these switching fabrics is covered next and several rearranging algorithms are presented. Finally repackable switching fabrics are examined. The Chapter ends with some remarks about practical implementations of three-stage switching fabrics.

Chapter 6 focuses on vertically replicated baseline switching fabrics, extended baseline switching fabrics, and vertically replicated baseline switching fabrics. The general architecture of these switching fabrics and basic terminology used in this chapter is introduced at the beginning, followed by the description of several control algorithms for unicast and multicast connections. Then combinatorial properties of these switching fabrics under differen switching and connection models are considered.

Chapter 2

CLASSIFICATION AND TAXONOMY

2.1 Switching Elements and Switching Fabrics

In this chapter we introduce some basic terminology which will be used later on in this book. In general, a switching function means to make connections between a given set of terminals. The simplest element which can realize this function is called a crosspoint. It connects one input with one output and is usually placed in the crossing point of lines representing an input and an output. A crosspoint is represented by a circle, as shown in Fig. 2.1a. It can be in one of two states: on or off. When the crosspoint is in the on state, the signal is transmitted from the input to the output (Fig. 2.1c). In the off state the signal is not passed to the output. Different elements can be used as crosspoints, for example electronic gates, semiconductor optical amplifiers or micromirrors [65, 177, 142, 110, 136].

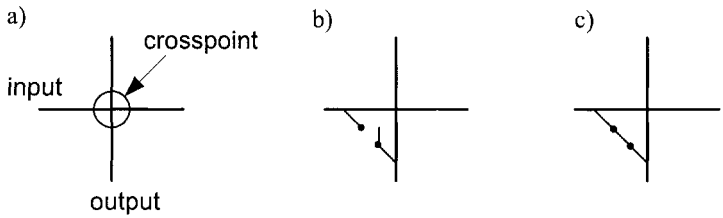


Figure 2.1. A crosspoint; a) symbol, b) off state, c) on state

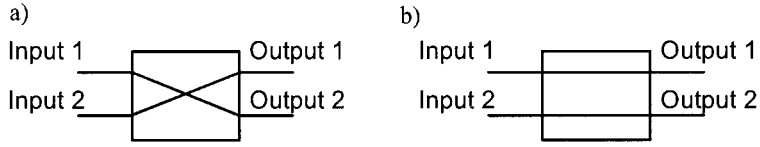


Figure 2.2. A directional coupler in the *cross* (a) and *bar* (b) states

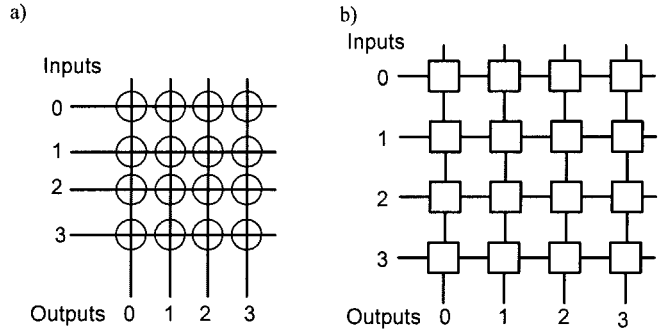


Figure 2.3. The 4×4 switch constructed from crosspoints (a) and 2×2 BSEs (b)

Another basic element which can realize the switching function is called a basic switching element (BSE). Unlike the crosspoint, BSE can connect more inputs and outputs. A directional coupler is one example of the BSE [48, 65]. It has two inputs and two outputs and it can be also in one of two states called *cross* and *bar*. In the *cross* state input 1 is connected with output 2 and input 2 is connected with output 1. In the *bar* state inputs 1 and 2 are connected with outputs 1 and 2, respectively (see Fig. 2.2). Directional couplers are used in optical switching, but there are also BSEs with two inputs and two outputs used in electronic ATM or IP switching. Some BSEs may have only one input and several outputs, one output and several inputs, or several inputs and outputs [41, 69, 168, 49, 21].

Crosspoints or BSEs are used to construct switching devices of greater capacities called switches. A switch with n inputs and m outputs is denoted by $n \times m$ switch. A switch has the limited capacity and is produced as an integrated circuit or is implemented on one printed board. Examples of 4×4 switches composed of crosspoints and 2×2 BSEs are shown in Fig. 2.3a and b, respectively. These switches are also often referred to as switching matrices or crossbars, since they use a crossbar architecture (this architecture will be discussed in more details in chapter 3.2).

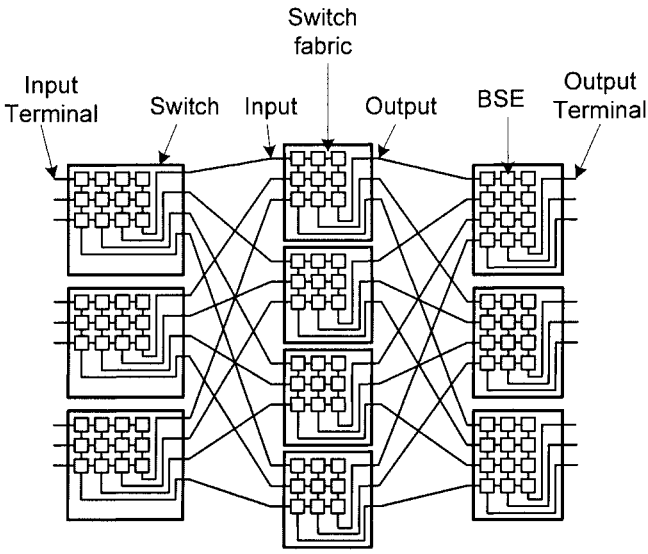


Figure 2.4. An example of 9×9 switching fabric

A capacity of switches expressed in the number of inputs and outputs is limited due to technological constraints (limited number of elements which can be placed on a chip or printed board, limited number of pins, etc.). When a switching device of greater capacity is needed more switches are to be connected between themselves. Such a device is called *a switching fabric*. A switching fabric with N_1 input terminals and N_2 output terminals has a capacity of $N_1 \times N_2$. Input and output terminals are also referred to as *inlets* and *outlets*, respectively. An example of 9×9 switching fabric composed of 3×4, 3×3, and 4×3 switches is shown in Fig. 2.4. Inputs of 3×4 switches constitutes input terminals of the switching fabric while outputs of 4×3 switches forms output terminals of the switching fabric. Other inputs and outputs are connected between themselves one to one by means of interstage links.

It should be noted that sometimes the terminology used in the literature can be found rather fuzzy. For instance 2×2 BSE may be a switch which is used to construct a switching fabric of greater capacity. This switching fabric may be implemented in a printed board, and more such printed boards may be connected between themselves to form another switching fabric. When it is necessary, in the former case we will refer to the term a switch fabric. However, architectures considered in this book may be used either as a switch fabric or a switching fabric.

2.2 Classification of Switching Fabrics

Switching fabrics can be classified into different categories, depending on the criteria being used. Some of these criteria and corresponding classes of switching fabrics are given in Table 2.1. Some of them were already discussed in chapter 1 while others will be discussed in this and the following sections in more details.

Technology used. Depending on the technology used nowadays switching fabrics can be divided into two major classes: electronic switching fabrics and photonic switching fabrics (called also optical switching fabrics). These and earlier technologies used in switching were already discussed in section 1.2 of chapter 1.

Relationships between input and output terminals sets. Using this criterion switching fabrics can be categorized into two categories: two-sided switching fabrics and one-sided switching fabrics. In *two-sided switching fabrics* sets of the input and the output terminals are disjoint. A connection is always set up from input terminals to output terminals. An example of 4×4 two-sided switching fabric is shown in Fig. 2.5a. In many practical applications any terminal may request a connection to any other terminal, i.e., any terminal could be either an input terminal or an output terminal. Such switching fabrics are called *one-sided switching fabrics* (also referred to as *folded switching fabrics*). They can be obtained in two ways. The first approach, considered by Clos [23], is to use so-called triangular switches. An example of 4×4 one-sided switching fabric composed of triangular switches is shown in Fig. 2.5b. In this switching fabric connections between terminals belonging to the same first stage switch (for example between terminals 0 and 1) are set up inside this switch (switch 1 of stage 1), without using other switches. In the second approach, a one-sided switching fabric is obtained from a two-sided one by looping input and output terminals having the same numbers. A one-sided switching fabric obtained in this way from that presented in Fig. 2.5a is shown in Fig. 2.5c. A mixed switching fabric may be also constructed. Such mixed architecture contains N_1 input terminals, N_2 output terminals, and N_3 terminals, each of which may be either an input or an output terminal.

Criterium	Switching fabrics classes
Technology used	– electromechanical switching fabrics – electronic switching fabrics – photonic switching fabrics
Relationships between inlets and outlets sets	– one-sided switching fabrics – two-sided (folded) switching fabrics – mixed switching fabrics
Number of stages	– single-stage (one-stage) switching fabrics – multistage switching fabrics (two-stage, three-stage, ...)
Signal transmission directions	– unidirectional switching fabrics – bidirectional switching fabrics
Number of inputs and outputs	– switching fabrics with compression – switching fabrics with expansion – switching fabrics with traffic distribution
Separation of data paths	– space-division switching fabrics – time-division switching fabrics – wavelength-division switching fabrics – code-division switching fabrics
Number of paths between input and output terminals pair	– single-path switching fabrics – multi-path switching fabrics
Number of links between switches in successive stages	– fully connected switching fabrics – partially connected switching fabrics
Output accessibility	– fully accessible switching fabrics – not fully accessible switching fabrics
Combinatorial properties	– strict-sense nonblocking switching fabrics – wide-sense nonblocking switching fabrics – rearrangeable switching fabrics – repackable switching fabrics – blocking switching fabrics
Number of input and output terminals which take part in a connection	– unicast (point-to-point) switching fabrics – multicast switching fabrics – broadcast switching fabrics – multiconnection switching fabrics
Bandwidth occupied by a connection	– single-rate switching fabrics – multi-rate switching fabrics – multi-channel switching fabrics

Table 2.1. Criteria and classes of switching fabrics

Number of stages. The capacity of a single switching element is limited due to the technological constraints. Therefore, many such elements are used to construct a switching fabric of greater capacity. The switches are usually arranged in stages. Outputs of switches in stage i, $1 \leqslant i \leqslant s-1$, s is the number of stages, are connected to inputs of

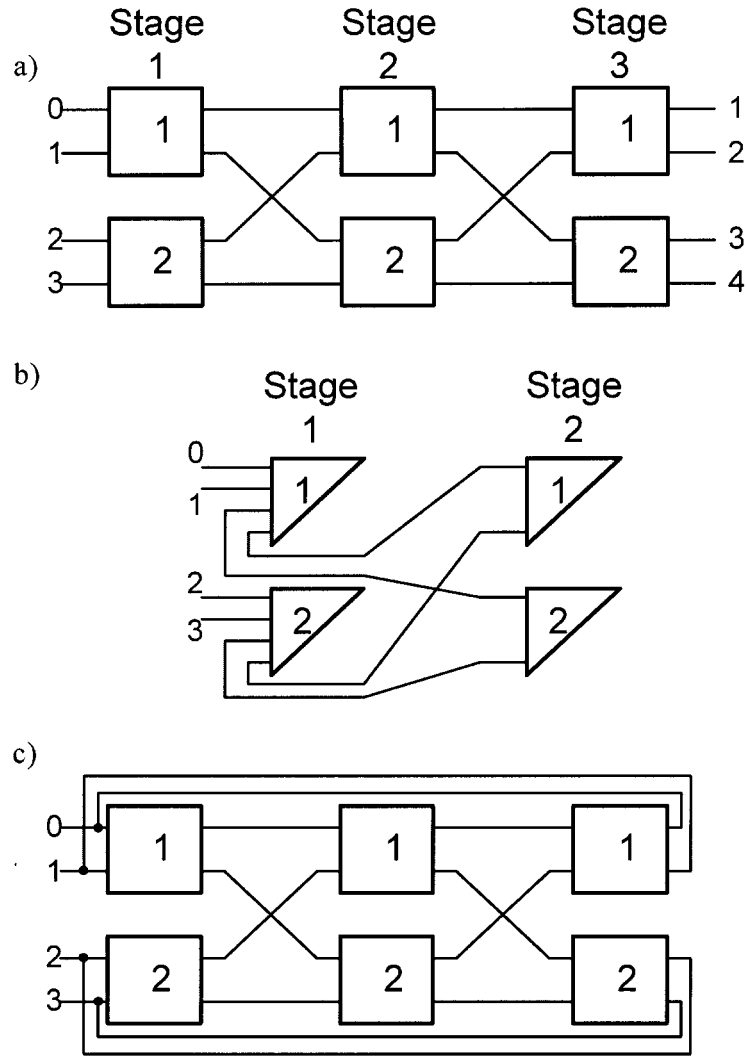


Figure 2.5. Examples of a two-sided switching fabric (a), a one-sided switching fabric composed of triangular switches (b) and with loops (c)

switches in stage $i + 1$ by means of interstage links. Inputs of switches in stage 1 constitute switching fabric's input terminals, while outputs of switches in stage s constitute its output terminals. Depending on the number of stages the respective switching fabric is called a two-stage, a three-stage, or *a multistage switching fabric* in general. The switching fabric of Fig. 2.5a is a three-stage switching fabric, while the switching fabric of Fig. 2.5b is a two-stage switching fabric.

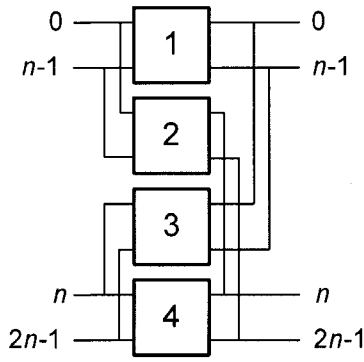


Figure 2.6. Two-sided $2N \times 2N$ one-stage switching fabric

In case $s = 1$ we speak about *a one-stage switching fabric*. If it is composed of more than one switch, inputs and outputs of these switches are connected in parallel. The examples of two-sided $2n \times 2n$ one-stage switching fabric composed of $n \times n$ switches is shown in Fig. 2.6. It is composed of four $n \times n$ switches. Inputs of switches 1 and 2 (3 and 4) are connected in parallel, while outputs of switches 1 and 3 (2 and 4) are connected between themselves in the similar way.

Signal transmission directions. Depending on the technology used a switching fabric may be able to send signals only form input to output terminals, or in both directions. We refer to these switching fabrics as *unidirectional* or *bidirectional switching fabrics*, respectively. For instance in switches based on directional couplers or MEMS switches using mirrors, light signals can be sent in either direction (Fig. 2.7a). Digital switches based on digital gates and memories can transport signals only in one direction. However, in many practical applications, bidirectional transmission is required. In unidirectional switching fabrics two-way communication can be ensured using two approaches. In one approach two unidirectional switching fabrics are used, each for one direction. Such switching fabrics are referred to as *four-wire switching fabrics* (Fig. 2.7b). In another approach two independent connections, each for one direction, are established in a common switching fabric, as it is shown in Fig. 2.7c. Such switching fabrics are called *two-wire switching fabrics*.

Number of input and output terminals. In general, a two-sided switching fabric may have N_1 input terminals and N_2 output terminals. When $N_1 > N_2$ a switching fabric is called *a switching fabric with con-*

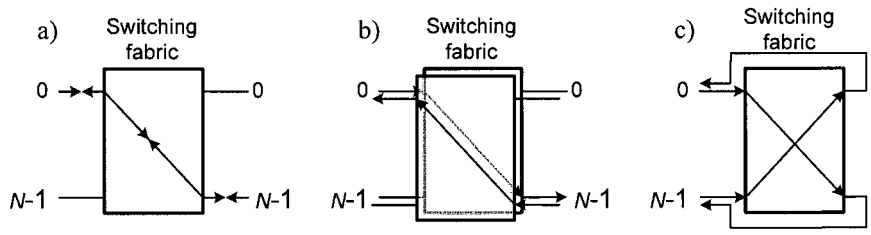


Figure 2.7. Two-way communication in two-sided bidirectional (a), unidirectional four-wire (b) and unidirectional two-wire (c) switching fabrics

centration or simply *a concentrator*. For $N_2 > N_1$ we say about *a switching fabric with expansion* (or *an expander*). Finally, when $N_1 = N_2$ a switching fabric is called *a distribution switching fabric* or *a distributor*. In the case of one-sided switching fabrics we say only about terminals and there is no such relation between the number of input and output terminals.

Number of paths between input and output terminals. To connect two terminals between themselves a connecting path (or simply a path) is set up through a switching fabric. This path consists of an input terminal, requested output terminal, respective switches and interstage links. In general, a switching fabric may provide more than one connecting path for a given pair of input and output terminals. We say that this is *a multi-path switching fabric*. When there is only one path provided, the switching fabric is *a single-path switching fabric*. Switching fabrics presented in Fig. 2.5 are multi-path switching fabrics. Examples of one-sided and two-sided single-path switching fabrics are shown in Fig. 2.8.

Separation of data paths. Different multiplexing methods are used in networks to use available resources efficiently. In the simplest approach, signals from different users are sent using separate links. This approach is called space-division multiplexing (SDM). But links may be also shared in time, where data from different users are sent in different time intervals called time slots, or in wavelengths (or frequency), when data are sent through the same fiber using different wavelengths. The former case is used in time-division multiplexing systems (TDM) like PCM or SONET/SDH systems, why the later case is called wavelength-division multiplexing (WDM). Code-division multiplexing (CDM) is also used in the network, where data from different users are multiplexed in the same link using orthogonal codes.

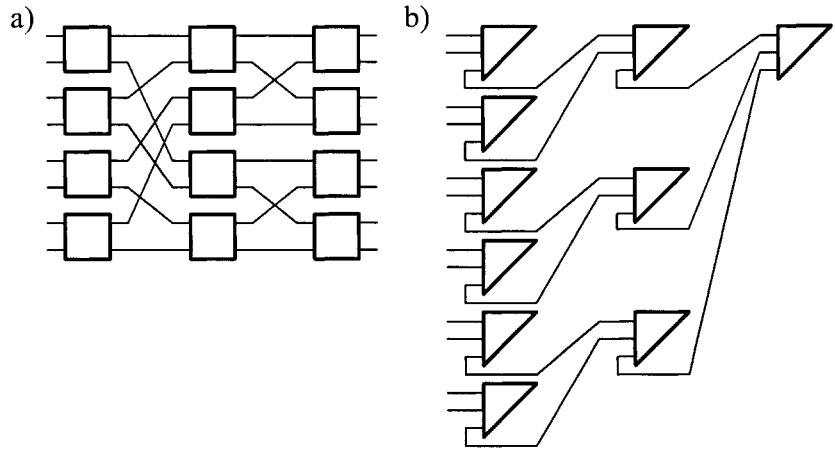


Figure 2.8. Single-path two-sided (a) and one-sided (b) switching fabrics

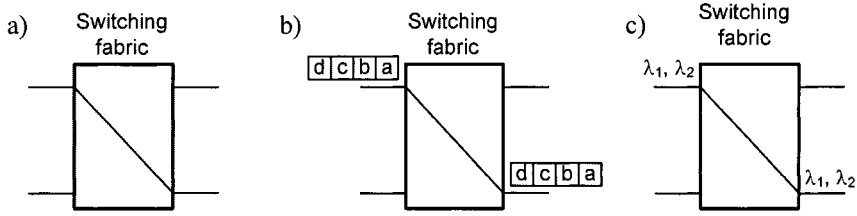


Figure 2.9. Space-division switching fabrics with SDM (a), TDM (b), and WDM (c) transmission

Similarly, connecting paths provided by a switching fabric can be separated in different domains. In *space-division switching fabrics* connecting paths are set up between the switching fabric's terminals. Whole data from any input terminal is transferred to a connected output terminal, as shown in Fig. 2.9a. These data may be time or wavelength multiplexed as shown in Figures 2.9b and c, respectively. The second case is also called a *fiber switching*, where a switching fabric switches all the wavelength of an incoming fiber to an outgoing fiber.

In *time-division switching fabrics* any time slot of an input TDM link can be connected to any time slot of an output TDM link. A switch realizing this function is called a *time switch*. An example of time-division switching in a time switch with one input and one output, each carrying four time slots, is shown in Fig. 2.10a. In practical implementations a switching fabric switches more links and can connect any time slot of

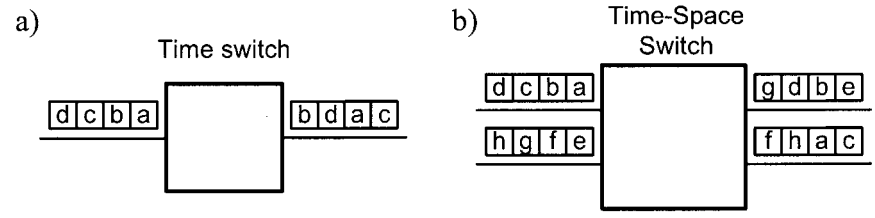


Figure 2.10. Time-division switching (a) and time-space-division switching (b)

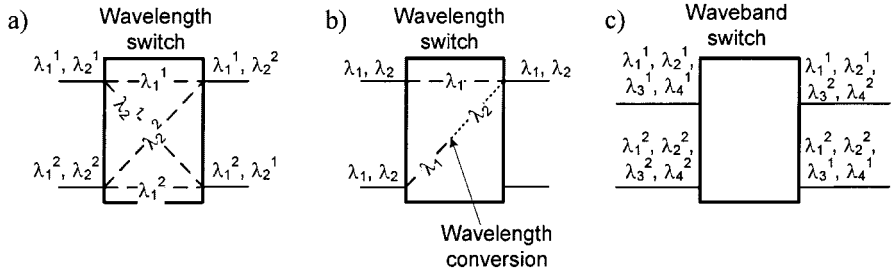


Figure 2.11. Wavelength-division switching fabrics with (a) and without (b) wavelength conversion and a waveband-division switching fabric (c)

any input terminal to any time slot in any output terminal, i.e., they can operate in time and space domains (Fig. 2.10b), however they are also called time-division switching fabrics.

When input links are WDM links it is often needed that any wavelength from any input fiber will be switched on any wavelength on any output fiber. This type of switching is called wavelength-division switching and respective switching fabrics are called *wavelength-division switching fabrics* (Fig. 2.11a). In this type of switching a wavelength conversion may be necessary to ensure full connectivity between wavelength in input and output terminals. This happens when two wavelengths of the same length in two input terminals are to be connected to the same output terminal. In this case one of these wavelengths is to be switched to another wavelength (Fig. 2.11b). In wavelength-division switching another approach is also possible, in which a set of wavelength (called a waveband) on an incoming fiber is switched to an outgoing fiber. This type of switching is called *waveband switching*. An example is shown in Fig. 2.11c, where wavelength λ_1 and λ_2 are in one waveband, and λ_3 and λ_4 are in the second waveband.

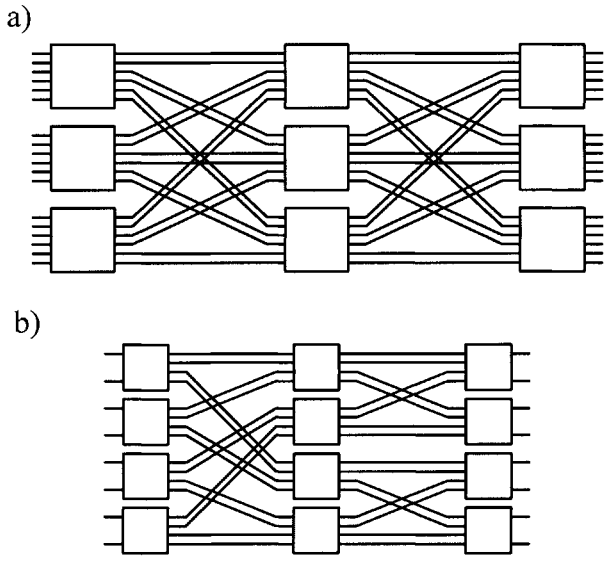


Figure 2.12. Fully connected (a) and partially connected (b) 2-dilated switching fabrics

In *code-division switching* a switching fabric switches an individual code from input terminal to a code on output terminal.

Number of links between switches in successive stages. Depending on the number of links between switches in successive stages switching fabrics can be categorized into two categories: *partially connected switching fabrics* and *fully connected switching fabrics*. In partially connected switching fabrics each switch in stage i is connected to some but not all switches in stage $i + 1$. When each switch of stage i is connected to every switch in stage $i + 1$, we say that a switching fabric is fully connected. Additionally, when two switches are connected by v links we say that the switching fabric is v-dilated. When $v = 1$ a switching fabric is 1-dilated. Switching fabrics presented in Fig. 2.5 are fully connected 1-dilated switching fabrics, while that presented in Fig. 2.8 are partially connected 1-dilated switching fabrics. Examples of 2-dilated fully connected and partially connected switching fabrics are shown in Fig. 2.12a and b, respectively. In the rest of this book we will assume that a switching fabric is 1-dilated if it is not clearly stated.

Output accessibility. If in a switching fabric each output terminal can be reached from each output terminal we say that it is *a fully ac-*

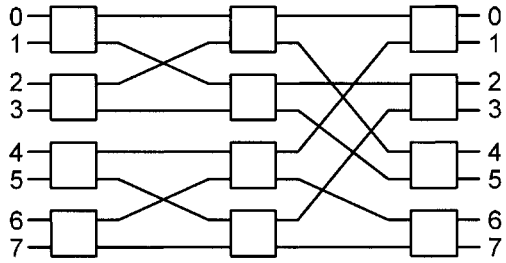


Figure 2.13. An example of not fully accessible switching fabric

cessible switching fabric. When from some input terminals it is not possible to set up a connecting path to any of the output terminals at all we say that this switching fabric is *not fully accessible.* Switching fabrics presented in Fig. 2.12 are fully accessible. An example of not fully accessible switching fabric is presented in Fig. 2.13. In this switching fabric, for instance, it is not possible to set up a connection between input terminal 1 and output terminal 7.

Blocking states. A switching fabric provides a connecting path between terminals. At a given time many connecting paths may be set up concurrently in a switching fabric, and they use different switches and crosspoints. We say that a switching fabric is in a certain state. We can also say that the state is a setting of switches. If in the given state it is not possible to connect an idle input terminal to an idle output terminal, but there are other states in which this connection is possible, than this state is called *a blocking state.* An example of the blocking state in the three-stage 4×4 switching fabric is shown in Fig. 2.14. Connections from input terminal 3 to output terminal 0 and from input terminal 0 to output terminal 3 are shown in bold lines. The connection between input terminal 1 and output terminal 1 is blocked, since it cannot be set up through either of the second stage switches. However, it would be possible to set up this connection, when, for instance, the connection between input terminal 3 and output terminal 0 had been set up through the first center stage switch.

Depending of the occurrence of blocking states, switching fabrics can be categorized into two main categories: *blocking* and *nonblocking switching fabrics.* In a blocking switching fabric the blocking states cannot be omitted, i.e., some combinations of connections cannot be realized. In a nonblocking switching fabric every combination of connections can be established. In other words, any permutation between input terminals

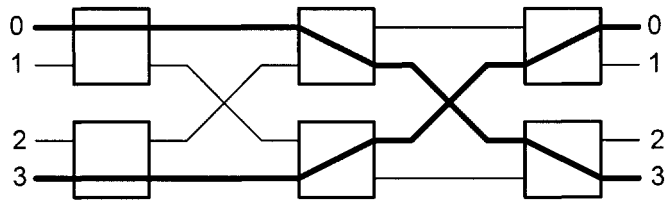


Figure 2.14. A blocking state in the switching fabric

and output terminals can be realized, and we say about combinatorial properties of switching fabrics. Nonblocking switching fabrics can be further divided into four classes: strict-sense nonblocking, wide-sense nonblocking, rearrangeably nonblocking, and repackably nonblocking.

- *Strict-sense nonblocking switching fabrics* (SNB). A switching fabric is strict-sense nonblocking if it can always connect each idle input terminal to an arbitrary idle output terminal independent of its current state and no matter how connecting paths were selected for the existing connections.

- *Wide-sense nonblocking switching fabrics* (WNB). A switching fabric is wide-sense nonblocking if it can also connect each idle input terminal to an arbitrary idle output terminal independent of its current state provided that some given path selection algorithm was used for setting up connections.

- *Rearrangeably nonblocking switching fabrics* (RNB) or *rearrangeable switching fabrics*. A switching fabric is rearrangeable if it can also connected a pair of idle input and output terminals, however, it may be necessary to move existing connections to alternate connecting paths.

- *Repackably nonblocking switching fabrics* (PNB) or *repackable switching fabrics*. A switching fabric is repackable if blocking states can be omitted by re-routing some of existing connecting paths using a repacking algorithm but in contrast to rearrangeable switching fabrics, they are executed after one of existing calls is terminated.

In some architectures of multi-path switching fabrics, only one path, from all paths between a given input and output terminals pair, is permitted to route a given connection, in order to save other paths for other connections. This path is called *a standard path*. If a switching fabric provides such standard paths for each input-output terminals pair and

these paths are disjoint for connections which may be set up in any state of the switching fabric, than such switching fabric is called a standard path switching fabric. This class of switching fabrics may be considered as WNB switching fabrics in which control algorithm use always a standard path allowed for a given connection.

Many other criteria may be used to classify switching fabrics. Some of them are based on connection types realized in switching fabrics. They will be the subject of the next section.

2.3 Connection Types

Up till know we referred to the connection as a call between one input terminal and one output terminal. In general, more input and output terminals may take part in a connection. We also assumed, that all connections require the same transmission capacity. This is also not always true. Number of terminals in a connection and a transmission bandwidth required by a connection are used to categorized connections into different classes which will be now discussed.

2.3.1 Connection Set-up Models

Connection set-up models refer to the time connections may arrive to a switching system. We can speak about

- one-by-one connection model, also called one-at-a-time connection model, and
- simultaneous connection model.

In the *one-by-one connection model* requests arrive to the system one-by-one. In this model there is only one connection being set up at a time in a switching fabric. This model is applied for instance in telephone exchanges.

In the *simultaneous connection model* requests arrive to the system simultaneously. In this model a set of compatible connections is being set up at the same time. The set of compatible connections means that one input terminal requests a connection with exactly one output terminal, and one output terminal is requested by exactly one input terminal. Such set of calls is also called *a permutation* or *a maximal assignment* if there is a request in each input terminal, and it is usually written as:

$$\Pi = \begin{pmatrix} 0 & 1 & 2 & \cdots & N-1 \\ \pi(0) & \pi(1) & \pi(2) & \cdots & \pi(N-1) \end{pmatrix}. \tag{2.1}$$

This means that input terminal 0 is to be connected to output terminal $\pi(0)$, input terminal 1 is to be connected to output terminal $\pi(1)$, etc.

Analogically, inverse permutation Π^{-1} denotes that output terminal 0 is to be connected to input terminal $\pi^{-1}(0)$, output terminal 1 is to be connected to input terminal $\pi^{-1}(1)$, and so forth. The simultaneous connection model is applied for instance in multiprocessor systems, where a set of microprocessors may request simultaneously access to the set of memories, or in synchronous packet switches, in which conflict free packets from selected input terminals are to be transferred in the same time slot to requested output terminals.

2.3.2 Unicast, Multicast, and Broadcast Connections

In future communication networks, apart from point-to-point connections, many services, for instance video-conference, video-distribution, multi-party communications, etc., will require connections between more than two users [107, 97, 50]. This can be realized by setting up separate connections for each pair of users, however, this results in increasing the volume of data sent in the network and the bandwidth used in transmission links. To reduce this volume of data connections are split inside the network. The multicast communication is desirable also in IP networks to support group communication, since it not only saves bandwidth, but also reduces host or server processing load [109]. Therefore, it is also expected that a switching fabric will be able of setting up connections between greater number of terminals. Depending on the number of input and output terminals used in a connection, connections can be divided into following classes:

- *Unicast connections.* A unicast connection, called also *a point-to-point connection* is a connection between one input terminal and one output terminal.

- *Multicast connections.* A multicast connection, called also *a point-to-multipoint* connection, is a connection between one input terminal and a set of output terminals, where the cardinality of this set is greater than 1.

- *Broadcast connections.* A broadcast connection is the special case of a mulitcast connection, in which one input terminal is to be connected to all output terminals.

- *Multiconnections.* A multiconnection is a connection between a set of input terminals and a set of output terminals, the cardinality of these sets is greater than 1, and all input-output pairs of these sets are connected between themselves.

- *Multipoint-to-point connections.* A multipoint-to-point connection is a connection between more than one input terminals and one output terminal.

The respective classification can be also used for switching fabrics, depending on the type of connections which can be set up. In practical solutions only unicast, multicast or broadcast connections are considered. In *multicast* and *broadcast switching fabrics* multicast or broadcast connections can be set up. Data from an input terminal are copied inside the switching fabric and sent to the appropriate number of connected output terminals.

Different types of connections result in the combinatorial properties of switching fabrics. Switching fabrics which are nonblocking for unicast connections usually become blocking when multicast connections may be also set up. In this book only combinatorial properties of switching fabrics with unicast, multicast, and broadcast connections will be considered. Some results for switching fabrics with multiconnections were considered in [53, 51, 61].

2.3.3 Single-rate and Multirate Connections

Connections may occupy different bandwidths in transmission links. In the first switching fabrics connections for telephone calls were provided and such connection occupied the whole bandwidth available in a transmission link. The development of time-division switching fabrics enables to share the transmission capacity among a large number of voice circuits. Each connection occupied one time slot of a TDM system and of a time-division switching fabric. The idea of integrating different services which require transmission of audio, data, image, and video in one telecommunication network changes requirements for switching fabrics. As different media demand for a broad range of bandwidths, each connection is associated with its demanded rate of bandwidth. These bit rates may change from a few kb/s up to 600 Mb/s in case of broadband services (videophone, high-speed data transmission, HDTV). Therefore, the future exchanges will have to be capable of switching connections with different bit rates. Two approaches were proposed to realize such connections in the network. One approach was based on synchronous TDM systems and connections could occupy different number of time slots. An alternative approach was derived from packet switching and is called *Asynchronous Transfer Mode* or ATM [116]. In this approach each connection is identified by a virtual channel identifier. This identifier is placed in the header of each cell. Virtual channels may occupy different bandwidths in a transmission link.

Depending on the bandwidth required, connections can be divided into the following classes.

- *Single-rate connections.* Each connection occupies the same bandwidth in a link. It may be the whole link as in the space-division switching or one time slot as in the time-division switching.
- *Multi-slot connections.* Connections may occupy a different number of time slots in one TDM link. These connections are also referred to as *multi-channel connections.*
- *Multirate connections.* Connections may occupy different bandwidths of a link.

Multirate connections can be further divided into discrete and continuous bandwidth cases [22, 104].

- *Discrete bandwidth.* In the discrete bandwidth case it is assumed that there is a finite number of distinct rates and the smallest rate divides all other rates.
- *Continuous bandwidth.* In the continuous bandwidth case connections may occupy any fraction of a link's transmission capacity.

The multi-slot connections are often considered as the discrete bandwidth case but it is true only if these connections can occupy any time slots in TDM links. This is not always true. Models where multi-slot connections can occupy only consecutive time slots were also considered [11, 143, 156]. Multi-slot switching fabrics were firstly considered by Niestegge [120]. The model for multirate switching fabrics was proposed by Melen and Turner [114]. These models will be later described in this chapter and will be further used in this book. Some researchers considered also multirate switching fabrics under the k-rate connection model, where k is an integer and it specifies the number of distinct connection rates. In the 1-rate model ($k = 1$) every request has the same rate ω. When $\omega = 1$ this model corresponds to space-division switching. When $\omega = 1/f$, the model corresponds to time-division switching with links carrying f time slots each.

2.4 Control Algorithms

When a new connection is to be set up, the controller has to find a connecting path in a switching fabric, check whether it is available (i.e., some elements are not occupied by other connections), issue respective control signals to change state of switching elements or crosspoints, and update the current state of a switching fabric. These tasks are performed

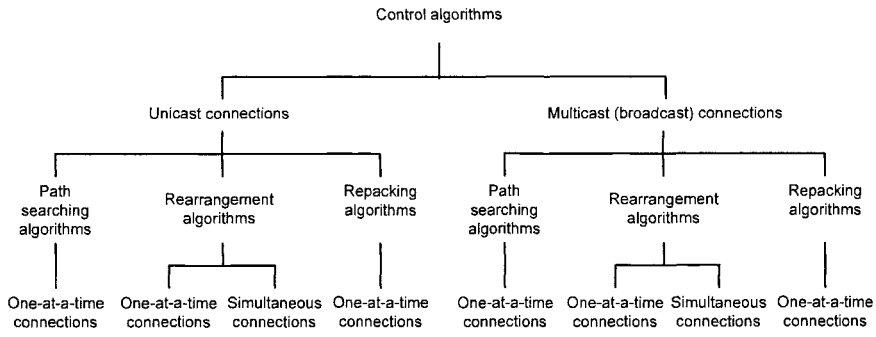


Figure 2.15. Classification of control algorithms in multi-path switching fabrics

by control algorithms mainly. The state of a switching fabric is usually stored in a table containing information about all connecting paths already set up. When any one of existing paths is to be disconnected, information about it is deleted from the table and the path in question is removed by resetting the switching fabric configuration appropriately. Apart from switches and crosspoints, there may be also a need to control some additional elements, for instance optical amplifiers in photonic switching fabrics.

The control algorithm used for finding a path depends on connection type such as unicast, multicast or broadcast. Numerous control algorithms have been proposed for different connection types and kinds of switching fabrics. These algorithms, the topology of the switching fabric, and the type of connection are usually related. Therefore, algorithms designed for one topology usually cannot be used in other topologies. However, they can be used in switching fabrics made in different technologies, so the same algorithm may be used for a switching fabric of a given topology regardless of whether it is fabricated using electronic or photonic technology. Special requirements (like crosstalk reduction) may result in elaborating more sophisticated algorithms, which are not necessary when another technology is used [111, 165].

Control algorithms in multi-path switching fabrics can be divided into three major groups (Fig. 2.15). *Path searching algorithms* are used for finding a connecting path through a switching fabric for one connection at a time. Depending on combinatorial properties of a switch fabric, the given algorithm may always lead to success (i.e., a connecting path will always be found), or not. Another group of control algorithms is called *rearrangement algorithms*. These algorithms can be used when a path searching algorithm fails. Their task is to find connecting paths which can be re-routed to unblock a new connection. Some of them are de-

signed to find a connecting path for one connection at a time, others are designed to find connecting paths for all new connections simultaneously. In call repacking some connecting paths are also re-routed in a switch fabric but in contrast to rearrangements, they are executed after one of existing connections is terminated. The role of *repacking algorithms* is to "pack" existing calls more efficiently, and thereby prevent switch fabrics from being in a blocking state when a new connection arrives. The introduced classification is true for both unicast and multicast connections; however, respective algorithms will be different.

2.5 Notation, Terminology, and Models

2.5.1 Switching Fabrics and Connections

The general notation concerning switching fabrics and connections, which will be used in the rest of this book, will be given here. More specific notation will be introduced as they arise. In an asymmetrical two-sided switching fabric input terminals will be numbered $0, 1, \ldots, N_1 - 1$, and output terminals will be numbered $0, 1, \ldots, N_2 - 1$, from top to bottom. In a symmetrical two-sided switching fabric number of input (output) terminals will be denoted by N, $N = N_1 = N_2$. In a one-sided switching fabric terminals will be also numbered $0, 1, \ldots, N - 1$, from top to bottom. Stages in a multistage switching fabric will be numbered from left to right starting with 1. Switches in each stage will be numbered from top to bottom also starting from 1. Switch i of the first stage will be denoted by I_i and switch j of the last stage in a two-sided switching fabric will be denoted by O_j.

A unicast connection between input terminal x and output terminal y (or terminals x and y in a one-sided switching fabric) will be denoted by $\langle x, y \rangle$. When x is an input of I_i and y is an output of O_j, and it is sufficient to describe connection $\langle x, y \rangle$ using only symbols of outer stage switches, than this connection will be denoted by (I_i, O_j). When any terminal of switch I_i is connected with any terminal of switch I_j in a one-sided switching fabric, than this connection will be denoted by (I_i, I_j).

A multicast connection from input terminal x to output terminals $y_0, y_1, \ldots, y_k$ will be denoted by $\langle x, \mathbb{Y} \rangle$, where $\mathbb{Y} = \{y_0, y_1, \ldots, y_k\}$, $0 \leqslant x \leqslant N_1 - 1$, $0 \leqslant y_0, y_1, \ldots, y_k \leqslant N_2 - 1$. This $\langle x, \mathbb{Y} \rangle$ is a point-to-point connection if and only if $|\mathbb{Y}| = 1$, it is a multicast connection when $1 < |\mathbb{Y}| < N_2 - 1$, and it is a broadcast connection when $|\mathbb{Y}| = N_2 - 1$, where $|\mathbb{Y}|$ is the cardinality of set $\mathbb{Y}$. For instance, connections $\langle 0, \{0\} \rangle$, $\langle 4, \{2\} \rangle$ and $\langle 16, \{6\} \rangle$ are point-to-point connections, while $\langle 31, \{3, 7\} \rangle$ is a multicast connection.

Multicast connection $\langle x, \mathbb{Y} \rangle$ is also referred to as q-cast connection, where $q = |\mathbb{Y}|$. The number of output terminals which may take part in a q-cast connection may be limited by q_1 and q_2, $1 \leqslant q_1 \leqslant q \leqslant q_2 \leqslant N_2$, where q_1 denotes the minimum number of output terminals and q_2 represents the maximum number of output terminals which can take part in a q-cast connection.

Multicast connections can be also divided depending on whether additional output terminals can be added to an existing multicast connection or not [56]. When a new output terminal may be added to an existing connection a connection is called *the open-end multicast connection.* When such addition is not permitted we refer to it as *the closed-end multicast connection.*

2.5.2 Multi-slot Switching Model

A switching fabric which has to switch connections at varying speed can be designed either for the highest or the lowest speed connections. In the first case the low-speed connections use the same bandwidth as the highest ones and, therefore, it is not economical for the network which carries many low speed connections. In the second case high-speed connections are connected by using more than one time slot. The connection which occupies s slots is called an s-slot connection. These s slots may be switched independently or in parallel [120]. In the first case slots in an s-slot connection are switched independently. They may be transferred through different multiplexed links and different switches, and therefore, they may encounter different propagation delays and they may appear at the output link out of sequence. When s-slot connection is set in parallel, all slots belonging to this connection are assigned to the same interstage links and switches in a multistage switching fabric. In this case it is easier to preserve a time-slot order. Some methods, in which this problem has been solved, are reported in the literature [11, 143, 156]. The idea of independent and parallel switching of 2-slot connection in the three-stage switching fabric is shown in Fig. 2.16. In Fig. 2.16a the 2-slot connections is set up independently, i.e., 2 slots are transferred through different interstage links and different second stage switches (slots are marked in dashed lines). In the second case, these slots use the same second stage switch and the same interstage links (Fig. 2.16b).

Another important problem in multi-slot switching is how idle slots are assigned to a new connection. These slots may be assigned using following assignment algorithms: [8, 89, 156]:

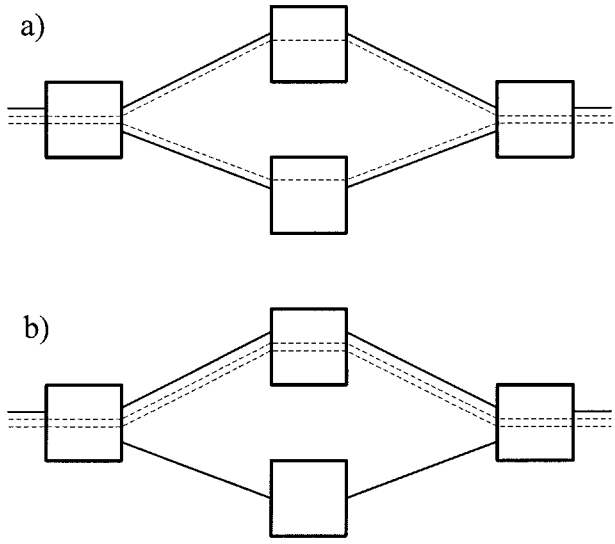


Figure 2.16. Independent (a) and parallel (b) switching of the 2-slot connection

- *Fixed assignment.* Only a fixed group of s adjacent time slots can be assigned to the s-slot connection.
- *Floating assignment.* Any s slots contiguous to each other can be assigned to the s-slot connection.
- *Periodic assignment.* This assignment permits only equal interval assignment, i.e., there are equal intervals between time slots which are assigned to the s-slot connection.
- *Flexible assignment.* This assignment imposes only that all slots must still be within the same interstage link, and the channel order must still be preserved, but the selected channels no longer need to be contiguous. This assignment is also called *a random assignment.*

The example of these different assignments is shown in Fig. 2.17. In the fixed assignment approach slots in a time-division multiplexed link are grouped and only groups are assigned to the s-slot call. For instance slots 1-6, 7-12, 13-15 and 17-19, 20-25, and 26-31 may be reserved for 6-slot connections in the 32 PCM link. In the example of Fig. 2.17a two 6-slot connections use slots 1-6 and 20-25. In the floating assignment a multi-slot connection has also to use slots contiguous to each other, but the first slot used by this connection may by any slot of the link. The two connections considered in the example (Fig. 2.17b) uses slots

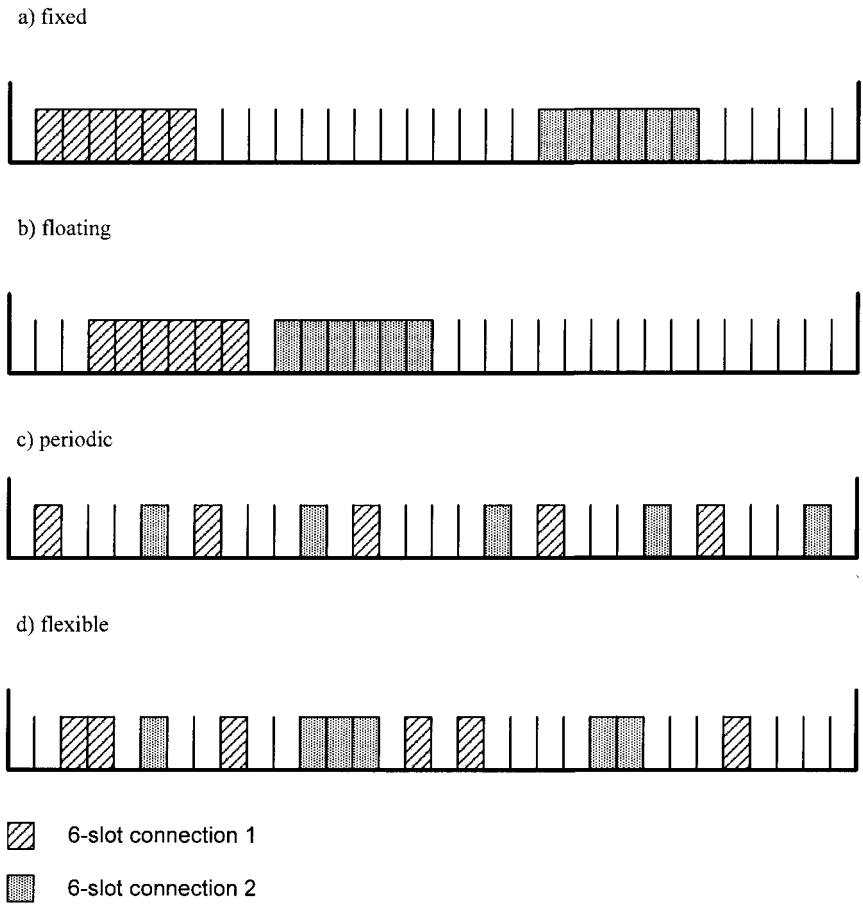


Figure 2.17. Fixed (a), floating (b), periodic (c) and random (d) assignment of time slots for two 6-slot connections in the 32 PCM link

3-8 and 10-15. Periodic and random assignments for these two 6-slot connections are shown in Fig. 2.17c and Fig. 2.17d, respectively.

If a new s-slot connection is to be set up through the switching fabric, the interstage link, which is accessible for this connection has to be found, and then s time slots should be assigned. A space-division switching network is strictly nonblocking if it is always possible to set up a connecting path between its free terminals independently of the already existing connections and the path search algorithm [9, 23]. In the multi-slot switching case we will call the switching network strictly nonblocking if it is possible to set up a new call independently of the path search algorithm and the time-slot assignment. The switching net-

work, which is nonblocking when the special path search algorithm or time slot assignment are used, is called nonblocking in the wide sense.

In the rest of this book the following notation will be used. The s-slot connection between input terminal x and output terminal y (or terminals x and y in a one-sided switching fabric) will be denoted by $\langle x, y, s\rangle$. When it will be sufficient to denote only switches, not terminals, between which connection is to be set up, then this s-slot connection will be denoted by (I_i, O_j, s), where I_i is the first stage switch to which input terminal x is connected to, and O_j is the last stage switch to which output terminal y is connected to (in the case of one-sided switching fabrics notation (I_i, I_j, s) will be used, where I_j is the first stage switch with terminal y).

Let us assume that in an input terminal (an output terminal) there are k connections which use $s_1, s_2, \ldots, s_k$ slots, respectively. Let as also assume that the capacity of the input terminal is f_1 and the capacity of the output terminal is f_2 (i.e., there are f_1 slots in each of input links and f_2 slots in each of output links). We have $\sum_{i=1}^{k} s_i \leqslant f_1$ ($\sum_{i=1}^{k} s_i \leqslant f_2$). Similarly, if l connections using $s_1, s_2, \ldots, s_l$ slots are set up through one interstage link of capacity f_0, following condition is true: $\sum_{i=1}^{l} s_i \leqslant f_0$. In an input terminal (an output terminal) carrying k connections, a new s-slot connection can be added if and only if $\sum_{i=1}^{k} s_i \leqslant f_1 - s$ ($\sum_{i=1}^{k} s_i \leqslant f_2 - s$). An interstage link carrying already l connections can be used by the s-slot connections if and only if $\sum_{i=1}^{l} s_i \leqslant f_0 - s$. A new s-slot connection will be called *compatible with the link* if there are s free slots in this link which fulfill the assignment algorithm used. For instance in the case of floating assignment a new s-slot connection is compatible with the link if there are s consecutive free slots in it. A new s-slot connection may be compatible with an input terminal (an output terminal) only if not more than $f_1 - s$ ($f_2 - s$) slots are occupied. Similarly, in an interstage link not more than $f_0 - s$ slots may be used. New connection $\langle x, y, s\rangle$ will be called *compatible with the state of a switching fabric*, if it is compatible with input terminal x and output terminal y. New connection (I_i, O_j, s) will be called compatible with the state of a switching fabric, if it is compatible with at least one input of switch I_i and with at least one output of switch O_j. If a new connection is not compatible with an interstage link, then this link will be called *inaccessible* by the connection. If a new connection is compatible with the state of a switching fabric, than switches in inner stages are to be found to route this connection. The inner stage switch will be called accessible by connection (I_i, O_j, s) if one of its inputs and one of its outputs which might be used by the connection are accessible by this

connection. When there is no such pair of input and output, the switch is *inaccessible* by the connection.

2.5.3 Multirate Switching Model

In multi-path multirate multistage switching fabrics cells may be routed from an input terminal to any output terminal through different paths. Depending on the routing decision time switching fabrics may be divided into to classes: connection based and [33]. In the cell based switching fabrics the routing decision is taken cell by cell. This means that different paths through the switching fabric can be taken by different cells of the same virtual connection, and cells can arrive out of sequence. In the connection based switching fabrics, the path of a virtual connection is determined once for the duration of the connection, and all cells of this connection will always follow the same path through the switching fabric. It means that full cell sequence integrity is guaranteed. In this connection oriented approach *a weight* is assigned to a connection, which represent the required bandwidth [116, 114]. This weight may correspond to maximum, minimum, average bandwidth used by the connection. The concept known as *equivalent bandwidth* or *effective bandwidth* my be also applied [10]. In this concept an appropriate effective bandwidth is assigned to each connection and each connection is treated as if it required this effective bandwidth throughout the active period of the connection. The effective bandwidth of the connection is usually some value between its average rate and its peak rate. A given set of connections can be admitted, provided that the sum of the effective bandwidths is less than or equal to the total available bandwidth of the connection path. Usually, the weight represents *normalized bandwidth* required by the connection. The total normalized bandwidth of the input (output) link is called the *link capacity*, and is equal to β ($\beta \leqslant 1$). Interstage links have capacity equal to 1. For instance, if the bandwidth of the inter-stage links is 620 Mb/s and the bit rate of the input (output) terminal is 155 Mb/s, then the normalized bandwidth is 1 and 0.25, respectively.

In the rest of this book following notation will be used in case of multirate connections. A new connection of weight ω between input terminal x and output terminal y will be denoted by $\langle x, y, \omega \rangle$. Notation (I_i, O_j, ω) will be used to denote a connection of weight ω between the first stage switch I_i and the last stage switch O_j if the numbers of input and output terminals are not important. In general, $0 \leqslant \omega \leqslant \min\{\beta_1; \beta_2\} \leqslant 1$, since it has to be accommodated in one of the input (output) terminals, β_1 and β_2 denote the normalized capacity an input terminal and an output terminal, respectively.

Let us assume that in an input terminal (an output terminal) there are k connections of weights $\omega_1, \omega_2, \ldots, \omega_k$. We have $\sum_{i=1}^{k} \omega_i \leqslant \beta_1$ ($\sum_{i=1}^{k} \omega_i \leqslant \beta_2$). If through an interstage link l connections of weights $\omega_1, \omega_2, \ldots, \omega_l$, are already set up, then we have $\sum_{i=1}^{l} \omega_i \leqslant 1$. In an input (output) terminal already carrying k connections, a new connection of weight ω can be set up, if and only if $\sum_{i=1}^{k} \omega_i \leqslant \beta_1 - \omega$ ($\sum_{i=1}^{k} \omega_i \leqslant \beta_2 - \omega$). If these conditions are not true, the input (output) terminal is *inaccessible* by a new connection of weight ω. Similarly, an interstage link already carrying l connections is *accessible* by a new connection of weight ω if and only if $\sum_{i=1}^{l} \omega_i \leqslant 1 - \omega$. Otherwise, this link is *inaccessible* by the new connection. A new connection is *compatible* with the state of a link, if this link is accessible for this connection. A new connection (I_i, O_j, ω) is *compatible* with the state of the switching fabric, if it is compatible with one of the inputs of switch I_i, and with one of the outputs of switch O_j. Let us assume that the new connection (I_i, O_j, ω) is to be set up, and this connection is compatible with the state of the switching fabric. To set up this connection, switches in inner stages accessible by this connection are to be found. An inner stage switch is *accessible* by connection (I_i, O_j, ω), if one of its inputs and one of its outputs which might be used by the connection are accessible by this connection. Otherwise this switch is *inaccessible* by this connection.

Usually, the weights of all connections belong to a closed interval $[b, B]$, where $0 \leqslant b \leqslant B \leqslant 1$. As already stated earlier, two cases are defined: discrete bandwidth and continuous bandwidth [22]. The formal definitions are as follows.

DEFINITION 2.1 *Connection $\langle x, y, \omega \rangle$ is the discrete bandwidth connection if ω belongs to a given finite set $\{b_1, b_2, \ldots, b_k\}$, where b_1 is a divisor of b_i, $i = 2, \ldots, k$. In this case $b = b_1$ and $B = \max\{b_i : i = 1, 2, \ldots, k\}$.*

DEFINITION 2.2 *Connection $\langle x, y, \omega \rangle$, is the continuous bandwidth connection if ω belongs to a closed interval $[b, B]$, where $0 \leqslant b \leqslant B \leqslant \beta \leqslant 1$.*

A multicast connection of weight ω from input terminal x to the set $\mathbb{Y}$ of output terminals will be denoted by $\langle x, \mathbb{Y}, \omega \rangle$.

2.5.4 Graph Representation

Graphs are generally used as models for describing issues deriving from different and independent fields of knowledge. When we are interested in finding the interdependence between a given number of elements in a set we can apply modeling by graphs. Switching fabrics and connections can also be modeled using graph representation. Such rep-

resentation enables, for instance, the use of graph algorithms for path searching or performance evaluation of a switching fabric. A graph consists of nodes (called also vertices) and lines joining pairs of nodes called edges. A set of nodes is usually denoted by $\mathbb{V}$, a set of edges by $\mathbb{E}$, and a graph is denoted by $\mathcal{G}(\mathbb{V}, \mathbb{E})$ or simply by $\mathcal{G}$.

The number of nodes of $\mathcal{G}$ is called *the order* of $\mathcal{G}$. Two nodes joined by an edge are said to be *adjacent*, and this edge is said to be *incident* to these nodes. Two edges of $\mathcal{G}$ incident to the same node are called *adjacent edges*. When two nodes are joined by two or more edges, then these edges are called *multiple edges*. A graph containing multiple edges is called *a multigraph*. The *degree of a node*, denoted by d_v, in a graph $\mathcal{G}$ is the number of edges of $\mathcal{G}$ incident with this node.

The maximum degree of all nodes in $\mathcal{G}$ is called the *degree of the graph* $\mathcal{G}$. When all nodes in $\mathcal{G}$ have degree n, a graph is called *n-regular*. A path in a graph $\mathcal{G}(\mathbb{V}, \mathbb{E})$ from node s to node t is a sequence of nodes $< v_0, v_1, v_2, \ldots, v_k >$ such as $s = v_0$, $t = v_k$, $(v_{l-1}, v_l) \in \mathbb{E}$. The number of edges k of the path is called *the length of the path*. A graph is *connected* if every node can be reached from any other node, i.e., if there is a path between any two nodes. A path containing at least two edges forms a cycle in a graph if $v_0 = v_k$. When a cycle traverses every edge of the graph exactly once, the cycle is called an Euler cycle. The problem of finding an Euler cycle in a graph is commonly seen in the form of puzzles where you are to draw a given figure without lifting your pencil from the paper, starting and ending at given points. A graph $\mathcal{G}$ has an Euler cycle if and only if it is connected and all its nodes are of even degree.

An *edge coloring* of $\mathcal{G}$ is an assignment of colors to the edges of $\mathcal{G}$ so that adjacent edges are colored with different colors. When all edges in $\mathcal{G}$ can be colored using n colors, the graph $\mathcal{G}$ is said to be *n-colorable*. A spanning subgraph of $\mathcal{G}$ is a graph $\mathcal{H}$, which contains all nodes and a set of edges of $\mathcal{G}$. Sometimes it is useful to assign a weight to edges, the weight is between 0 and 1. Such graph is called *a weighted graph*. A weighted graph may be also edge-colored. The requirement is now that, for all nodes, all edges incident to a given node and colored with the same color will have the total weight not greater than 1.

When nodes of graph $\mathcal{G}$ can be divided into two disjoint subset in such a way that each edge joins a node of the first set to a node of the second set, than such graph is called *a bipartite graph*. When a bipartite graph has multiple edges it is called *a bipartite multigraph*. *A perfect matching* in $\mathcal{G}$ is a set of edges of $\mathcal{G}$ no two of which are adjacent and which include every node in $\mathcal{G}$. An example of $\mathcal{G}$ and three different perfect matchings are shown in Fig. 2.18.

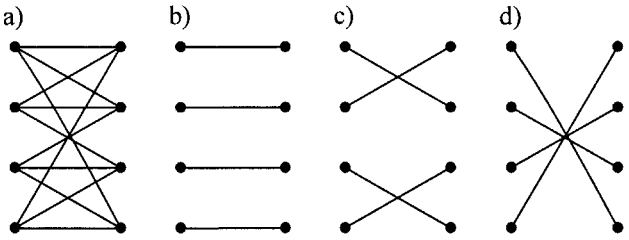


Figure 2.18. A bipartite graph (a) and perfect matchings (b), (c), and (d)

There are different approaches to represent a switching fabric as a graph. In one approach switches correspond to nodes and interstage links correspond to edges. An example of 6×6 switching fabric is shown in Fig. 2.19a and its graph representation is shown in Fig. 2.19b. Nodes may also represent inputs and outputs of a switch and edges may correspond to crosspoint in the switch. A switch is then represented by a bipartite graph as it is shown in Fig. 2.20. Switch fabric may be also represented in this way. Such representation of the switching fabric presented in Fig. 2.19a is shown in Fig. 2.21. In this approach input and output terminals correspond to the nodes of the first and last stage switches, respectively. In both approaches the graph representing a switching fabric is the bipartite graph since nodes of even numbered stages from the first set of nodes and nodes of odd numbered stages forms the second set of nodes. To determine some properties of switching fabrics (for instance blocking probability) it is only important to know possible paths between any pair of first and last stage switches. A graph representing such paths is called *a channel graph*. The channel graph for the switching fabric of Fig. 2.19a is shown in Fig. 2.22. When a switching fabric is v-dilated the respective graph is than a multigraph.

Graphs can be also used to represent connections in switching fabrics. A maximal assignment can be represented by a bipartite multigraph, in which the first set of nodes corresponds to the first stage switches, while the second set of nodes represents the last stage switches. Each connection is represented by an edge joining respective nodes. Such graph representing a maximal assignment has r_1 nodes in the first set and r_2 nodes in the second set, where r_1 and r_2 are the numbers of switches in the first and last stages, respectively. The degree of this graph is $\max\{n_1; n_2\}$, where n_1 is the number of inputs of the first stage switches and n_2 is the number of outputs of the last stage switches. Let the following maximal assignment be realized in the switching fabric of

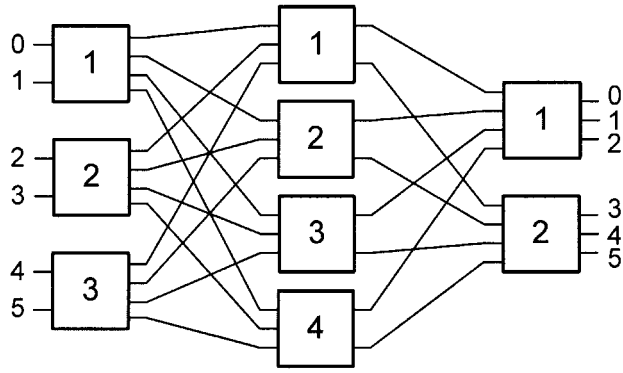


Figure 2.19a. An example of 6×6 switching fabric

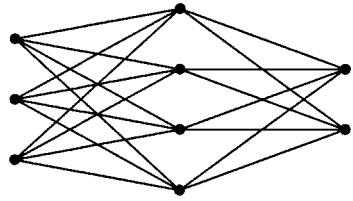


Figure 2.19b. The graph representation – nodes correspond to switches

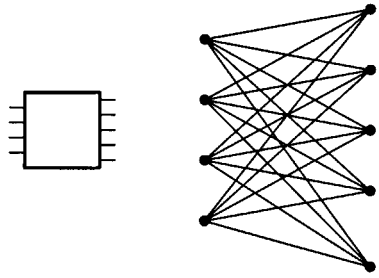


Figure 2.20. The 4×5 switch and its graph representation

Fig. 2.19a:

$$\Pi = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 0 & 2 & 1 & 4 \end{pmatrix}. \tag{2.2}$$

The bipartite graph representation of this assignment is shown in Fig. 2.23. The same approach may be used in case of multirate connections, however, this time respective weights are assigned to edges.

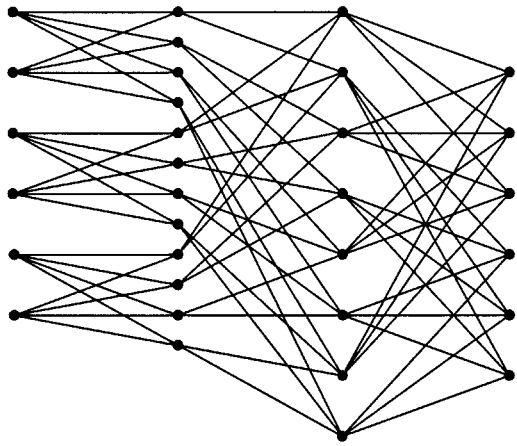


Figure 2.21. The graph representation – nodes correspond to links

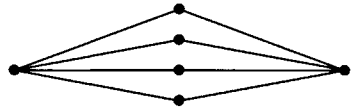


Figure 2.22. The channel graph of the switching fabric of Fig. 2.19b

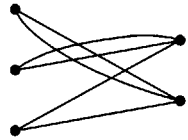


Figure 2.23. The graph representation of permutation (2.2)

Chapter 3

SINGLE-PATH AND STANDARD PATH SWITCHING FABRICS

3.1 Introduction

Structures and control algorithms of some single-path and standard path switching fabrics will be considered in this chapter. In a single-path switching fabric there is only one connecting path between any input terminal-output terminals pair. When a switching fabric is also strict-sense or wide-sense nonblocking, a control algorithm has only to change the states of appropriate switching elements in order to set up a connecting path. For switching fabrics of low capacity a control algorithm may use a state table with all possible states. When a new connection arrives, a control algorithm reads out how to set up switching elements directly from this table. The state table may contain information on how to set up each switching element separately, or how a group of these elements may be controlled together by one signal, depending on the switch fabric architecture. In the latter case, the number of control signals is less than the number of switching elements. The state of switching elements and control signals may be also deduced directly from input and output addresses.

3.2 Crossbar Switches

The crossbar architecture is often used to construct electronic and photonic switches. It may be composed either of crosspoints or 2×2 BSEs. In general, in this topology we have a matrix with rows and columns, and crosspoints (or BSEs) placed at intersections of rows and columns, as was shown in Fig. 2.3. Therefore, the crossbar switch is also called the *switching matrix* or the *matrix switch*. Usually, crossbar switches are considered as strictly nonblocking switches. However, this

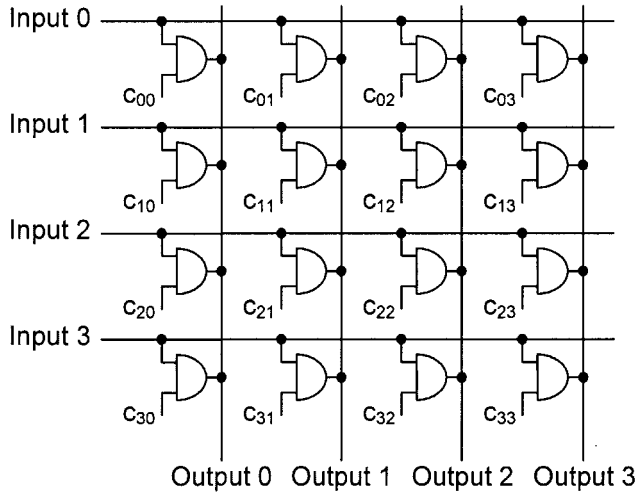


Figure 3.1a. The 4 × 4 crossbar switch with logical gates as crosspoints

	0	1	2	3
0	1	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	0	0	1

Figure 3.1b. The state, matrix for 4 × 4 crossbar switch with unicast connections

	0	1	2	3
0	1	1	0	0
1	0	0	0	0
2	0	0	1	1
3	0	0	0	0

Figure 3.1c. The state, matrix for 4 × 4 crossbar switch with multicast connections

property depends on the technology used. When, for instance, logical gates are used as crosspoints, signals are transferred from inputs to outputs depending on the control signals at the other inputs of these gates. There is only one possibility to transfer signal from any input to any output and only one gate can be used to switch a signal. So the switch is strictly nonblocking. In other realizations more than one path is possible, which will be described later on. The example of a 4 × 4 switch composed of logical gates is shown in Fig. 3.1a. Each gate is controlled by a control signal $c_{x,y}$. When input x is to be connected with output y control signal $c_{x,y}$ should be set to logical 1. Each gate has to be controlled separately, i.e., none of the control signals can control more than one gate. The state of control signals, which also determines the

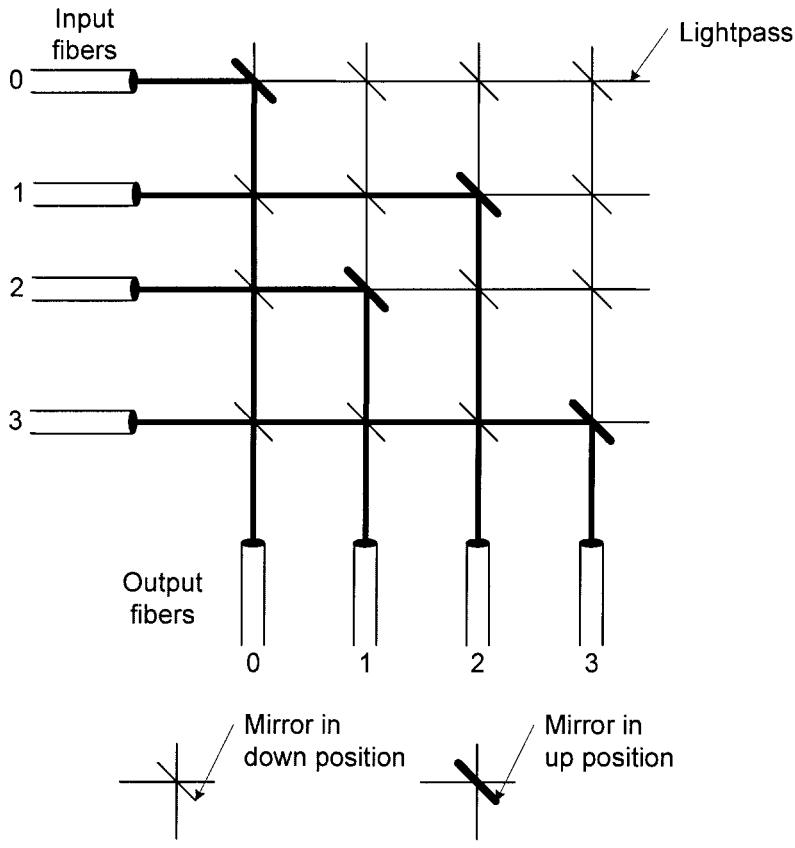


Figure 3.2. The 4 × 4 crossbar switch with micromirrors

state of the switch, may be arranged in state matrix **C**, in which each row corresponds to one input, each column corresponds to one output, and an entry $c_{x,y}$ corresponds to the state of respective control signal. In case of unicast connections, each column and each row has to contain only one logical 1, i.e., only one gate can be opened in each row and each column. An example of the state matrix **C** for connections $\langle 0,0\rangle$, $\langle 1,2\rangle$, $\langle 2,1\rangle$, and $\langle 3,3\rangle$ is shown in Fig. 3.1b. Such implementation enables also to realize multicast connections. The example of the state matrix for connections $\langle 0,\{0,1\}\rangle$ and $\langle 2,\{2,3\}\rangle$ is shown in Fig. 3.1c.

In photonic switches micromirrors are often used as crosspoints [110, 177]. When the mirror is in the down position, the light beam is passing over and does not change its direction. When the mirror is in up position, the light beam is reflected and changes its direction to the output.

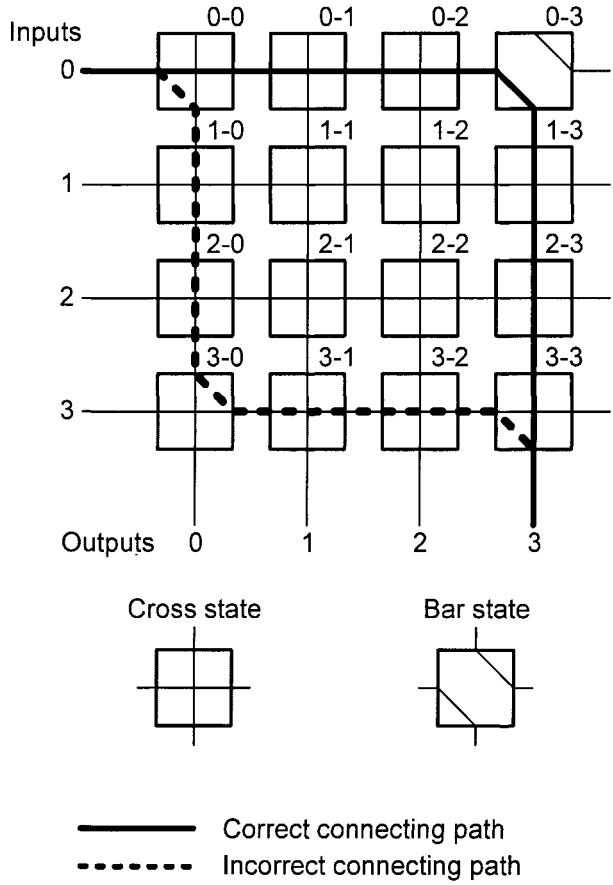


Figure 3.3. The 4 × 4 crossbar switch with BSEs

The example is shown in Fig. 3.2. The state of mirrors can be again represented in the state matrix, in which logical 1 denotes, for instance, the up position and logical 0 denotes the down position. In Fig. 3.2 state of mirrors correspond to the state matrix presented in Fig. 3.1b. This switch does not enable the realization of multicast connections.

Other implementations of crossbar switches use 2 × 2 BSEs (see Fig. 2.3b). They are used in both electronic and photonic switching. In this implementation in fact there are more connecting paths between any input-output pair. For instance, when connection $\langle 0, 3\rangle$ is set up as it is shown in Fig. 3.3 by dashed line, it is not possible to set up connection $\langle 3, 1\rangle$ or $\langle 3, 2\rangle$. The nonblocking operation is ensured when all BSEs are initially in cross state and for connecting input x with

output y the state of switch x-y is changed to the bar state. Therefore, crossbar switches with such implementation are considered as wide-sense nonblocking. This switch may be also treated as the standard path switch, since there are several connecting path between any arbitrary pair of input and output, in general; however, only one path (called the standard path) is always used to connect this input-output pair in order to preserve some switch fabric characteristics like nonblockingness. The state of the switch can be again stored in the state matrix $\mathbf{C}$, in which logical 1 corresponds to the bar state of respective switching element, and 0 to the cross state.

In the crossbar switch composed of 2×2 BSEs three BSEs can be omitted, and the whole architecture will be still nonblocking in the wide sense. Let us consider a 3×3 switch in which three switches above diagonal are omitted. All possible permutations which can be realized in this switch are shown in Fig. 3.4a. It can be seen that each connection use always the same route in the switch. For instance to set up connection $\langle 0, 2\rangle$ the state of the upper BSE in the middle column has to be changed from cross to bar, regardless of the other connections. The way BSEs should be controlled are given in Fig. 3.4b, where x-y near the BSE denotes that the state of this BSE is to changed when connection $\langle x, y\rangle$ is to be set up. All BSEs are controlled by one connection (one signal) except the upper BSE in the middle column, which is controlled by two connections, $\langle 0, 2\rangle$ and $\langle 1, 1\rangle$, but they are not in conflict between themselves and with other connections. In this case either of signals (or both) changes the state of this BSE.

In general, two top BSEs of the last column and one top BSE of the one before the last column are not necessary in the crossbar switch. This architecture will be called the reduced crossbar switch. The difference from the crossbar switch is that at the initial state, when all switches are in cross state, we have connections $\langle 1, n-1\rangle$ and $\langle 0, n-2\rangle$ set up in this state. The 4×4 reduced crossbar switch is shown in Fig. 3.5.

3.3 Triangular Switches

Triangular switches were proposed by Clos [23] as one-sided switches. The structure of such switch is presented in Fig. 3.6. It has n terminals and is composed of $(n^2 - n)/2$ crosspoints. One crosspoint is always used to connect two terminals. If we want to connect terminals x and y, $x < y$, we should set crosspoint at row x and column y, as shown in Fig 3.6 by the filled circle. If $x > y$ the connection is $\langle y, x\rangle$ since in such switch when there is connection $\langle y, x\rangle$ then there is also connection $\langle x, y\rangle$. The state of crosspoints may by controlled similarly as in the

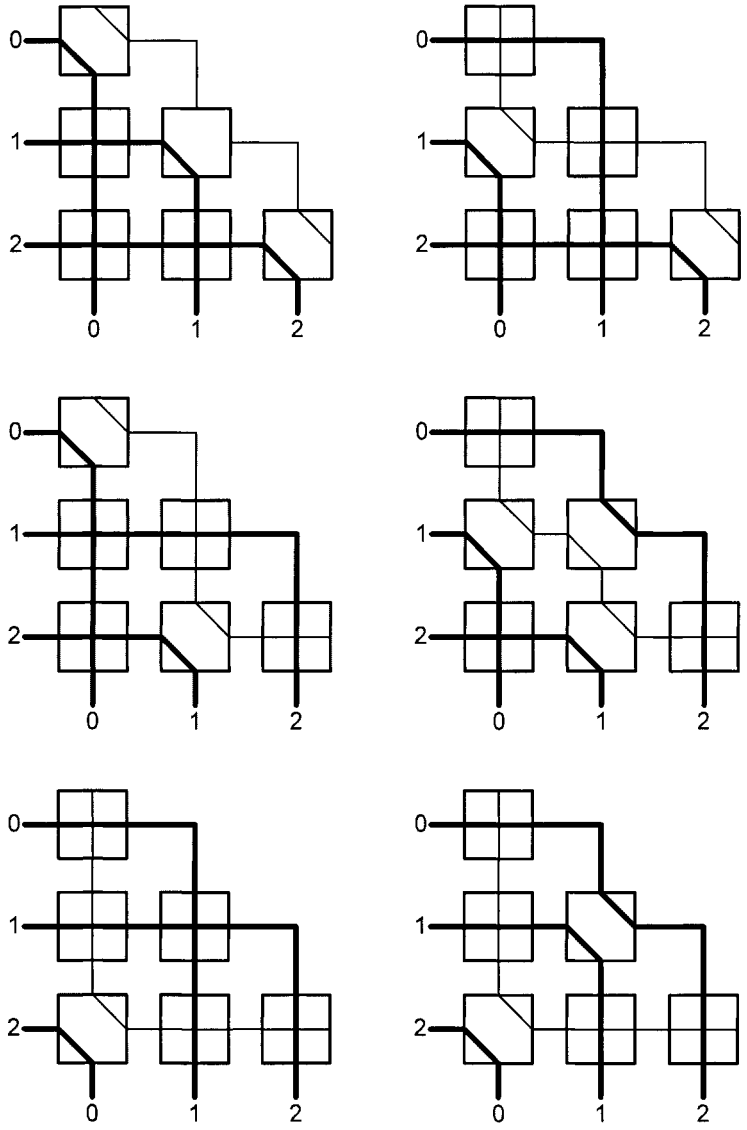


Figure 3.4a. Realization of all permutations in the 3 × 3 reduced crossbar switch with BSEs

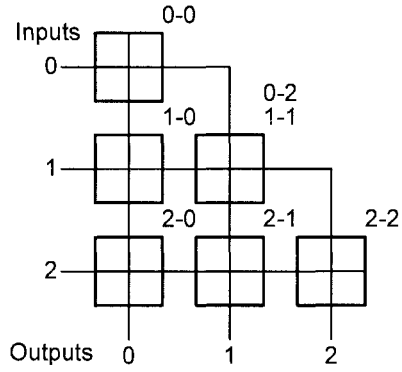


Figure 3.4b. Switch control in 3 × 3 triangular switch

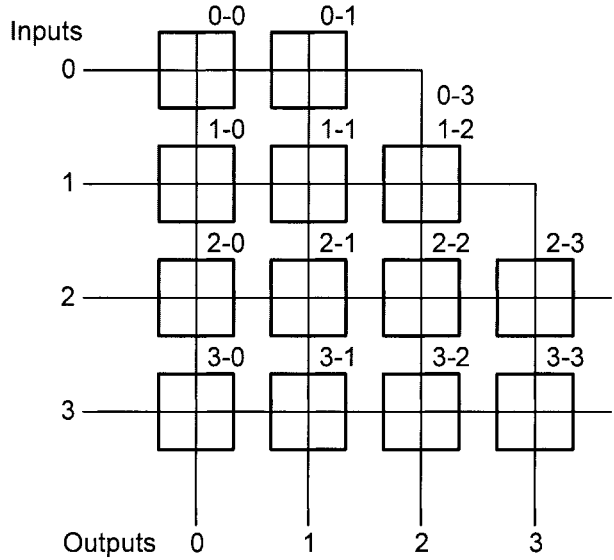


Figure 3.5. The 4 × 4 reduced crossbar switch with BSEs

crossbar switch by state matrix **C** but only cells above the diagonal are used.

Crossbar switch needs n^2 crossponts or BSEs. The triangular topology may be also used in case of two-sided switches composed of 2 × 2 elements, to reduce the required number of these elements [85, 134]. The general architecture of such switch is shown in Fig. 3.7 (for simpler further description the input and output numbering is changed). It contains

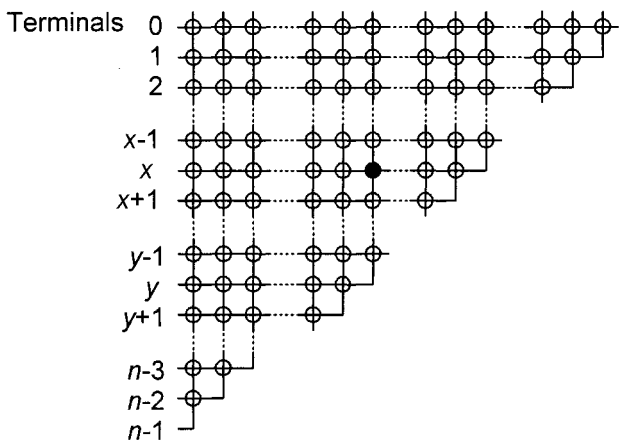


Figure 3.6. Triangular one-sided switch with n terminals

$(n^2-n)/2$ BSEs. This switch is neither strictly nor wide-sense nonblocking. However, all possible assignments between inputs and outputs can be realized in this switch [85].

THEOREM 3.1 *Triangular switch presented in Fig. 3.7 is rearrangeable nonblocking.*

Proof. We will use induction to prove this theorem. It is clear, that this switch realize all possible assignments for $n = 2$, since it is a 2×2 switching element. Let assume $(n-1) \times (n-1)$ switch is rearrangeable nonblocking. The switch of capacity $n \times n$ can be obtained by adding one column of $n-1$ switches as it is shown in Fig. 3.8. Output $n-1$ can be connected to any input using respective switch in this column. Let assume that permutation Π is to be realized in which $\pi(x) = n-1$:

$$\Pi = \begin{pmatrix} 0 & 1 & \cdots & x & \cdots & n-1 \\ \pi(0) & \pi(1) & \cdots & \pi(x) & \cdots & \pi(n-1) \end{pmatrix}. \tag{3.1}$$

To connect input x with output $n-1$ BSE x-$(n-1)$ is to be set to the bar state (normally all BSEs are in the cross state). We have to leave all BSEs k-$(n-1)$, $0 \leqslant k \leqslant x-1$ unchange. Through these BSEs inputs 0 to $x-1$ are connected to the same numbered inputs of $(n-1) \times (n-1)$ switch. By changing BSEs $(x+1)$-$(n-1)$ to $(n-2)$-$(n-1)$ also to the bar state, inputs $x+1$ to $n-1$ will be connected to inputs from x to $n-2$ of $(n-1) \times (n-1)$ switch, respectively (see Fig. 3.8). Now in $(n-1) \times (n-1)$ switch the permutation of $n-1$ elements is to be realized and since by induction assumption this switch is rearrangeable

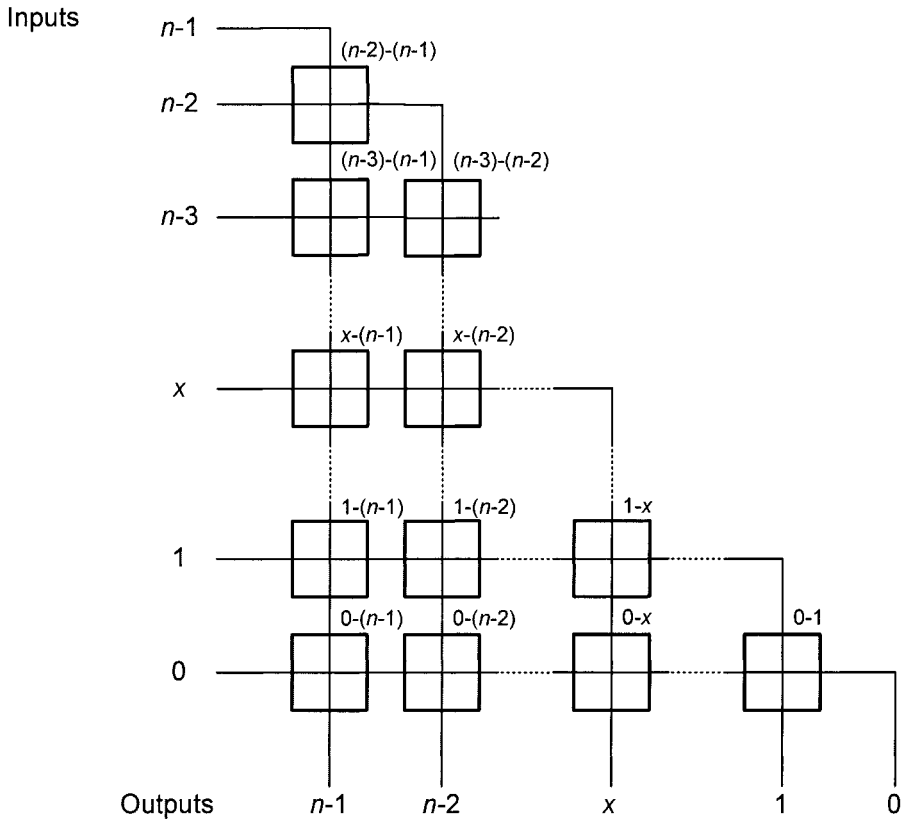


Figure 3.7. The general architecture of $n \times n$ triangular switch

nonblocking then any permutation can be realized in this switch. This means that also switch $n \times n$ is rearrangeable. □

The proof of the above theorem gives also the algorithm which can be used for simultaneous setting of any assignment.

ALGORITHM 3.1 *Permutation triangular*

Step 1 Connect not connected output y with the highest number to respective input x by changing the state of BSE x-y to bar.

Step 2 Change the state of BSEs k-y, $x < k < n$.

Step 3 Reduce by one input numbers greater than x.

Step 4 Repeat steps 1 to 3 until all outputs are connected.

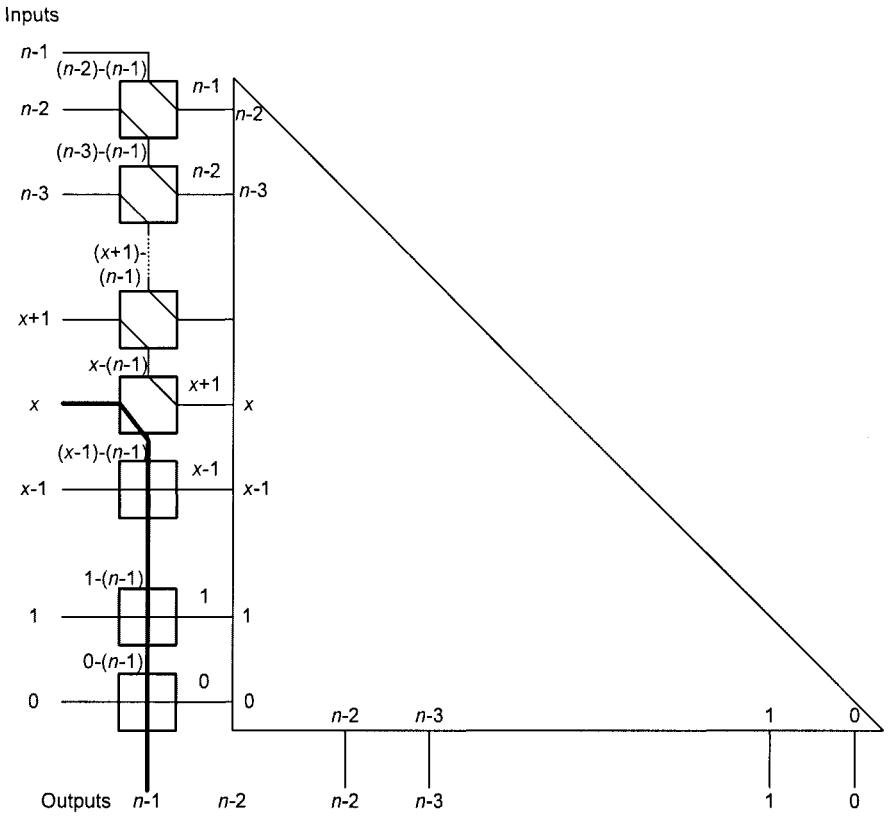


Figure 3.8. Recursive construction of $n \times n$ triangular switch and path set up for one output

Let the following permutation is to be set up in 8×8 triangular switch:

$$\Pi = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 4 & 7 & 5 & 2 & 6 & 3 & 0 \end{pmatrix}. \tag{3.2}$$

This switch with connections is shown in Fig. 3.9. First output 7 will be connected to input 2. Therefore BSE 2-7 is changed to the bar state, and also BSEs from 3-7 to 6-7 are changed to this state. Since input 2 is connected we remove connection $\langle 2, 7 \rangle$ from permutation Π and reduce by one input numbers from 3 to 7. The new permutation is

$$\Pi' = \begin{pmatrix} 0 & 1 & 2' & 3' & 4' & 5' & 6' \\ 1 & 4 & 5 & 2 & 6 & 3 & 0 \end{pmatrix}, \tag{3.3}$$

where mark $'$ denotes inputs whose numbering is reduced by one. Now the next not connected output with the biggest number is output 6 and it

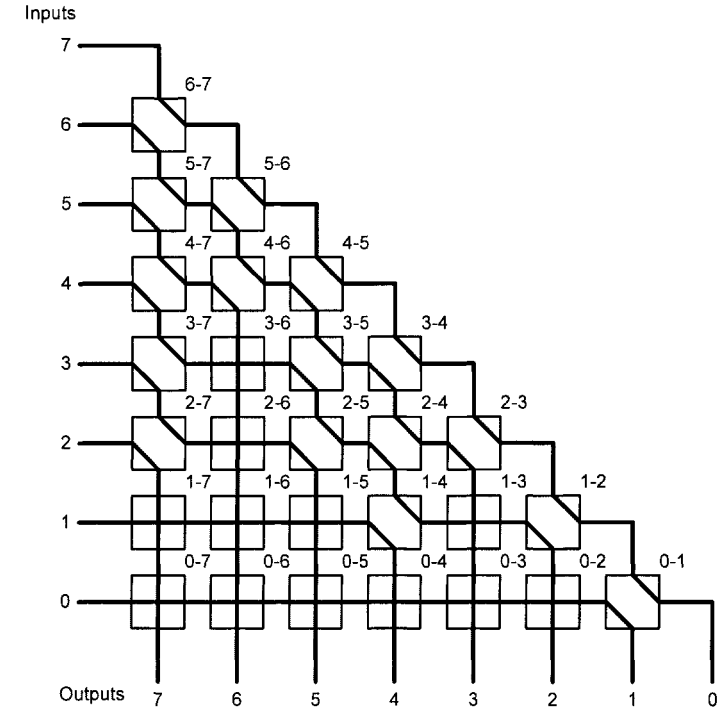


Figure 3.9. The example of connections in 8×8 triangular switch

is to be connected with input $4'$ (originally 5). Thus BSE 4-6 is changed to bar state and the state of BSE 5-6 is also changed. Now numbers of inputs greater than $4'$ are reduced by one so the remaining connections are represented by permutation

$$\Pi'' = \begin{pmatrix} 0 & 1 & 2' & 3' & 4'' & 5'' \\ 1 & 4 & 5 & 2 & 3 & 0 \end{pmatrix}, \tag{3.4}$$

where marks $'$ and $''$ denote input numbers which were changed after connecting input 7 and 6, respectively. The final set up of BSEs is shown in Fig. 3.9.

3.4 Tree-type Switching Fabrics

The general structure of $n \times n$ switch of tree architecture is shown in Fig. 3.10. It consists of n $1 \times n$ splitters and n $n \times 1$ combiners. Output y of splitter x, $0 \leqslant x, y \leqslant n-1$, is connected to input x of combiner y. Any $1 \times n$ ($n \times 1$) switch can be used as the splitter (combiner). Splitters and combiners can be also fabricated from 1×2 (2×1) or 2×2 BSEs.

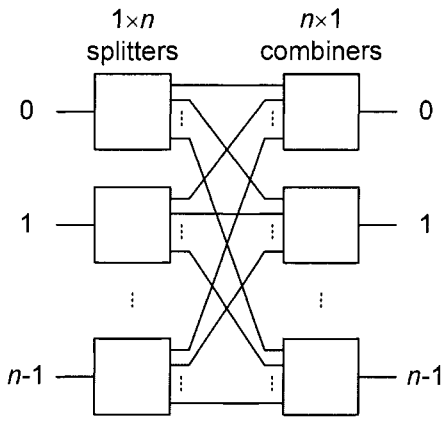


Figure 3.10. The $n \times n$ tree-structured switch

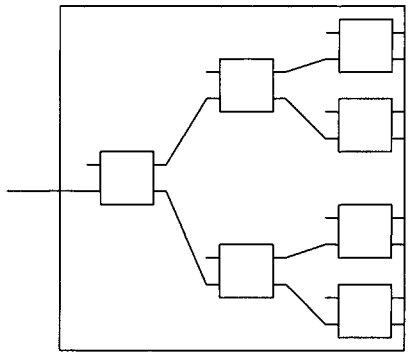


Figure 3.11. The 1×8 splitter composed of 2×2 BSEs

The 1×8 splitter composed of 2×2 BSEs is shown in Fig. 3.11. In each 2×2 BSE only one input is used, the other one is left unused. BSEs are arrange in $\log_2 n$ stages. The combiner is the mirror image of the splitter.

In the tree architecture there is only one path between any input-output pair. Connection is set up by changing states of appropriate BSEs in respective splitter and combiner. Since splitters and combiners are controlled, they are called active splitters and active combiners, and the whole tree architecture is referred to as active splitters/active combiners (AS/AC). Two other approaches can be used in tree type structure. In the first one signals from each input are broadcasted to all outputs of splitters, so each combiner receives data from all inputs and it selects one of them. In this approach only combiners are controlled. Such structure is called passive splitters/active combiners (PS/AC). The next approach uses active splitters and passive combiners (AS/PC).

In the tree-structured switches, switching elements placed in one stage of active splitters or active combiners can be controlled by one control signal. Let cross and bar states of 2×2 switches be controlled by signals 0 and 1, respectively. Let a new connection $\langle x, y \rangle$ is to be set up, and let $x_{n-1}, \ldots, x_0$ and $y_{n-1}, \ldots, y_0$ be binary representations of x and y, respectively. The stages of switching elements in active splitter x are controlled by signals $y_{n-1}, \ldots, y_0$, while the stages in active combiner y – by $x_0, \ldots x_{n-1}$. The example of 4×4 switch with connection $\langle 1, 2 \rangle$ is shown in Fig. 3.12. Control signals of splitter 1 and combiner 2 are shown as arrows.

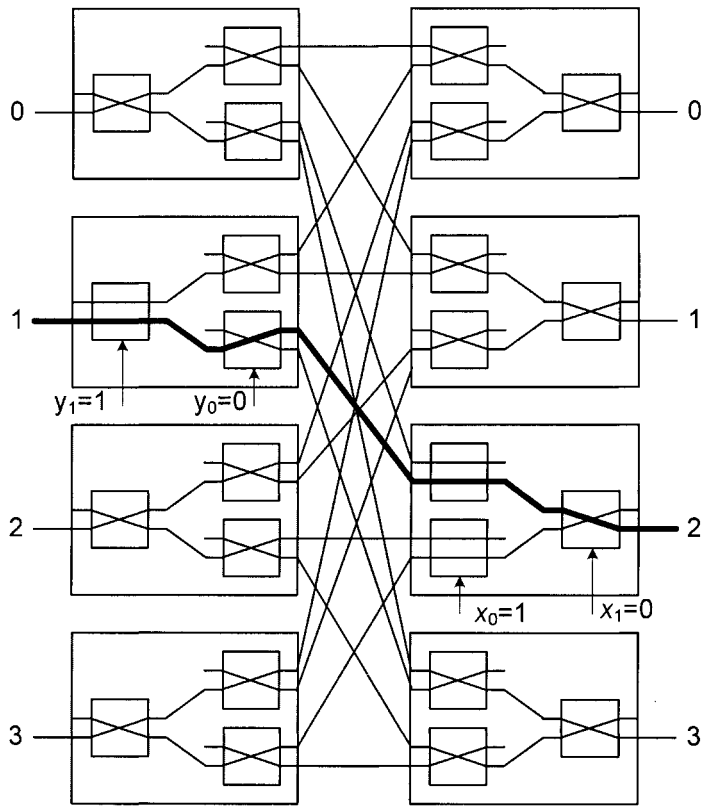


Figure 3.12. Path setup in a tree architecture

3.5 One-stage Switching Fabrics

The $2n \times 2n$ one-stage switching fabric composed of four $n \times n$ switches is shown in Fig. 2.6. In general, a switching fabric of capacity $kn \times kn$ may be constructed. The one-stage switching fabric of such capacity is shown in Fig. 3.13. Switches are grouped into k groups, each group contains k switches. Inputs of switches in each group are connected in parallel. Outputs of switches with the same numbers in each group are also connected in parallel. Such switching fabric contains k^2 switches.

The one-stage switching fabric has a single path between any input-output terminals pair (we assume that crossbar switches are used). If any input terminal in group i, $1 \leqslant i \leqslant k$, is to be connected with arbitrary output terminal in group j, $1 \leqslant j \leqslant k$, the connecting path is set up through switch j of group i.

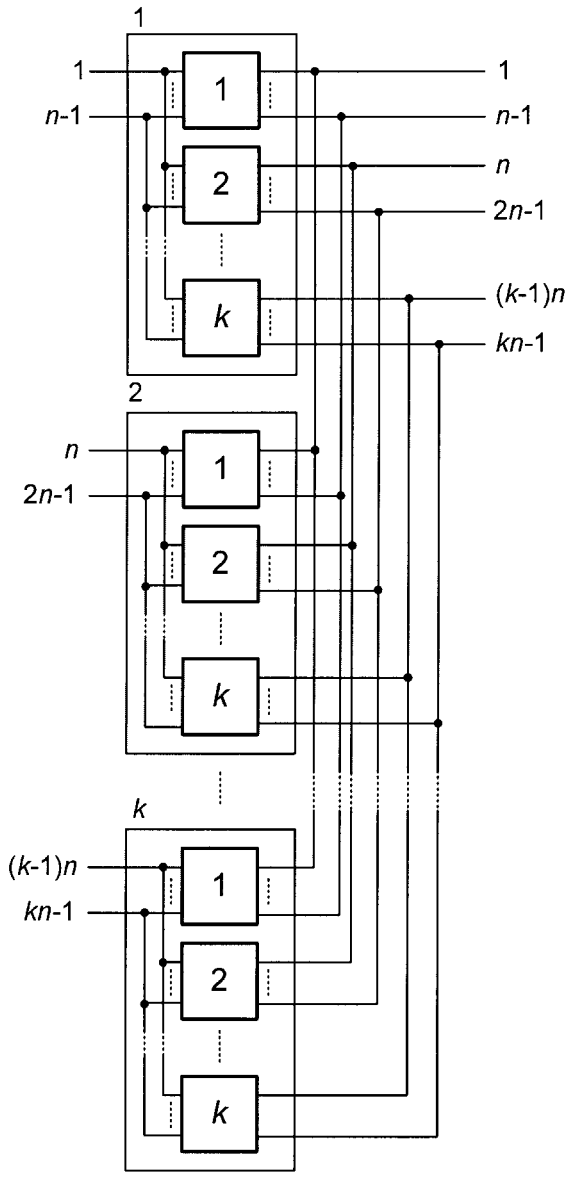


Figure 3.13. The one-stage $kn \times kn$ switching fabric composed of $n \times n$ switches

An alternate way of drawing the one-stage switching fabric is shown in Fig. 3.14. It has $kn \times kn$ capacity, and it may be seen as a multi-stage switching fabric. One may consider switches 1,1; 2,1; ... k,1 as switches of stage 1, but outputs of these switches are not connected to

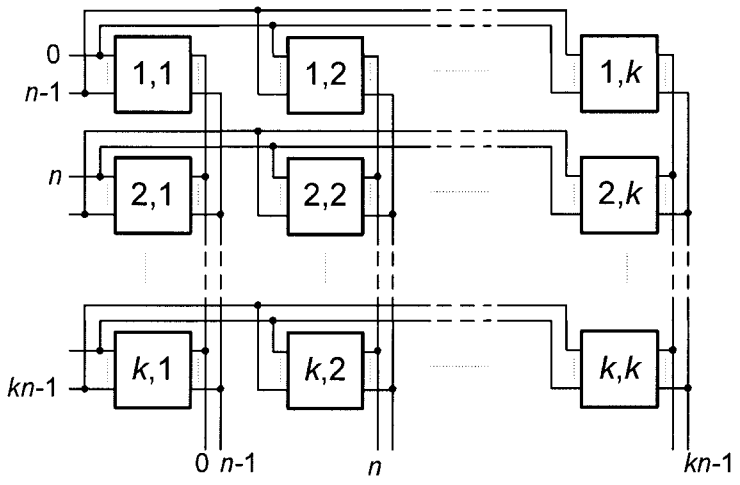


Figure 3.14. The alternate way of drawing of $kn \times kn$ one-stage switching fabric

inputs of switches 1,2; 2,2; ... k,2, but they constitute the switching fabric's output terminals. The connecting path considered earlier is set up through the switch denoted by i, j.

3.6 Banyan-type Switching Fabrics

3.6.1 Structures

Banyan-type switching fabrics were proposed for using in computer networks, but they now also attract attention in high-speed electronic and photonic switching fabrics of high-performance and high-capacity routers and optical cross-connect systems. In general the switching fabric of capacity $N \times N$ is constructed from $d \times d$ switches arranged in $n = \log_d N$ stages. They are also referred to as $\log_d N$ or d-nary switching fabrics. An example of 3-nary switching fabric is sown in Fig. 3.15. When $d = 2$ the switching fabric is called *binary*. There are several structures of binary networks, namely: banyan, baseline, omega, n-cube. They differ in the way of interconnecting switches in adjacent stages and input (or output) terminals to the switches of the first (last) stage [138]. Examples of 16×16 banyan, baseline, and omega switching fabrics are shown in Fig. 3.16a, 3.16b, and 3.16c, respectively. The reverse versions of these topologies, obtained as the mirror image of the network itself, are also known. All these topologies are equivalent, i.e., one can be obtained from another by reordering switches in stages without changing the way switches were interconnected between themselves. For instance, 8×8

banyan^{-1} (i.e., reverse banyan) switching fabric can be obtained from banyan one by exchanging switches 2 and 3 in all stages. When switches 2 and 3 are exchanged in only the first stage, the baseline switching fabric is obtained. These examples are shown in Fig. 3.17. The equivalence of banyan-type switching fabrics was considered in some papers [2, 137, 170].

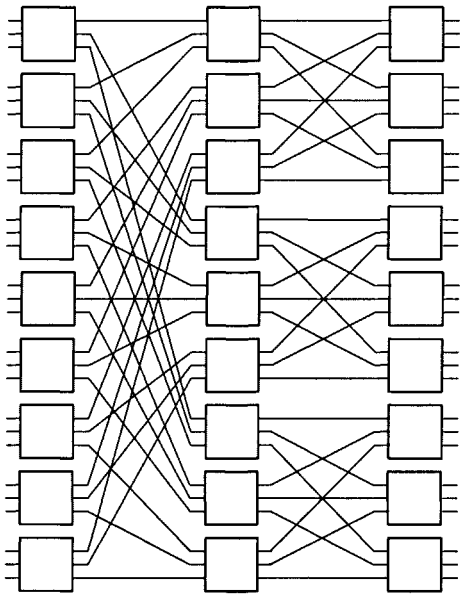


Figure 3.15. 27×27 3-nary switching fabric

In the rest of this book we will use terms banyan-type and $\log_d N$ switching fabrics interchangeably, and we will mainly refer to the baseline topology. This topology may by also constructed recursively. An $N \times N$ switching fabric is constructed by taking two copies of $N/2 \times N/2$ switching fabrics and adding one stage of $N/2$ 2×2 switching elements. Outputs of each 2×2 switching element are connected one-to-one to $N/2 \times N/2$ switching fabrics, as shown in Fig. 3.18.

In switching fabrics considered above, two switches of adjacent stages were interconnected by means of one interstage link. In general, there may be more than one such link. If there are v such links, we obtain the v-dilated switching fabric. Its general architecture is shown in Fig. 3.19. The $N \times N$ switching fabric contains d^{n-1} switches in each of n stages. Switches of the first stage have the capacity of $d \times dv$, switches of the nth stage are of capacity $dv \times d$, and switches in the remaining stages have dv inputs and dv outputs.

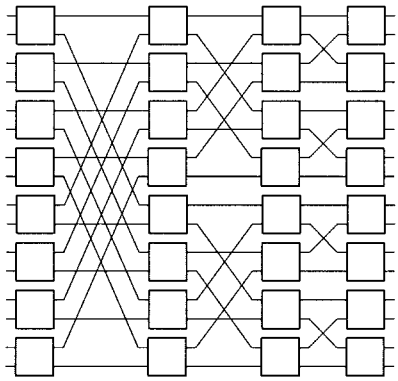


Figure 3.16a. The 16 × 16 banyan switching fabric

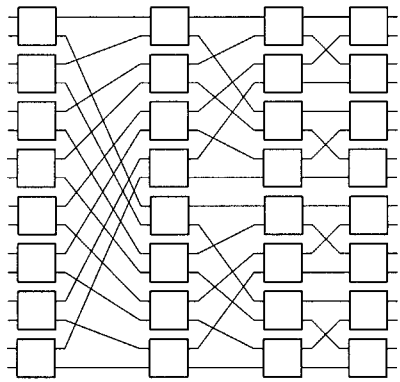


Figure 3.16b. The 16 × 16 baseline switching fabric

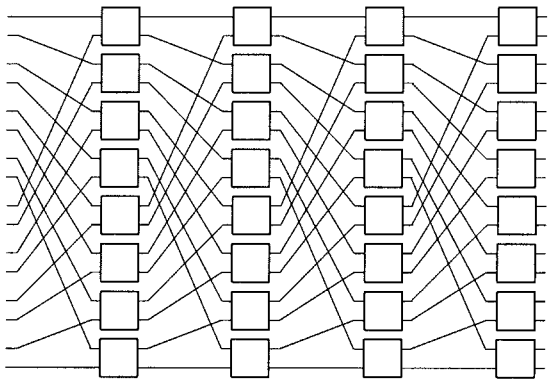


Figure 3.16c. The 16 × 16 omega switching fabric

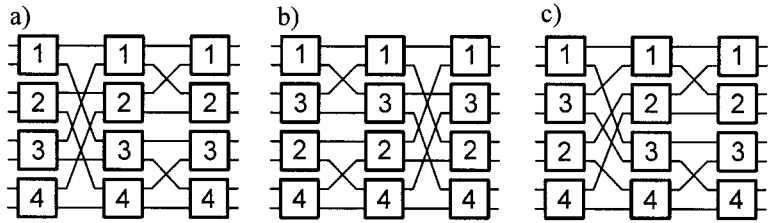


Figure 3.17. Equivalence of 8×8 banyan (a), banyan^{-1} (b) and baseline (c) switching fabrics

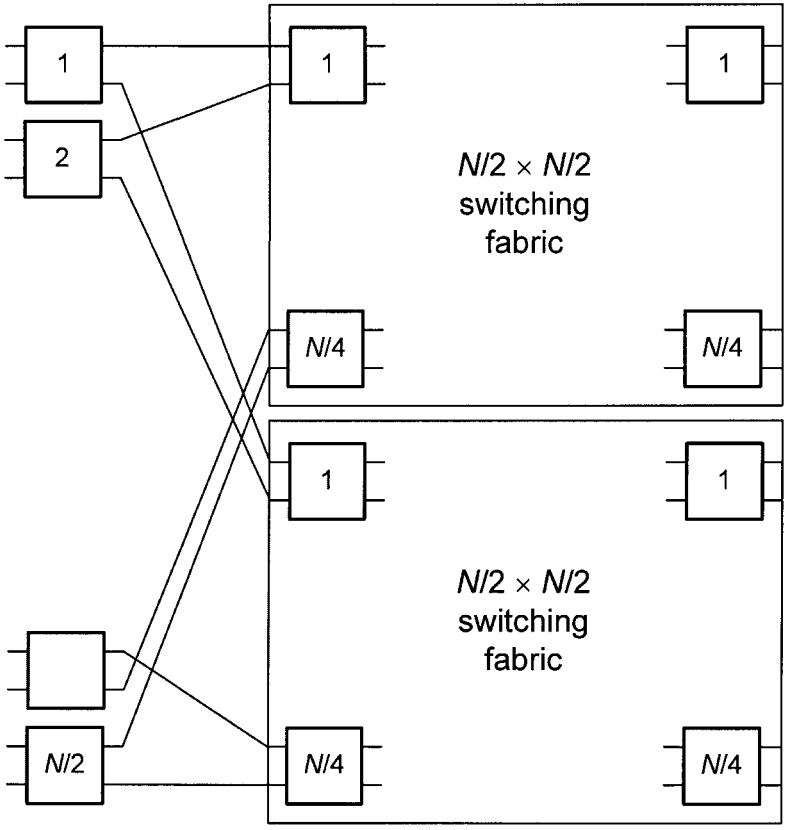


Figure 3.18. Recursive construction of $N \times N$ baseline switching fabric

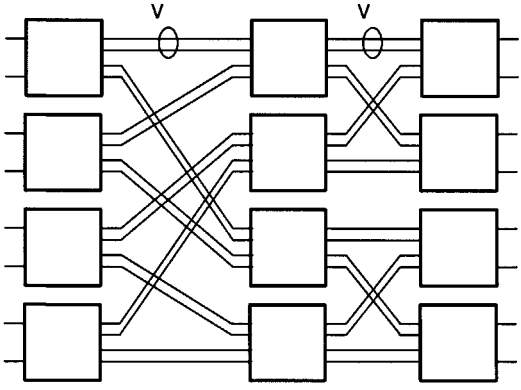


Figure 3.19. A v-dilated 8×8 baseline switching fabric

3.6.2 Properties and Control

Let input terminals in the $\log_2 N$ switching fabric be numbered from 0 to $N-1$, from top to bottom; let stages be numbered from 1 to n, $n = \log_2 N$ from left to right; and let switches in each stage be numbered from 1 to $N/2$, from top to bottom. Banyan type binary networks have following properties:

- From each input terminal there is exactly one connecting path to each output terminal.
- *Buddy property*: If switch I at stage i is connected to two different switches J and K at stage $i+1$, then these two switches are connected also to the switch L at stage i, $L \neq I$.
- *Constrained reachability property*: If a switch at stage i can reach 2^j switches at stage $i+j$, then these switches are also reachable by exactly $2^j - 1$ other switches at stage i.
- *Self-routing*: It is sufficient to know input and output terminals to be connected to determine the route in the switching fabric.

One interesting property of $\log_2 N$ switching fabrics is their self-routing capability. The self-routing of connections was proposed in [90] for the omega switching fabric. It means that the only knowledge needed to route a connection from an input terminal to any output terminal is to know the numbers of these terminals. This property is used, for instance, to route packets in the switching fabric. Each packet has a label added in front of it and one bit of this label is used to control one switching element. The way this 2×2 switching element is controlled is shown in Fig. 3.20. The upper on lower outputs are numbered by 0 and 1, respectively. If a packet from any input is to be directed to the upper output, the first bit is 0, and it is set to 1, when the packet should be sent to the lower output. When the switching fabric has more stages, one bit is used to control the switch in one stage. Let $y_{n-1}, y_{n-2}, \ldots, y_1, y_0$ be a binary representation of output terminal y. We have $\log_2 N$ stages in the switching fabric and the same number of bits in this binary representation. Each bit is used to control one switch. This binary representation is added at the front of the packet with the most significant bit first transmitted. The example of self-routing is shown in Fig. 3.21. In this figure the packet from input terminal 0 is to be sent to output terminal 8 (1000 in binary) and packet from input terminal 15 is directed to output terminal 5 (0101). Stages from 1 to 4 are controlled by bits y_3 to y_0, respectively. Labels 1000 and 0101 are added to respective packets with

Figure 3.20. Self-routing capability of the 2×2 switching element

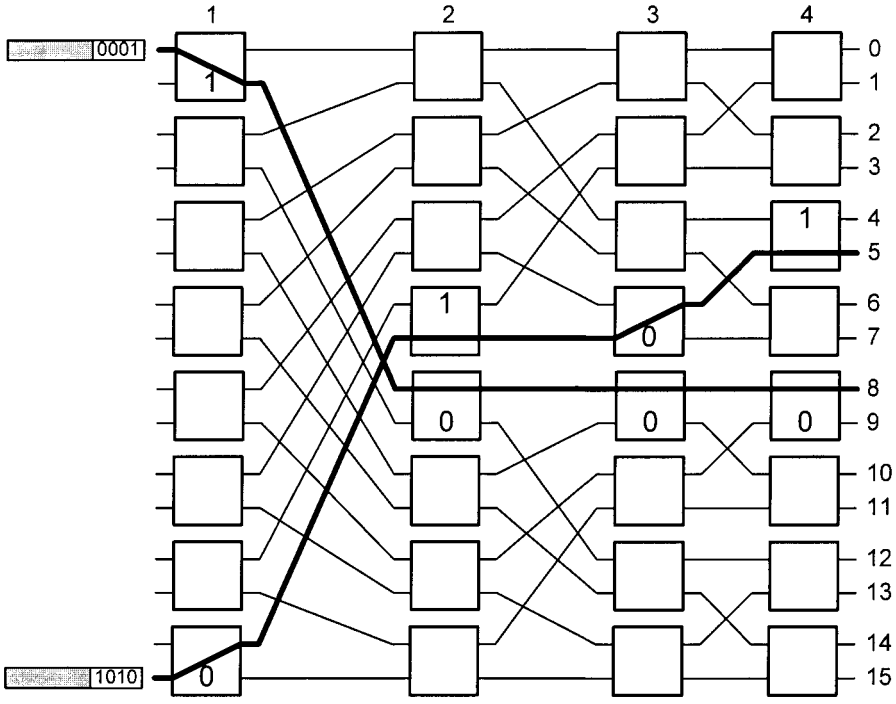


Figure 3.21. Self-routing in the 16×16 switching fabric

most significant bits at the first position. Routs are shown in bold lines and control bits are shown in appropriate switches.

Banyan-type switching fabrics have blocking states, i.e., not all possible assignments can be realized. The example of the conflict between two connecting paths is shown in Fig. 3.22. In this figure, connections $\langle 1, 1 \rangle$ and $\langle 3, 3 \rangle$ try to use the same interstage link between first switches of stages 2 and 3. Connections $\langle 12, 15 \rangle$ and $\langle 14, 14 \rangle$ are also in conflict since they have to use the same interstage links between last switches in stages 2, 3, and 4.

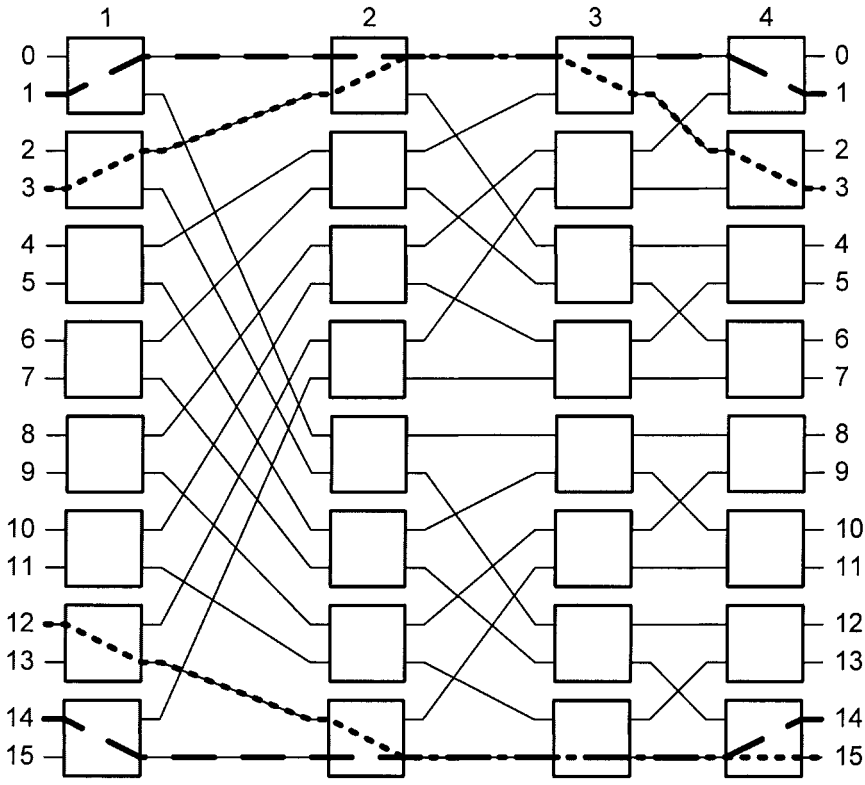


Figure 3.22. A conflict between two connections in 16 baseline switching fabric

3.6.3 Graph Representation

The bipartite graph representation of banyan-type switching fabrics was proposed in [92, 91, 95]. In this representation an edge (or link) corresponds to a switching element (crosspoint) and a node represents an input or an output of the 2×2 switching element. The bipartite graph representation of such switching element is shown in Fig. 3.23. The graph representation of $\log_2 16$ switching fabric contains $n+1$ stages of nodes, numbered $0, 1, 2, \ldots, n$ from left to right, respectively. The nodes in stage 0 correspond to input terminals and the nodes in stage n correspond to output terminals. These nodes are numbered $0, 1, \ldots, N-1$ from top to bottom, respectively. Such representation of the switching fabric of Fig. 3.22 is shown in Fig. 3.24.

Connection $\langle x, y \rangle$ is represented by the connecting path from node x in stage 0 to node y in stage n. Connecting paths representing connections $\langle 1, 1 \rangle$, $\langle 3, 3 \rangle$, $\langle 12, 15 \rangle$, and $\langle 14, 14 \rangle$ are shown in Fig. 3.24 in dashed lines.

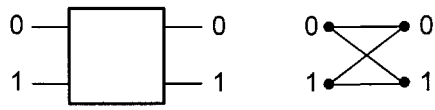


Figure 3.23. A graph representation of 2×2 switching element

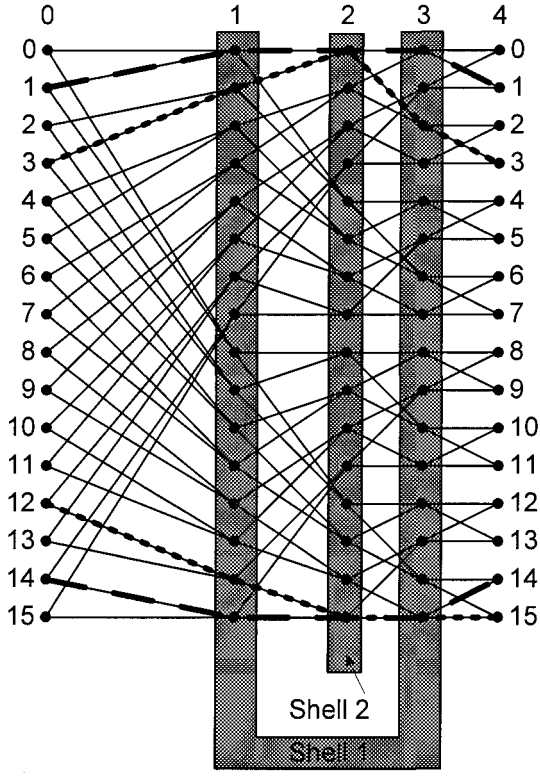


Figure 3.24. A graph representation of 16×16 baseline switching fabric

When two connecting paths representing two connections intersect in at least one node of the bipartite graph, then one of these connections is blocked. For instance connections $\langle 1, 1 \rangle$ and $\langle 3, 3 \rangle$ intersect in node 0 of stage 2, so only one of them can be set up in the switching fabric at the same time. Similarly, connections $\langle 12, 15 \rangle$ and $\langle 14, 14 \rangle$ intersect in nodes 16 of stages 2 and 3.

In the bipartite graph representing the $\log_2 N$ switching fabric, stage 1 and stage $\log_2 N - 1$ are called the first shell. Similarly stage 2 and stage $\log_2 N - 2$ are called the second shell, and so forth. Stage i will be

called the left part of shell i, while stage $\log_2 N - i$ will be called the right part of shell i. The total number of shells in the $\log_2 N$ switching fabric is equal to the center shell number, which is equal to $1/2(\log_2 N - 1)$ or $1/2 \log_2 N$, when $\log_2 N$ is odd or even, respectively. The center shell of the $\log_2 N$ fabric consists of stages $1/2(\log_2 N - 1)$ and $1/2(\log_2 N + 1)$ when $\log_2 N$ is odd, and it consists of stage $1/2 \log_2 N$ when $\log_2 N$ is even. The structure of the $\log_2 N$ switching fabric is symmetrical with respect to the center shell. It should be noted that a shell usually consists of two stages. The only exception is the center shell of a $\log_2 N$ switching fabric, when $\log_2 N$ is even; this center shell consists of only one stage. The shells in $\log_2 16$ switching fabric (n is even) are marked in Fig. 3.24. The bipartite graph representation with shells marked on it for $\log_2 32$ switching fabric (n is odd) is shown in Fig. 3.25.

Let us consider a connecting path representing connection $\langle x, y \rangle$. This path contains node x of stage 0, node y of stage n, and nodes in stages 1 trough $n - 1$. In nodes x and y path $\langle x, y \rangle$ cannot intersect with any other path. In stage 1 the considered path may intersects with one path from one node in stage 0. In stage 2 this path may intersects with the additional paths from two nodes in stage 0. In general, the considered path may intersects in a node of stage j, $1 \leqslant j \leqslant n - 1$, with 2^{j-1} additional paths from 2^{j-1} nodes in stage 0.

DEFINITION 3.2 *Let $\mathbb{SI}_j$ be the set of these input terminals (excluding input terminal x and all $\mathbb{SI}_{i's}$, where $i < j$) whose paths can intersect with path $\langle x, y \rangle$ in a node of stage j. Input terminals belonging to set $\mathbb{SI}_j$ will be called accessible from stage j.*

DEFINITION 3.3 *Let $\mathbb{SO}_j$ be the set of those output terminals (excluding output terminal y and all $\mathbb{SO}_{i's}$, where $i > j$) whose paths can intersect with path $\langle x, y \rangle$ in a node of stage j. Output terminals belonging to set $\mathbb{SO}_j$ will be called accessible from stage j.*

We have $|\mathbb{SI}_j| = 2^{j-1}$ and $|\mathbb{SO}_j| = 2^{n-j-1}$, where $|\mathbb{X}|$ denotes the cardinality of set $\mathbb{X}$. For example, let us consider path $\langle 0, 0 \rangle$ in the graph of Fig. 3.24, and $j = 3$. In this case $|\mathbb{SI}_3| = 2^2 = 4$, $\mathbb{SI}_3 = \{4, 5, 6, 7\}$ and $|\mathbb{SO}_3| = 2^{4-3-1} = 1$, $\mathbb{SO}_3 = \{1\}$.

For any connecting path we can construct the graph of intersecting paths. This graph has the same structure for any connecting path in the switching fabric. The topology of these graphs for switching fabrics with different number of stages differ, depending on n being odd or even. Graphs of intersecting paths in switching fabrics with n even and odd are shown in Fig. 3.26a and Fig. 3.26b for $n = 4$ and $n = 5$, respectively. The maximum number of paths meet in the node of stage $n/2$ for n even,

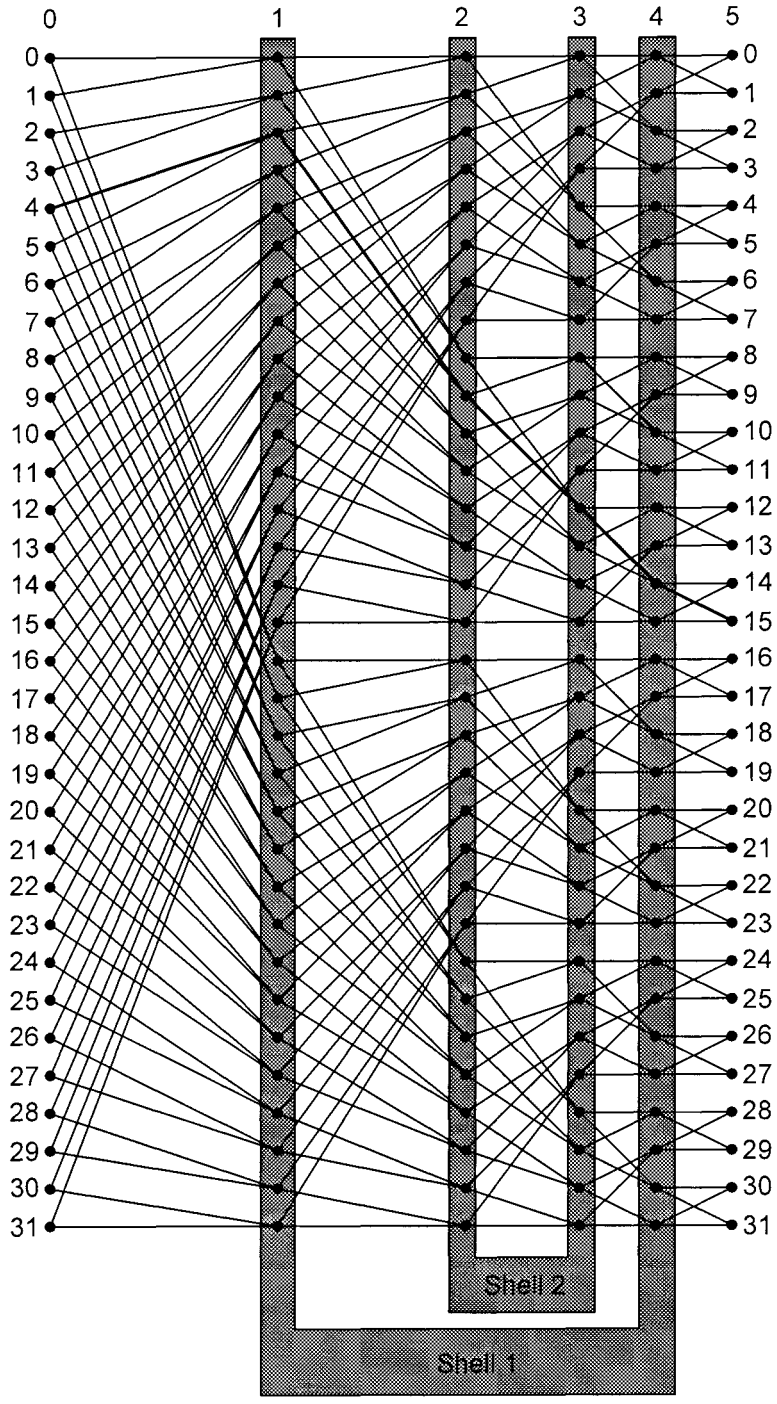


Figure 3.25. A graph representation of 32 × 32 baseline switching fabric

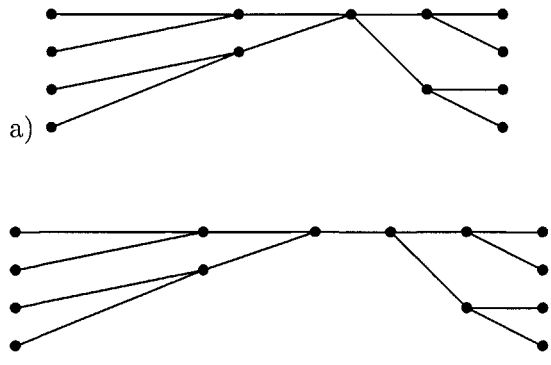


Figure 3.26. Graph of intersecting paths in $\log_2 16$ (a) and $\log_2 32$ (b) switching fabrics

and stages $(n-1)/2$ and $(n+1)/2$ for n odd [91, 92]. This number of paths is equal to $2^{\lfloor \frac{n}{2} \rfloor}$.

Let $x_{n-1}x_{n-2}\ldots x_1x_0$ and $y_{n-1}y_{n-2}\ldots y_1y_0$ be binary representations of input x and output y, respectively. Input and output terminals are divided into following sets:

DEFINITION 3.4 *The $\mathbb{I}_i$ denotes the set of input terminals x with the same values on bits $x_{n-1}x_{n-2}\ldots x_{\lfloor \frac{n}{2} \rfloor}$, where i is binary represented by $x_{n-1}x_{n-2}\ldots x_{\lfloor \frac{n}{2} \rfloor}$.*

DEFINITION 3.5 *The $\mathbb{O}_j$ denotes the set of output terminals y with the same values on bits $y_{n-1}y_{n-2}\ldots y_{\lfloor \frac{n}{2} \rfloor}$ where j is binary represented by $y_{n-1}y_{n-2}\ldots y_{\lfloor \frac{n}{2} \rfloor}$.*

When input terminals in $\mathbb{I}_i$ are to be connected with output terminals in $\mathbb{O}_j$, all paths for these connections are in conflict and their graph is a graph of intersecting paths. We will denote this graph by $\mathcal{G}(\mathbb{I}_i, \mathbb{O}_j)$. The connection between an arbitrary input in $\mathbb{I}_i$ and an arbitrary output in $\mathbb{O}_j$ will be denoted by $(\mathbb{I}_i, \mathbb{O}_j)$. When n is even, all paths in $\mathcal{G}(\mathbb{I}_i, \mathbb{O}_j)$ use node $j \cdot 2^{\frac{n}{2}} + i$ in stage $\frac{n}{2}$. When n is odd, all paths in $\mathcal{G}(\mathbb{I}_i, \mathbb{O}_j)$ use the same nodes in stages $\frac{n-1}{2}$ and $\frac{n+1}{2}$. However, in this case, these nodes may be also used by paths in other graphs. Let $\mathbb{I}_{\widehat{i}}$ denotes a set of inputs complementary to $\mathbb{I}_i$ in such a way, that i is represented by binary digits $x_{n-1}\ldots x_{\lfloor \frac{n}{2} \rfloor+1}0$, while $\widehat{i}$ is represented by $x_{n-1}\ldots x_{\lfloor \frac{n}{2} \rfloor+1}1$. The same is true for $\mathbb{O}_{\widehat{j}}$ and $\mathbb{O}_j$. For instance $\mathbb{I}_5$ is complementary to $\mathbb{I}_4$, $i = 4$ and $\widehat{i} = 5$. Graphs $\mathcal{G}(\mathbb{I}_i, \mathbb{O}_j)$ and $\mathcal{G}(\mathbb{I}_i, \mathbb{O}_{\widehat{j}})$ use the same node $j \cdot 2^{\frac{n-1}{2}} + i$ in stage $\frac{n-1}{2}$, while $\mathcal{G}(\mathbb{I}_{\widehat{i}}, \mathbb{O}_j)$ and $\mathcal{G}(\mathbb{I}_{\widehat{i}}, \mathbb{O}_{\widehat{j}})$ use the same node $j \cdot 2^{\frac{n-1}{2}} + \widehat{i}$

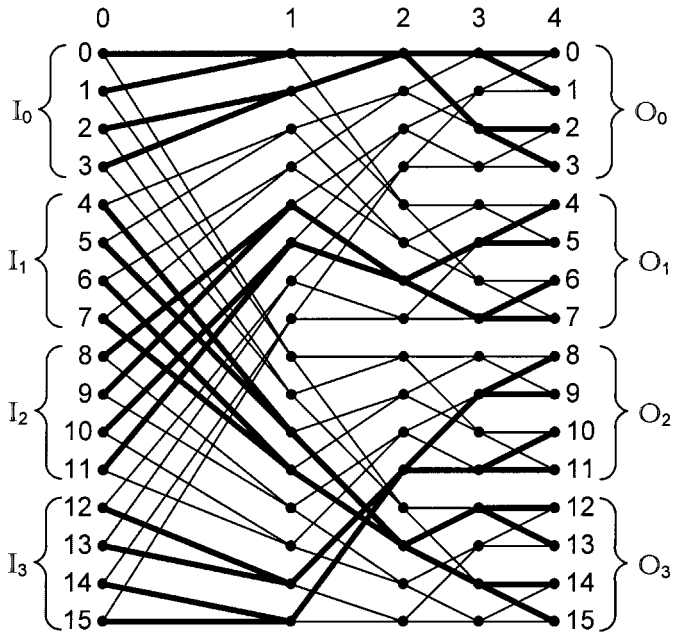


Figure 3.27. Graphs $\mathfrak{G}(\mathbb{I}_0, \mathbb{O}_0)$, $\mathfrak{G}(\mathbb{I}_1, \mathbb{O}_3)$, $\mathfrak{G}(\mathbb{I}_2, \mathbb{O}_1)$, and $\mathfrak{G}(\mathbb{I}_3, \mathbb{O}_2)$ in the 16×16 baseline switching fabric

(or $j \cdot 2^{\frac{n-1}{2}} + i + 1$). Similarly, connection paths in graphs $\mathcal{G}(\mathbb{I}_i, \mathbb{O}_j)$ and $\mathcal{G}(\mathbb{I}_{\widehat{i}}, \mathbb{O}_j)$ go through the same node $j \cdot 2^{\frac{n-1}{2}} + \frac{i}{2}$ in stage $\frac{n+1}{2}$ $(\widehat{j} \cdot 2^{\frac{n-1}{2}} + \frac{i}{2}$ for graphs $\mathcal{G}(\mathbb{I}_i, \mathbb{O}_{\widehat{j}})$ and $\mathcal{G}(\mathbb{I}_{\widehat{i}}, \mathbb{O}_{\widehat{j}}))$.

An example for 16×16 baseline switching fabric is shown in Fig. 3.27. We have $\mathbb{I}_0 = \mathbb{O}_0 = \{0, 1, 2, 3\}$, $\mathbb{I}_1 = \mathbb{O}_1 = \{4, 5, 6, 7\}$, $\mathbb{I}_2 = \mathbb{O}_2 = \{8, 9, 10, 11\}$, and $\mathbb{I}_3 = \mathbb{O}_3 = \{12, 13, 14, 15\}$. In this example $\mathcal{G}(\mathbb{I}_0, \mathbb{O}_0)$ use node 0 in stage 2, $\mathcal{G}(\mathbb{I}_1, \mathbb{O}_3)$ – node 13, $\mathcal{G}(\mathbb{I}_2, \mathbb{O}_1)$ – node 6, and $\mathcal{G}(\mathbb{I}_3, \mathbb{O}_2)$ – node 11, respectively. When inputs in $\mathbb{I}_i$ are to be connected with outputs in different sets $\mathbb{O}_j$ and $\mathbb{O}_k$, $k \neq j$, respective paths will go through different nodes in stage $n/2$.

The similar example in case of 32×32 baseline switching fabric is shown in Fig. 3.28. In this example, $\mathcal{G}(\mathbb{I}_0, \mathbb{O}_0)$ uses nodes 0 and 0 in stages 2 and 3, respectively, $\mathcal{G}(\mathbb{I}_1, \mathbb{O}_7)$ – nodes 25 and 28, $\mathcal{G}(\mathbb{I}_2, \mathbb{O}_3)$ – nodes 10 and 13, $\mathcal{G}(\mathbb{I}_3, \mathbb{O}_2)$ – nodes 11 and 9, $\mathcal{G}(\mathbb{I}_4, \mathbb{O}_1)$ – nodes 4 and 6, $\mathcal{G}(\mathbb{I}_5, \mathbb{O}_6)$ – nodes 29 and 26, $\mathcal{G}(\mathbb{I}_6, \mathbb{O}_5)$ – nodes 22 and 23, and $\mathcal{G}(\mathbb{I}_7, \mathbb{O}_4)$ – nodes 23 and 19. Node 0 in stage 2 is used by $\mathfrak{G}(\mathbb{I}_0, \mathbb{O}_0)$ and $\mathcal{G}(\mathbb{I}_0, \mathbb{O}_1)$. Similarly, node 1 of this stage is used by $\mathcal{G}(\mathbb{I}_1, \mathbb{O}_0)$ and $\mathcal{G}(\mathbb{I}_1, \mathbb{O}_1)$. In stage 3, for instance, node 4 is used by $\mathcal{G}(\mathbb{I}_0, \mathbb{O}_1)$ and $\mathcal{G}(\mathbb{I}_1, \mathbb{O}_1)$.

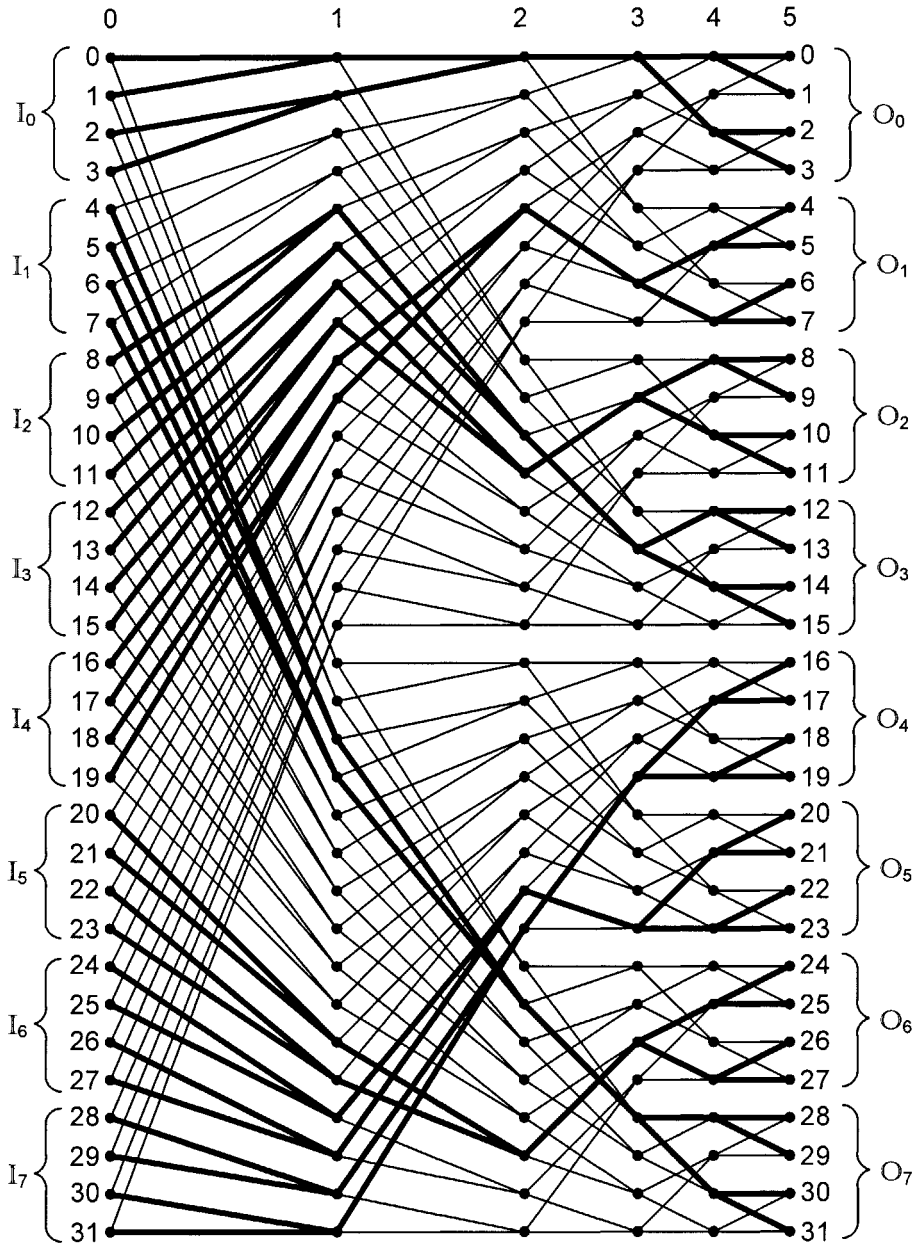


Figure 3.28. Graphs $\mathcal{G}(\mathbb{I}_0, \mathbb{O}_0)$, $\mathcal{G}(\mathbb{I}_1, \mathbb{O}_7)$, $\mathcal{G}(\mathbb{I}_2, \mathbb{O}_3)$, $\mathcal{G}(\mathbb{I}_3, \mathbb{O}_2)$, $\mathcal{G}(\mathbb{I}_4, \mathbb{O}_1)$, $\mathcal{G}(\mathbb{I}_5, \mathbb{O}_6)$, $\mathcal{G}(\mathbb{I}_6, \mathbb{O}_5)$, and $\mathcal{G}(\mathbb{I}_7, \mathbb{O}_4)$ in the 32×32 baseline switching fabirc

3.7 Implementations

Switching fabrics considered in this chapter have many practical implementations. Crossbar switches are implemented in deferent technologies and for different applications. This architecture is used to design integrated switches for packet switching (ATM, IP) and optical switching. In case of packet switches different solutions was proposed to overcome the head of line blocking (HOL) problem and improve the limited throughput of input buffered switches. Therefore, crossbar switches with greater internal speed of operation were proposed and developed [86, 123, 172]. Another approach uses multiple crossbar switches connected in parallel [119], and implemented in cross-connect systems [6, 100]. The drawback of this solution is that an additional mechanism is needed to preserve the cell sequence. Another solution of multiple crossbar implementation with no speedup and which preserves cell sequence was described in [127, 130]. Crosspoint buffering is also one of solutions for improving the throughput of packet switches. Implementation of the crossbar switch with crosspoint buffering was reported in some papers [125, 126, 128].

Crossbar architecture is also often used in optical switches design and implemented in different technologies. Many papers report such designs [59]. Recently a 4×4 crossbar switch implementation in InGaAsP-InP technology was described [34]. This switch supports 10 Gbit/s line speed and power losses less than 1 dB. Crosstalk is lower than -60 dB. Switches based on MEMS technology was also reported in many papers. This technology enables large optical switches with low losses and crosstalk to be built. For instance switch with 1296 ports, up to 6 dB insertion loss and worst case crosstalk not greater than -38 dB was reported in [144]. This switch was implemented in prototype switching fabric with 2.07 Petabit/s capacity, 1.6 Tbit/s capacity of each port carrying 40 DWDM channels, 40 Gbit/s each. Another experimental switch was reported in [98].

The implementation of triangular switch using MEMS technology was described in [147]. The 4×4 switch is shown in Fig. 3.29. In this implementation both sides of mirrors are covered with the reflective material, so when a mirror is in up position two beams, from the top and left side, are reflected. In the example shown in Fig. 3.29 following permutation is set up:

$$\Pi = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 3 & 0 & 1 \end{pmatrix}, \tag{3.5}$$

and respective connections are shown in different lines, while mirrors in the up position are drawn in bold lines.

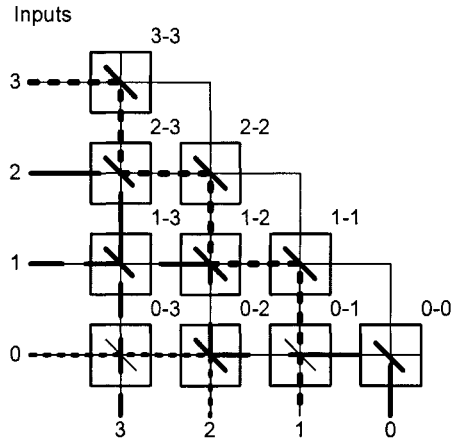


Figure 3.29. Implementation of the 4 × 4 triangular switch in MEMS technology

The tree-type 8×8 switch architecture fabricated in MEMS technology was presented in [42]. Implementation of tree-structured 1 × 8 switch in InGaAsP-InP technology was described in [145]. The same technology was used then to fabricate 8 × 8 switch [169].

Implementation of banyan-type switching fabric composed of directional couplers and fabricated using lithium niobate technology was reported in [117]. The switching fabric is composed of two types switches. One switch is the 1 × 8 tree-structured switch (two such switches were implemented in one integrated circuit) and the other is 16 × 16 baseline switch. The whole switching fabric is shown in Fig. 3.30. It contains 7 baseline switches; therefore one output of each 1 × 8 switch is left unused. This architecture will be discussed in Chapter 6. Another switch of capacity 32 × 32 and banyan architecture fabricated in lithium niobate technology was reported in [124]. Implementation of 1024 × 1024 switching systems using three-dimensional layout of banyan-type switching fabric was described in [179]. Recently a 4 × 4 switch was reported, which supports the speed of up to 160 Gbit/s at each input/output port [178]. This switch uses 2 × 2 switching elements arrange in banyan-type architecture and was implemented using SFQ (single flux quantum) technology.

Other architectures of switching fabrics were also considered and fabricated in different technologies [13, 135, 148].

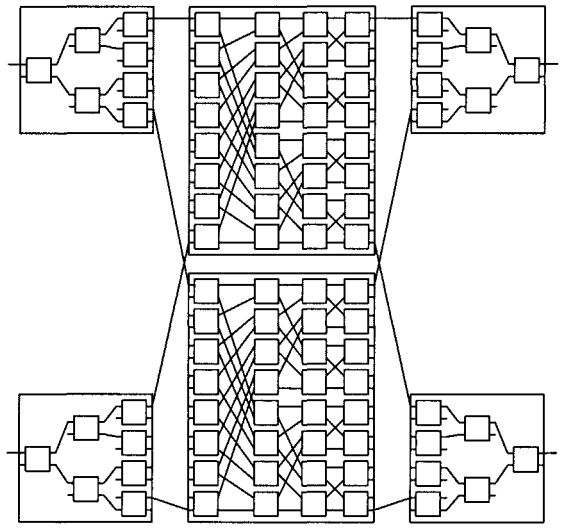


Figure 3.30. Implementation of the 16 × 16 switching fabric using baseline and tree topologies

Chapter 4

TWO-STAGE SWITCHING FABRICS

4.1 Two-sided Switching Fabrics

The general architecture of the space-division v-dilated two-sided two-stage switching fabric is shown in Fig. 4.1. It consists of r_1 switches of capacity $n_1 \times r_2 v$ at the first stage and r_2 switches of capacity $r_1 v \times n_2$ at the second stage. The switching fabric is fully accessible, but the performance of this switch is very low for $v = 1$ due to the high blocking probability. Only one connection can be set up from any input of switch I_i, $1 \leqslant i \leqslant r_1$, to any output of switch O_j, $1 \leqslant j \leqslant r_2$. The lower blocking probability can be obtained by increasing v. For strict-sense nonblocking operation of this switching fabric the following condition should be fulfilled:

THEOREM 4.1 *The space-division v-dilated two-sided two-stage switching fabric is strict-sense nonblocking if and only if:*

$$v = \min\{n_1; n_2\}. \tag{4.1}$$

This condition is obvious, since not more than $\min\{n_1; n_2\}$ connections can be set up between any pair of the first and the second stage switches. However, the number of crosspoints required in the switches is v times greater than in 1-dilated switching fabric. For the symmetrical switching fabric with $n_1 = n_2 = n$ and $r_1 = r_2 = r$ the number of crosspoints is two times higher than in the crossbar switch of the same capacity [138]. Therefore, two-sided two-stage switching fabrics did not receive more attention between researches and engineers.

Considerations given above are true not only for space-division switching fabrics, but also for time-division and multirate switching fabrics. Also, time-division two-stage switching fabrics composed of time and

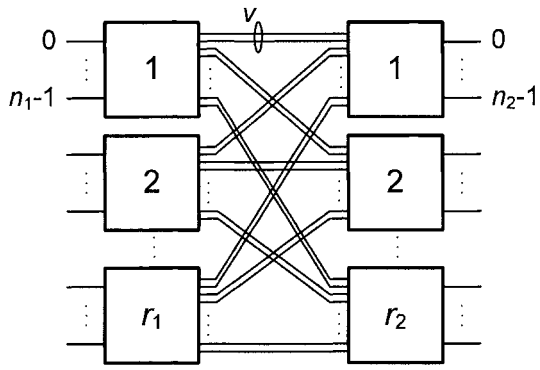


Figure 4.1. The structure of v-dilated two-sided two-stage switching fabric

space division switches were used in some practical implementations [25], [159]; however, the nonblocking two-stage switching fabric makes also no practical sense, since its capacity is no greater than the capacity of a single time and space switch. If each input link carries f_1 channels and each output link carries f_2 channels, than the two-stage switching fabric will be strict-sense nonblocking if $vf_0 = \min\{n_1 f_1; n_2 f_2\}$, where f_0 denotes the number of channels in each of interstage links [62].

4.2 One-sided Switching Fabrics

4.2.1 Space-division Switching

4.2.1.1 Switching Fabric's Architecture

The general architecture of the space-division v-dilated one-sided two-stage switching fabric composed of triangular switches is shown in Fig. 4.2. It includes r switches with $n+mv$ terminals at the first stage and m switches with rv terminals at the second stage. In each of the first stage switches n terminals constitute the switching fabric terminals and mv terminals are used to connect these first stage switches with the second stage switches. Each of the second stage switches has the capacity of rv terminals. Numbers n, m, r, and v determines the architecture of this switching fabric and it will be denoted by $T_{SD}(n, m, r, v)$.

4.2.1.2 Path Searching Algorithms

When a new call arrives at the switching fabric, a connection is to be set up between the two terminals requested. Let terminals of this switching fabric be numbered form 0 to $N-1$, $N = nr$; let the first stage switches be numbered from 1 to r; and let the second stage switches be

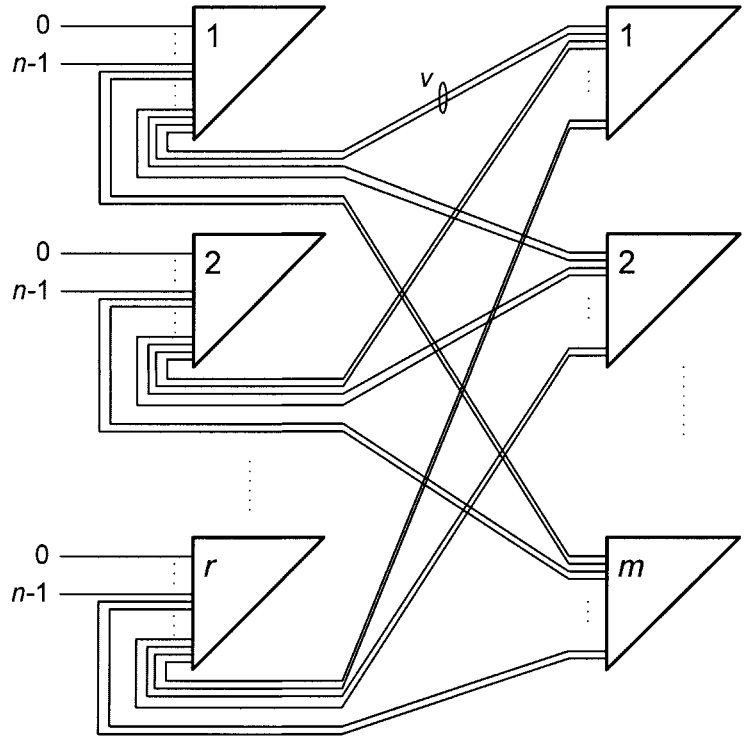


Figure 4.2. The structure of the v-dilated one-sided two-stage switching fabric composed of triangular switches

numbered from 1 to m. The first stage switch i will be denoted by I_i and the second stage switch k will be denoted by M_k. Let (I_i, I_j) denotes a new connection request from any terminal of switch I_i to any terminal of switch I_j, $1 \leqslant i, j \leqslant r$. The connecting path consists of the terminal in switch I_i, switch I_i, interstage link from switch I_i to the second stage switch M_k, $1 \leqslant k \leqslant m$, switch M_k, interstage link from switch M_k to switch I_j, switch I_j, and terminal of the first stage switch I_j.

To set up the new connection the second stage switch M_k with free links to switches I_i and I_j is to be found. This switch M_k is called free or available for the new connection. The following path searching algorithms can be used for finding an available second stage switch:

ALGORITHM 4.1 *Random*

Check second stage switches randomly and set up the connection through the first available switch.

ALGORITHM 4.2 *Sequential*

Check second stage switches sequentially starting form the second stage switch M_k, $1 \leqslant k \leqslant m$ and choose the first available switch.

ALGORITHM 4.3 *Minimum index*

The same as sequential, but $k = 1$.

ALGORITHM 4.4 *Quasi-random*

The same as sequential, but $k = l+1$, where M_l denotes the switch used to route the last request ($k = 1$ for $l = m$). This algorithm is also called *cyclic dynamic* or *round-robin*).

ALGORITHM 4.5 *Cyclic static*

The same as Quasi-random, but $k = l$, where M_l denotes the switch used to route the last request ($k = 1$ for $l = m$).

ALGORITHM 4.6 *Save the unused*

Do not route a new connection through any empty second stage switch unless there is no choice.

ALGORITHM 4.7 *Packing*

Route a new connection through the busiest but available second stage switch.

In all these algorithms when all second stage switches were checked and the switch for the new connection was not found, the connection is blocked.

4.2.1.3 Strict-sense Nonblocking Conditions

The conditions for strict-sense nonblocking operation of $T_{SD}(n, m, r, 1)$ under unicast connections are given by the following theorem [23, 70, 141].

THEOREM 4.2 *$T_{SD}(n, m, r, 1)$ is nonblocking in the strict sense if and only if:*

$$m \geqslant 2n - 1, \qquad \text{for } r > 3; \tag{4.2}$$

$$m \geqslant \left\lfloor \frac{3n}{2} \right\rfloor, \qquad \text{for } r = 3; \tag{4.3}$$

$$m \geqslant n, \qquad \text{for } r = 2. \tag{4.4}$$

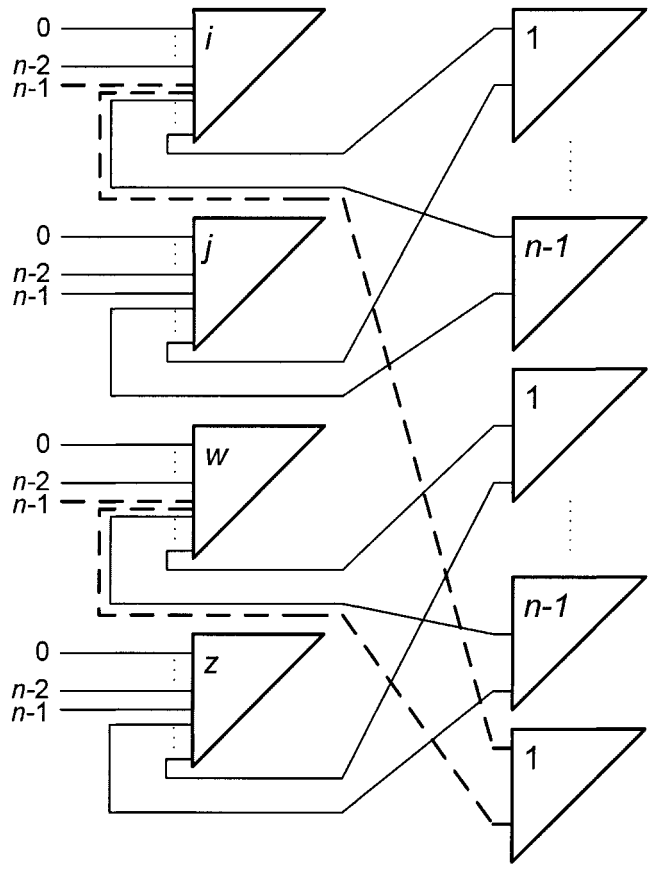


Figure 4.3. The worst state of the one-sided switching fabric composed of triangular switches with $r > 3$

Proof. Sufficiency can be proved by showing the worst state in the switching fabric.

Case 1: $r > 3$. In switch I_i, $1 \leqslant i \leqslant r$ there may be at most $n - 1$ connections to terminals in switch I_j, $1 \leqslant j \leqslant r$, $i \neq j$. These connections will occupy $n - 1$ second stage switches. Similarly, to switch I_w, $1 \leqslant w \leqslant r$ $w \neq j \neq i$ there may be at most $n - 1$ connections to terminals in the first stage switch I_z, $1 \leqslant z \leqslant r$, $z \neq w \neq j \neq i$. These connections may occupy another set of $n - 1$ second stage switches. In the worst case these sets of switches are disjoint and one more switch is needed to set up connection (I_i, I_w). Thus, at least $m = (n - 1) + (n - 1) + 1 = 2n - 1$ switches are needed in the second stage. This state is shown in Fig. 4.3.

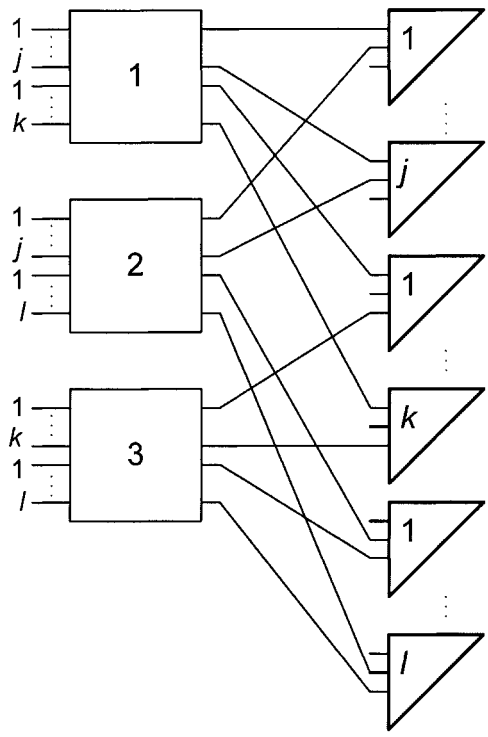


Figure 4.4. The worst state of the one-sided switching fabric composed of triangular switches with $r = 3$

Case 2: $r = 3$. We may have the following state:

- j connections between switches I_1 and I_2 through j second stage switches;
- k connections between switches I_1 and I_3 through k second stage switches;
- l connections between switches I_2 and I_3 through l second stage switches.

This state is acceptable if $j + k \leqslant n$, $k + l \leqslant n$, and $j + l \leqslant n$. Thus we have: $2j + 2k + 2l \leqslant 3n$, $j + k + l \leqslant 3n/2$. When $3n/2$ is not an integer, the result is rounded to the nearest integer less than $3n/2$. This state is shown in Fig. 4.4. For simplicity the first stage switches are drawn as squares.

Case 3: $r = 2$. We have n links between the first stage and the second

stage switches, and through these n links all n terminals of switch I_1 can be connected with all n terminals of switch I_2.

Necessity can be proved by showing the blocking state when less second stage switches are used than the number given by Theorem 4.2. We will consider only the case when $r > 3$. Set of events for $r \leqslant 3$ can be derived by analogy. Let assume that $m = 2n - 2$ and the second stage switches are chosen cyclicly starting from the switch used for setting up the previous connection (algorithm *cyclic static*). Let us also assume that there is not any connection in the switching fabric at the beginning, and path searching for the first connection will start from the second stage switch 1. The following set of events lead to the blocking state:

Step 1 Set up n connections (I_1, I_2). These connections will occupy switches from M_1 to M_n.

Step 2 Set up connection (I_3, I_4). This connection will be set up through switch M_n.

Step 3 Set up $n-2$ connections (I_3, I_4). These connections will be set up through switches from M_{n+1} to M_{2n-2}.

Step 4 Disconnect connection (I_1, I_2) which was set up through switch M_n.

At the end of step 4 there are $n-1$ connections (I_1, I_2) set up through switches from M_1 to M_{n-1} and $n-1$ connections (I_3, I_4) which uses switches from M_n to M_{2n-2}. In this state connection (I_1, I_3) is blocked and one more switch is the second stage is needed. □

4.2.1.4 Wide-sense Nonblocking Conditions

One-sided two-stage space-division switching fabrics composed of triangular switches nonblocking in the wide-sense were not considered in the literature. However, similar results as for two-sided three stage switching fabrics can be derived (see chapter 5.1.4). It is easy to show that when the quasi-random algorithm is used the switching fabric requires the number of second stage switches given by Theorem 4.2. For sequential and minimum index strategies following theorem can be proved.

THEOREM 4.3 *$T_{SD}(n, m, r, 1)$ is wide-sense nonblocking under sequential routing and minimum index routing for $r \geqslant 4$ if and only if: $m \geqslant 2n - 1$.*

Proof. We will consider only the minimum index algorithm, since for sequential routing the procedure for constructing the set of connections

and disconnections is similar, and it only needs renumbering of the second stage switches. It is sufficient to show the necessary condition for $r = 4$ because the set of events leading to the occupancy of $2n-1$ second stage switches can be the same if $r > 4$. This set of events is as follows.

Step 1 Set up two connections (I_1, I_2) through the switches M_1 and M_2. Disconnect connection (I_1, I_2) from switch M_1 and set up connection (I_3, I_4). This connection will use switch M_1. The state of the switching fabric is:

connection	the second stage switch
(I_3, I_4)	M_1
(I_1, I_2)	M_2

Step 2 Set up two connections (I_1, I_3) through the second stage switches M_3 and M_4. Disconnect connection (I_1, I_3) realized through switch M_3 and set up connection (I_2, I_4). This connection will use switch M_3. Disconnect connection (I_1, I_2) from switch M_2 and set up connection (I_1, I_3) through this switch. Disconnect (I_3, I_4) from switch M_1 and set up (I_2, I_4) through this switch. The state of the switching fabric is changed to:

connection	the second stage switch
(I_2, I_4)	M_1
(I_1, I_3)	M_2
(I_2, I_4)	M_3
(I_1, I_3)	M_4

Assume that after step $i-1$ following state is in the switching fabric:

connection	the second stage switch
(I_2, I_4)	M_1
(I_1, I_3)	M_2
(I_2, I_4)	M_3
(I_1, I_3)	M_4
...	...
(I_2, I_4)	M_{2i-3}
(I_1, I_3)	M_{2i-2}

Step *i* Set up two connections (I_1, I_2) through the second stage switches M_3 and M_4. Disconnect connection (I_1, I_2) from switch M_{2i-1} and set up connection (I_3, I_4). This connection will use switch M_{2i-1}. Then do:
Disconnect (I_1, I_3) from M_{2i-2} and set up (I_1, I_2) through this switch.
Disconnect (I_2, I_4) from M_{2i-3} and set up (I_3, I_4) through this switch.
Disconnect (I_1, I_3) from M_{2i-4} and set up (I_1, I_2) through this switch.
Disconnect (I_2, I_4) from M_{2i-5} and set up (I_3, I_4) through this switch.

...
Disconnect (I_1, I_3) from M_4 and set up (I_1, I_2) through this switch.
Disconnect (I_2, I_4) from M_3 and set up (I_3, I_4) through this switch.
Disconnect (I_1, I_3) from M_2 and set up (I_1, I_2) through this switch.
Disconnect (I_2, I_4) from M_1 and set up (I_3, I_4) through this switch.
The state of the switching fabric is changed to:

connection	the second stage switch
(I_3, I_4)	M_1
(I_1, I_2)	M_2
(I_3, I_4)	M_3
(I_1, I_2)	M_4
...	...
(I_3, I_4)	M_{2i-3}
(I_1, I_2)	M_{2i-2}
(I_3, I_4)	M_{2i-1}
(I_1, I_2)	M_{2i}

During execution of step i no more than $i + 1$ connections were set up in any of the first stage switches. Step i can be executed until $i = n - 1$. After step $n - 1$ we have $n - 1$ connections (I_1, I_2) and $n - 1$ connections (I_3, I_4) (or (I_1, I_3) and (I_2, I_4), depending on n being odd or even) which are set up through $2n - 2$ different second stage switches. Connection (I_1, I_4) has to be set up through switch M_{2n-1}. □

The example of obtaining the state given in the proof of Theorem 4.3 in the $T_{SD}(4, 7, 4, 1)$ is given in Table 4.1. Connections which are in the switching fabric at the end of each step are marked by asterisk. At the end of step 3 second stage switches from M_1 to M_6 are inaccessible by connection (I_1, I_4) and switch M_7 is to be used.

For packing strategy the question is which switch should be used from several equally loaded switches? When the switch recently used to set up the previous connection is used, the following theorem holds.

THEOREM 4.4 *$T_{SD}(n, m, r, 1)$ is wide-sense nonblocking for $r \geqslant 5$ and under packing routing algorithm with recently used switch preferences in case more switches are equally loaded, if and only if: $m \geqslant 2n - 1$.*

Proof. The condition $m \geqslant 2n - 1$ guarantees wide-sense nonblocking condition since it is also a strict-sense nonblocking condition. We prove now the necessary condition for $r = 5$ (this condition is then obviously true also for $r > 5$) by giving a sequence of events leading to the occupancy of $2n - 1$ second stage switches.

Step 1 Set up $n - 1$ connections (I_1, I_2). These connections will occupy switches from M_1 to M_{n-1}.

Step	action	connection	the second stage switch
1	connect	(I_1, I_2)	M_1
	connect	$(I_1, I_2)^*$	M_2
	disconnect	(I_1, I_2)	M_1
	connect	$(I_3, I_4)^*$	M_1
2	connect	(I_1, I_3)	M_3
	connect	$(I_1, I_3)^*$	M_4
	disconnect	(I_1, I_3)	M_3
	connect	$(I_2, I_4)^*$	M_3
	disconnect	(I_1, I_2)	M_2
	connect	$(I_1, I_3)^*$	M_2
	disconnect	(I_3, I_4)	M_1
	connect	$(I_2, I_4)^*$	M_1
3	connect	(I_1, I_2)	M_5
	connect	$(I_1, I_2)^*$	M_6
	disconnect	(I_1, I_2)	M_5
	connect	$(I_3, I_4)^*$	M_5
	disconnect	(I_1, I_3)	M_4
	connect	$(I_1, I_2)^*$	M_4
	disconnect	(I_2, I_4)	M_3
	connect	$(I_3, I_4)^*$	M_3
	disconnect	(I_1, I_3)	M_2
	connect	$(I_1, I_2)^*$	M_2
	disconnect	(I_2, I_4)	M_1
	connect	$(I_3, I_4)^*$	M_1

Table 4.1. The state in $T_{SD}(4, 7, 4, 1)$ with minimum index routing algorithm

Step 2 Repeat $n - 1$ times steps 2.1 - 2.5.

Step 2.1 Set up connection (I_2, I_3). This connection will have to occupy switch M_{n+i-1}

Step 2.2 Set up connection (I_1, I_5). Since each of all not empty second stage switches carries 1 connection, this connection will be set up through switch M_{n+i-1}.

Step 2.3 Disconnect connection (I_2, I_3).

Step 2.4 Set up connection (I_3, I_4). Again, each of all not empty second stage switches carries 1 connection, this connection will be set up through switch M_{n+i-1}.

Step 2.5 Disconnect connection (I_1, I_5).

After these steps we have $n-1$ connections (I_1, I_2) and $n-1$ connections (I_3, I_4) which are set up through $2n - 2$ different second stage switches.

Connection (I_1, I_3) will have to occupy the second stage switch M_{2n-1}. □

4.2.1.5 Rearrangeable Switching Fabrics

Rearrangeable one-sided switching fabrics composed of triangular switches were considered by Bassalygo, Grushko, and Neiman [7]. They proved following theorem:

THEOREM 4.5 *$T_{SD}(n, m, r, 1)$ is rearrangeable nonblocking for $r \geqslant 3$ if and only if:*

$$m \geqslant \left\lfloor \frac{3n}{2} \right\rfloor . \tag{4.5}$$

Proof. The necessity can be proved by giving the set of connections similar to that in proof of Theorem 4.2 for $r = 3$. We may have $\lfloor (n+1)/2 \rfloor$ connections (I_1, I_2), $n - \lfloor (n+1)/2 \rfloor$ connections (I_1, I_3), and $n - \lfloor (n+1)/2 \rfloor$ connections (I_2, I_3), which will occupy $\lfloor 3n/2 \rfloor$ second stage switches and no rearrangements can be done to reduce this number of required switches.

Sufficiency can be proved using connection matrix $\mathbf{H}_r^{(n)}$, which will be described in more detail in section 5.1.6.2 of chapter 5. Similarly as for two-sided three-stage switching fabrics, this matrix can be decomposed into n elementary permutation matrices $\mathbf{E}_r$. However, in the case of one-sided switching fabrics composed of triangular switches, connections corresponding to two matrices $\mathbf{E}_r$ requires three second stage switches. Therefore, $\lfloor 3n/2 \rfloor$ switches are sufficient to route all possible permutations. □

4.2.2 Time-division Switching

4.2.2.1 Switching Fabric's Architecture

The general architecture of the time-division v-dilated one-sided two-stage switching fabric composed of triangular switches is shown in Fig. 4.5. It differs from the space-division switching fabric shown in Fig. 4.2 in interstage links and terminals which are now TDM links. The first stage contains r switches with $n + mv$ terminals, n of these terminals carry f_1 time slots each, and mv terminals carry f_0 time slots. Terminals with f_1 time slots constitute the switching fabric's terminals. The other terminals are used to connect the first stage switches with the second stage switches. The second stage contains m switches, each of them having rv terminals with f_0 time slots. This switching fabric architecture will be denoted by $T_{TD}(n, f_1, m, r, v, f_0)$.

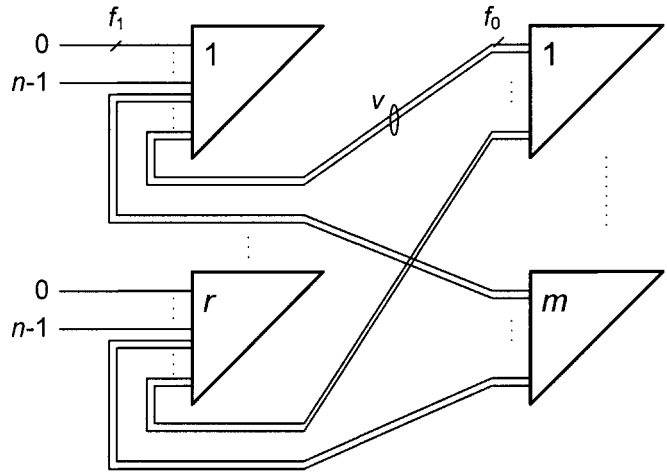


Figure 4.5. One-sided two-stage time-division switching fabric composed of triangular switches

Connections may be set up between any time slot of any terminal to another time slot of any terminal. When these time slots are in terminals of different first stage switches, the second stage switch is to be selected. Path searching algorithms are similar to those used in space-division switching fabrics, but apart from the second stage switch, time slots in two interstage links are to be chosen. In the set of v links between the first and the second stage switches, a time slot for the new connection may be selected

- randomly,
- sequentially,
- quasi-randomly,
- using the time slot recently released,
- using the time slot which is the longest time free,
- in the most loaded link in the set,
- using the most unloaded link in the set.

When a new connection is an s-slot connection, s slots may be selected using fixed, floating, or flexible assignment, which were described in section 2.5.2 of chapter 2. These s slots are to be selected in one TDM link. When a new connection requires all time slots in one TDM link, any connection which use one time slot will make the TDM link inaccessible by this new connection. As will be shown later, switching fabrics with s-slot connections for which s may vary in range from 1 to f_1 will require a high number of the second stage switches for nonblocking

operation. Therefore, another routing strategy was proposed for using in such connections [120]:

- *Functional division of second stage switches*: Second stage switches are divided into two subsets of m_1 and m_2 switches, $m_1 + m_2 = m$. The set of m_1 switches is used to route connections with $s \leqslant \lfloor f_0/2 \rfloor$, and the set of m_2 switches is used to route connections with $s > \lfloor f_0/2 \rfloor$.

In each of these sets connection may be set up using one of earlier described algorithms for switch selection and for time-slot assignment.

4.2.2.2 Strict-sense Nonblocking Conditions

Strict-sense nonblocking conditions of $T_{TD}(n, f_1, m, r, v, f_0)$ for unicast 1-slot connections where considered by Jajszczyk [63]. He proved the following theorem.

THEOREM 4.6 *$T_{TD}(n, f_1, m, r, v, f_0)$ is nonblocking in the strict sense for 1-slot connections if and only if:*

$$m \geqslant 2\left\lfloor \frac{nf_1 - 1}{vf_0} \right\rfloor + 1, \qquad \text{for } r > 3; \tag{4.6}$$

$$m \geqslant \left\lfloor \frac{3nf_1 - 2}{2vf_0} \right\rfloor + 1, \qquad \text{for } r = 3 \text{ and } nf_1 > vf_0; \tag{4.7}$$

$$m \geqslant \left\lfloor \frac{nf_1 - 1}{vf_0} \right\rfloor + 1, \qquad \text{for } r = 2. \tag{4.8}$$

Proof. When $r > 3$ there may be not more than $nf_1 - 1$ connections in each of switches I_i and I_j, $i \neq j$, and there is no connection (I_i, I_j). Connections from each of these switches will use all time slots in sets of v links connected to at most $\lfloor (nf_1 - 1)/vf_0 \rfloor$ second stage switches. In the worst scenario connections from I_i and I_j are set up through disjoint sets of the second stage switches and one more switch is needed for connection (I_i, I_j). The necessity can be proved by analogy to the space division case (Theorem 4.2). For $r = 3$ the set of connections which may block connection (I_i, I_j), $i \neq j$, is also similar to the space-division case with $r = 3$. We may have k_1 connections (I_i, I_j), k_2 connections (I_i, I_l), and k_3 connections (I_j, I_l). Numbers k_1, k_2, and k_3 must fulfil following conditions:

$k_1 + k_2 = nf_1 - 1$ (one time slot free in I_i),
$k_1 + k_3 = nf_1 - 1$ (one time slot free in I_j),
$k_2 + k_3 \leqslant nf_1$ (switch I_l has nf_1 time slots).

We have not more than $k_1+k_2+k_3 \leqslant (3nf_1-2)/2$ connections and they will occupy at most $(k_1+k_2+k_3)/vf_0$ switches. When $(3nf_1-2)/2vf_0$ is an integer one more switch is needed for connection (I_i, I_j), otherwise $\lfloor (3nf_1-2)/2vf_0 \rfloor$ switches will be inaccessible by this connection and the additional switch will be partially occupied but still accessible by (I_i, I_j). For $r=2$ the worst case is when all time slots of terminals in I_1 are to be connected with time slots of terminals in I_2 and through one second stage switch vf_0 connections (I_1, I_2) can be set up. □

COROLLARY 4.7 *The $T_{SD}(n,m,r,v)$ is nonblocking in the strict sense if and only if:*

$$m \geqslant 2\left\lfloor \frac{n-1}{v} \right\rfloor + 1, \qquad \text{for } r > 3; \tag{4.9}$$

$$m \geqslant \left\lfloor \frac{3n-2}{2v} \right\rfloor + 1, \qquad \text{for } r = 3 \text{ and } n > v; \tag{4.10}$$

$$m \geqslant \left\lfloor \frac{n-1}{v} \right\rfloor + 1, \qquad \text{for } r = 2. \tag{4.11}$$

Proof. Set $f_1 = f_0 = 1$ in inequalities (4.6), (4.7), and (4.8). □

Conditions under which $T_{TD}(n, f_1, m, r, v, f_0)$ is strict-sense nonblocking for s-slot connections where considered in [71, 72]:

THEOREM 4.8 *$T_{TD}(n, f_1, m, r, v, f_0)$ is nonblocking in the strict sense for s-slot connections, $1 \leqslant s \leqslant s_{\max} \leqslant f_1$, if and only if:*

$$m \geqslant 2 \max_{1 \leqslant s \leqslant s_{\max}} \left\lfloor \frac{nf_1 - s}{v \left\lfloor \frac{f_0}{s} \right\rfloor} \right\rfloor + 1, \quad \text{for } r \geqslant 4; \tag{4.12}$$

$$m \geqslant \max_{1 \leqslant s \leqslant s_{\max}} \left\lfloor \frac{3nf_1 - 2s}{2v \left\lfloor \frac{f_0}{s} \right\rfloor} \right\rfloor + 1, \quad \text{for } r = 3; \tag{4.13}$$

$$m \geqslant \max_{1 \leqslant s \leqslant s_{\max}} \left\lfloor \frac{nf_1 - s}{v \left\lfloor \frac{f_0}{s} \right\rfloor} \right\rfloor + 1, \qquad \text{for } r = 2. \tag{4.14}$$

Proof. *Case 1, $r \geqslant 4$.* Necessity will be proved by giving the set of events which results in occupancy of all of the second stage switches

given by inequality (4.12). The path for a new s-slot connection will be set up using s consecutive free slots (floating assignment). If the new connection is from the same first stage switch from which the last connection was set up, searching will start from the time slot next to the last used. If the last used time slot is the last time slot in the link, searching will start from the next link. This may be the next link in the set of v links to the same second stage switch or the first link to the next second stage switch. When the new s-slot connection is from another first stage switch, searching will start from the second stage switch next to that recently used for previous connection. We will also assume that if connection (I_i, I_j, s) is set up through switch M_1, then connection (I_j, I_i, s) for the second direction of connection is also set up through this switch. Let as assume that equation (4.12) reaches maximum for $s = s_1$.

Step 1 Set up connection $(I_i, I_j, (s_1 - 1))$, $i \neq j$, and then set up connection $(I_i, I_j, 1)$. Disconnect connection $(I_i, I_j, (s_1 - 1))$.

Step 2 Repeat events from step 1 $\lfloor f_0/s_1 \rfloor$ times. If $f_0 \oplus s_1 = s_1 - 1$[1] set up one additional connection $(I_i, I_j, (s_1 - 1))$. After step 2 in the link to the second stage switch $\lfloor f_0/s_1 \rfloor$ time slots will be occupied by 1-slot connections, and there will be exactly $s_1 - 1$ free time slots between occupied time slots.

Step 3 Set up connection $(I_i, I_j, (s_1 - 1))$. This connection will use the next link to the same second stage switch, if $v > 1$, or link to the next second stage switch, if $v = 1$. If $f_0 \oplus s_1 = s_1 - 1$, disconnect the additional connection $(I_i, I_j, (s_1 - 1))$ and set up connection $(I_i, I_j, 1)$. Disconnect connection $(I_i, I_j, (s_1 - 1))$.

Step 4 Repeat events from step 3 $\lfloor f_0/s_1 \rfloor$ times. If $f_0 \oplus s_1 = s_1 - 1$ set up one additional connection $(I_i, I_j, (s_1 - 1))$.

Step 5 Repeat events from steps 3 and 4 up until all links to $\lfloor (nf_1 - s_1)/(v\lfloor f_0/s_1 \rfloor) \rfloor$ second stage switches will be occupied.

Step 6 Set up connection $(I_w, I_z, (s_1 - 1))$, $w \neq z$, $z \neq j$, $w \neq i$, and then set up connection $(I_w, I_z, 1)$. Disconnect connection $(I_w, I_z, (s_1 - 1))$.

Step 7 Repeat events from step 6 $\lfloor f_0/s_1 \rfloor$ times. If $f_0 \oplus s_1 = s_1 - 1$ set up one additional connection $(I_w, I_z, (s_1 - 1))$.

[1] $x \oplus y$ denotes the rest from dividing x by y

Step 8 Set up connection $(I_w, I_z, (s_1 - 1))$. If $f_0 \oplus s_1 = s_1 - 1$, disconnect the additional connection $(I_w, I_z, (s_1 - 1))$ and set up connection $(I_w, I_z, 1)$. Disconnect connection $(I_w, I_z, (s_1 - 1))$.

Step 9 Repeat events from step 8 $\lfloor f_0/s_1 \rfloor$ times. If $f_0 \oplus s_1 = s_1 - 1$ set up one additional connection $(I_w, I_z, (s_1 - 1))$.

Step 10 Repeat events from steps 8 and 9 up until all links to $\lfloor (nf_1 - s_1)/(v\lfloor f_0/s_1 \rfloor) \rfloor$ second stage switches will be occupied.

There are $2\lfloor (nf_1 - s_1)/(v\lfloor f_0/s_1 \rfloor) \rfloor$ second stage switches inaccessible by the connection (I_i, I_w, s_1) and one more second stage switch is needed.

Sufficiency will be proved by showing the worst state of the switching fabric. Assume that (I_i, I_j, s), $i \neq j$, is compatible with the state of the switching fabric. There may be at most $nf_1 - s$ 1-slot connections from terminals of switch I_i, and these connections may occupy links to the second stage switches in such a way that there are no more than $s - 1$ consecutive free time slots. Such links are inaccessible by the new connection if fixed or floating assignment is used. These connections will occupy $\lfloor (nf_1 - s)/(v\lfloor f_0/s \rfloor) \rfloor$ second stage switches. The same number of the second stage switches may be occupied by $nf_1 - s$ 1-slot connections from terminals of switch I_j. If these sets of switches are disjoint one more switch is required by connection (I_i, I_j, s). Thus, at least $2\lfloor (nf_1 - s)/(v\lfloor f_0/s \rfloor) \rfloor + 1$ second stage switches are needed, and this number must be maximized through all possible s.

Case 2, $r = 3$. Necessity. We will start to search the path for the new connection from the time slot next to the time slot used by the previous connection set up in the switch with the new request. Let's assume that equation 4.13 reaches maximum for $s = s_1$. Events resulting in occupancy of $\lfloor (3nf_1 - 2s)/(2v\lfloor f_0/s \rfloor) \rfloor + 1$ second stage switches is:

Step 1 Repeat $\lfloor nf_1/2 \rfloor$ times the set of events: set up connection $(I_1, I_2, (s_1 - 1))$, connect $(I_1, I_2, 1)$, and disconnect $(I_1, I_2, (s_1 - 1))$. Then, if $f_0 \oplus s_1 = s_1 - 1$, set up two additional connections $(I_1, I_2, (s_1 - 1))$ and disconnect them (this additional connection has to be set up also in the next steps).

Step 2 Repeat $\lfloor (nf_1 - 2s_1)/2 \rfloor$ times the set of events: connect $(I_2, I_3, (s_1 - 1))$, connect $(I_2, I_3, 1)$, and disconnect $(I_2, I_3, (s_1 - 1))$.

Step 3 Repeat s_1 times the set of events: connect $(I_3, I_1, (s_1 - 1))$, connect $(I_3, I_1, 1)$, and disconnect $(I_3, I_1, (s_1 - 1))$.

Step 4 Repeat $\lceil (nf_1 - 2s_1)/2 \rceil$ times the set of events: connect $(I_3, I_1, (s_1 - 1))$, connect $(I_3, I_1, 1)$, and disconnect $(I_3, I_1, (s_1 - 1))$.

We have now $\lfloor nf_1/2 \rfloor$ connections $(I_1, I_2, 1)$, $\lfloor nf_1/2 \rfloor$ connections $(I_2, I_3, 1)$, and $\lceil (nf_1 - 2s_1)/2 \rceil$ connections $(I_3, I_1, 1)$. These connections are set up in such a way that there are at most $s_1 - 1$ free time slots between them, and they occupy $\lfloor m_1 \rfloor$ second stage switches, where

$$m_1 = \frac{\left\lceil \frac{nf_1 - 2s_1}{2} \right\rceil + 2 \left\lfloor \frac{nf_1}{2} \right\rfloor}{v \left\lfloor \frac{f_0}{s_1} \right\rfloor} = \frac{3nf_1 - 2s_1}{2v \left\lfloor \frac{f_0}{s_1} \right\rfloor}, \tag{4.15}$$

when m_1 is an integer, or $\lfloor m_1 \rfloor + 1$ switches, when m_1 in not an integer. Connection (I_1, I_3, s_1) requires one more switch in the second stage if m_1 is an integer. When m_1 in not an integer, this connection can be set up through one of $\lfloor m_1 \rfloor + 1$ switches. In both cases $\lfloor m_1 \rfloor + 1$ switches are needed.

Sufficiency. Let assume that connection (I_i, I_j, s) is to be set up, $i \neq j$. Let k_1 denote the number of existing connections $(I_i, I_j, 1)$; let k_2 denote the number of existing connections $(I_i, I_l, 1)$, $i \neq l$; and let k_3 denote the number of existing connections $(I_j, I_l, 1)$, $j \neq l$. This state is acceptable if

$k_1 + k_2 = nf_1 - s$, since there are s free time slots in I_i,
$k_1 + k_3 = nf_1 - s$, since there are s free time slots in I_j,
$k_2 + k_3 \leqslant nf_1$, since there are only nf_1 time slots in I_3.

After solving these three equations we obtain $k_1 = \lceil (nf_1 - 2s)/2 \rceil$ and $k_3 = k_2 = \lfloor nf_1)/2 \rfloor$. In interstage links $\lfloor f_0/s \rfloor$ time slots are occupied and these links are inaccessible for the new s-slot connection. In the second stage m_1 switches are inaccessible if m_1 is an integer, or $\lfloor m_1 \rfloor + 1$ switches, if m_1 is not an integer $(m_1 = (k_1 + k_2 + k_3)/(v \lfloor f_0/s \rfloor))$. Finally we obtain:

$$m_1 = \frac{\left\lceil \frac{nf_1 - 2s}{2} \right\rceil + 2 \left\lfloor \frac{nf_1}{2} \right\rfloor}{v \left\lfloor \frac{f_0}{s} \right\rfloor} = \frac{3nf_1 - 2s}{2v \left\lfloor \frac{f_0}{s} \right\rfloor}. \tag{4.16}$$

Taking into account that m_1 must be maximized through all s we get equation (4.13).

Case 3, $r = 2$. Necessity can be proved by setting up connection $(I_1, I_2, (s_1 - 1)$, connection $(I_1, I_2, 1)$ and disconnecting $(I_1, I_2, (s_1 - 1)$ and taking into account the case when $f_0 \oplus s_1 = s_1 - 1$. In the worst case we may have $nf_1 - s$ connections $(I_1, I_2, 1)$. So, $\lfloor (nf_1 - s)/(v \lfloor f_0/s \rfloor) \rfloor$ second stage switches are inaccessible by connection (I_1, I_2, s) and one

more second stage switch is needed. □

It can be seen that for $f_1 = f_0 = v = s = 1$, we obtain conditions for space-division switching fabrics given in Theorem 4.2. If only $s = 1$, we obtain the results for 1-slot connections given in Theorem 4.6.

4.2.2.3 Wide-sense Nonblocking Conditions

For $T_{TD}(n, f_1, m, r, v, f_0)$ with 1-slot connections similar conclusions may be derived as for space-division switching fabrics, i.e., all currently known algorithms will not result in the reduction of the second stage switches required. More promising results can be obtained when s-slot connections are realized in switching fabrics. The reduction in the number of second stage switches may be obtained either using more efficient time slot assignment, or using other algorithms for selecting switches for connections. Firstly we will give nonblocking conditions for switching fabrics with random assignment, which were earlier considered by Niestegge [120].

THEOREM 4.9 *$T_{TD}(n, f_1, m, r, v, f_0)$ is nonblocking in the wide sense for s-slot connections, $1 \leqslant s \leqslant s_{\max} \leqslant f_1$, under flexible assignment if and only if:*

$$m \geqslant 2\left\lfloor \frac{nf_1 - s_{\max}}{v\,(f_0 - s_{\max} + 1)} \right\rfloor + 1, \quad \text{for } r \geqslant 4; \tag{4.17}$$

$$m \geqslant \left\lfloor \frac{3nf_1 - 2s_{\max}}{2v\,(f_0 - s_{\max} + 1)} \right\rfloor + 1, \quad \text{for } r = 3; \tag{4.18}$$

$$m \geqslant \left\lfloor \frac{nf_1 - s_{\max}}{v\,(f_0 - s_{\max} + 1)} \right\rfloor + 1, \quad \text{for } r = 2. \tag{4.19}$$

Proof. *Case 1*, $r \geqslant 4$. Similarly as in previous theorems, necessity will be proved by giving the set of events which results in occupancy of all of the second stage switches given by inequality (4.17). If the new connection is from the same first stage switch the last connection was set up, selection of s slots for the new s-slot connection will start from the first time slot in the link recently selected for the previous connection. If there are not s slot free in this link the next link from the first stage switch will be checked. When the new s-slot connection is from another first stage switch, searching will start from the second stage switch next to that recently used for previous connection.

Step 1 Set up $f_0 - s_{\max} + 1$ connections $(I_i, I_j, 1)$, $i \neq j$.

Step 2 Set up connection $(I_i, I_j, (s_{\max} - 1))$.

Step 3 Set up connection $(I_i, I_j, 1)$, disconnect $(I_i, I_j, (s_{\max} - 1))$, and set up $f_0 - s_{\max}$ connections $(I_i, I_j, 1)$.

Step 4 Repeat connections and disconnections from steps 2 and 3 up until $v\,(f_0 - s_{\max} + 1)\,\lfloor (nf_1 - s_{\max})/(v\,(f_0 - s_{\max} + 1))\rfloor$ connections $(I_i,\ I_j, 1)$ will be set up in I_i. These connections will make $\lfloor (nf_1 - s_{\max})/(v\,(f_0 - s_{\max} + 1))\rfloor$ second stage switches inaccessible by the connection $(I_i, I_w, s_{\max})$.

Step 5 Set up $f_0 - s_{\max} + 1$ connections $(I_w, I_z, 1)$, $w \neq z$, $z \neq j$, $w \neq i$.

Step 6 Set up connection $(I_w, I_z, (s_{\max} - 1))$.

Step 7 Set up connection $(I_w, I_z, 1)$, disconnect $(I_w, I_z, (s_{\max} - 1))$, and set up $f_0 - s_{\max}$ connections $(I_w, I_z, 1)$.

Step 8 Repeat connections and disconnections from steps 6 and 7 up until $v\,(f_0 - s_{\max} + 1)\,\lfloor (nf_1 - s_{\max})/(v\,(f_0 - s_{\max} + 1))\rfloor$ connections $(I_w,\ I_z, 1)$ will be set up in I_w. These connections will make next $\lfloor (nf_1 - s_{\max})/(v\,(f_0 - s_{\max} + 1))\rfloor$ second stage switches inaccessible by the connection $(I_w, I_z, s_{\max})$.

There are $2\,\lfloor (nf_1 - s_{\max})/(v\,(f_0 - s_{\max} + 1))\rfloor$ second stage switches inaccessible by the connection $(I_i, I_w, s_{\max})$ and one more second stage switch is needed.

Sufficiency will be proved by showing the worst state of the switching fabric. Assume that (I_i, I_j, s), $i \neq j$ is compatible with the state of the switching fabric. There may be at most $nf_1 - s$ 1-slot connections from terminals of switch I_i, and these connections may occupy links to the second stage switches in such a way that not more than $s - 1$ time slots are free. Such links are inaccessible by the new connection. These connections will occupy $\lfloor (nf_1 - s)/(v\,(f_0 - s + 1))\rfloor$ second stage switches. The same number of the second stage switches may be occupied by $nf_1 - s$ 1-slot connections from terminals of switch I_j. If these sets of switches are disjoint, one more switch is required by connection (I_i, I_j, s). Thus, at least $2\,\lfloor (nf_1 - s)/(v\,(f_0 - s + 1))\rfloor$ second stage switches are needed. This number must be maximized through all possible s and it reaches maximum for $s = s_{\max}$.

Remaining cases can be proved by analogy to the proof of case 1 of this Theorem and cases 2 and 3 of Theorem 4.8. □

For $f_1 = f_0 = f$ and $v = 1$, we obtain the results given by Niestegge [120]. Theorem 4.9 includes also the known results for space-division switching and for time-division switching with 1-slot connections.

THEOREM 4.10 *$T_{TD}(n, f_1, m, r, v, f_0)$ is nonblocking in the wide sense for s-slot connections, $1 \leqslant s \leqslant s_{\max} \leqslant f_1$, under the algorithm with functional division of second stage switches if and only if:*

$$m \geqslant m_1 + m_2, \tag{4.20}$$

where

$$m_1 \geqslant 2\left\lfloor \frac{nf_1 - \left\lfloor \frac{f_0}{2} \right\rfloor}{v\left(f_0 - \left\lfloor \frac{f_0}{2} \right\rfloor + 1\right)} \right\rfloor + 1 \quad \text{and}$$

$$m_2 \geqslant 2\left\lfloor \frac{n-1}{v} \right\rfloor + 1, \qquad \text{for } r \geqslant 4; \tag{4.21}$$

$$m_1 \geqslant \left\lfloor \frac{3nf_1 - 2\left\lfloor \frac{f_0}{2} \right\rfloor}{2v\left(f_0 - \left\lfloor \frac{f_0}{2} \right\rfloor + 1\right)} \right\rfloor + 1 \quad \text{and}$$

$$m_2 \geqslant \left\lfloor \frac{3n-2}{2v} \right\rfloor + 1, \qquad \text{for } r = 3; \tag{4.22}$$

$$m_1 \geqslant \left\lfloor \frac{nf_1 - \left\lfloor \frac{f_0}{2} \right\rfloor}{v\left(f_0 - \left\lfloor \frac{f_0}{2} \right\rfloor + 1\right)} \right\rfloor + 1 \quad \text{and}$$

$$m_2 \geqslant \left\lfloor \frac{n-1}{v} \right\rfloor + 1, \qquad \text{for } r = 2; \tag{4.23}$$

m_1 denotes the number of second stage switches which serve s-slot connections, for which $1 \leqslant s \leqslant \lfloor f_0/2 \rfloor$, and m_2 denotes the number of second stage switches used for serving s-slot connections, for which $\lfloor f_0/2 \rfloor < s \leqslant f_1$.

Proof. For m_1 set $s_{\max} = \lfloor f_0/2 \rfloor$ in inequalities (4.17), (4.18), and (4.19). When $s > \lfloor f_0/2 \rfloor$ only one connection can be set up in one link, so it corresponds to space-division switching, i.e., for m_2 inequalities (4.9), (4.10), and (4.11) holds. □

It is clear that the algorithm with functional division of second stage switches may reduce the number of switches only if $s_{\max} > \lfloor f_0/2 \rfloor$. However, it must be checked if in this case the algorithm will result in switch savings. For instance $T_{TD}(8, 32, m, r, 1, 32)$ switching fabric will

require $m \geqslant 44$ switches when the algorithm with functional division is used, $m_1 = 29$ switches for connections with $s \leqslant 16$ and $m_2 = 15$ switches for connections with $s > 16$. However, when $s_{\max}$ for instance is limited to 21, is is sufficient to have $m = 39$ switches in the second stage for algorithm with flexible assignment of time slots.

4.2.2.4 Rearrangeable Switching Fabrics

Rearrangeable $T_{TD}(n, f_1, m, r, v, f_0)$ switching fabrics were considered by Jajszczyk [63] who proved conditions under which such switching fabrics with 1-slot connections are rearrangeable for vf_0 even:

THEOREM 4.11 *$T_{TD}(n, f_1, m, r, v, f_0)$ with 1-slot connections is rearrangeable for vf_0 even if and only if:*

$$m \geqslant 2 \left\lfloor \frac{nf_1 - 1}{vf_0} \right\rfloor + 1. \tag{4.24}$$

It should be noted that when $v = f_0 = f_1 = 1$ then vf_0 is not even and this theorem not include the space-division case. An example of the state of $T_{TD}(2, 2, 2, 4, 1, 2)$ is shown in Fig 4.6a. In this state two connections (I_1, I_2) are set up through switch M_1. These connections are numbered 1 and 2, and these numbers are placed in time slots. Other two connections (I_3, I_4) are set up through switch M_2 (these connections are indexed by numbers 3 and 4). In this state connection (I_1, I_3) is blocked, but the state can be rearranged to the state in which one connection (I_1, I_2) is moved form switch M_1 to switch switch M_2, and one connection (I_3, I_4) is moved from switch M_2 to switch M_1. Connection (I_1, I_3) may be now set up through any of switches M_1 and M_2. The state after these rearrangements is shown in Fig. 4.6b.

4.2.3 Multirate Switching

4.2.3.1 Switching Fabric Architecture

An architecture of the one-sided two-stage multirate switching fabric is shown in Fig. 4.7. Similarly as in the space-division and time-division switching fabrics, it consists of r switches in the first stage, and m switches in the second stage. Each first stage switch is connected with each of the second stage switches by means of v bidirectional links. In each of the first stage switches n links constitute the the switching fabric's terminals. The first stage switches have $n + mv$ terminals, and the second stage switches — rv terminals. The interstage links have the normalized bandwidth equal to 1. The bandwidth of the terminal is equal to β ($\beta \leqslant 1$). This switching fabric will be denoted by $T_{MR}(n, \beta, r, m, v)$. Connections may be set up between any terminal's pair. When these

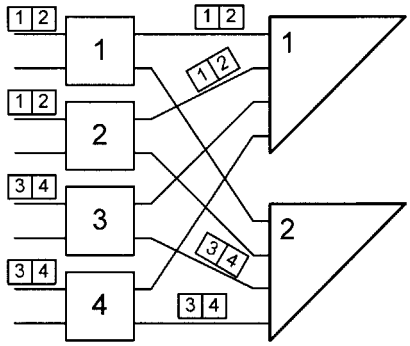


Figure 4.6a. The state of $T_{TD}(2,2,2,4,1,2)$ in which (I_1, I_3) is blocked

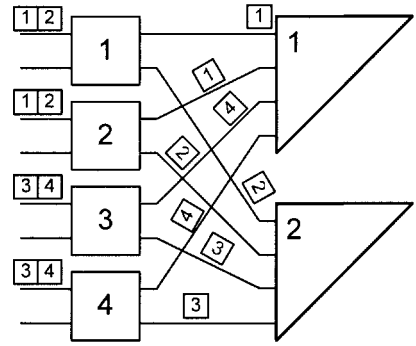


Figure 4.6b. The state of $T_{TD}(2,2,2,4,1,2)$ after rearrangements

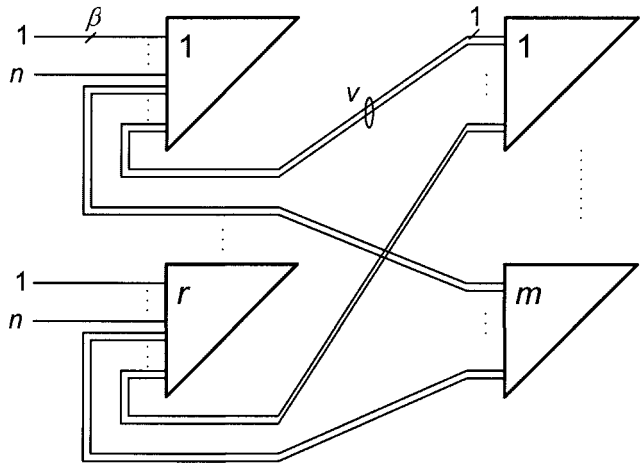


Figure 4.7. One-sided two-stage multirate switching fabric composed of triangular switches

terminals are in different first stage switches, the second stage switch is to be selected. Path searching algorithms are similar to those used in space-division switching fabrics. In multirate switching, similarly as in time-division switching, the routing strategy with functional division of second stage switches may be used. In this strategy, m_1 second stage switches are reserved for connections with weights less than or equal to 0.5 (i.e., $b \leqslant \omega \leqslant 0.5$) and m_2 second stage switches are reserved for connections of weights greater than 0.5 (i.e., $0.5 < \omega \leqslant B$).

4.2.3.2 Strict-sense Nonblocking Conditions

First the discrete bandwidth case will be considered. Only switching fabrics with $r > 3$ will be considered. Conditions for $r \leqslant 3$ may be derived by analogy to space-division and time-division switching fabrics.

THEOREM 4.12 *$T_{MR}(n, \beta, m, r, v)$ with $r > 3$ is nonblocking in the strict sense in the discrete bandwidth case for multirate connections of weight ω, $0 < b \leqslant \omega \leqslant B \leqslant \beta \leqslant 1$, if and only if:*

$$m \geqslant 2 \left\lfloor \frac{(n-1)\left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - B}{b} \right\rfloor}{v \left\lfloor \frac{1 - B + b}{b} \right\rfloor} \right\rfloor + 1. \tag{4.25}$$

Proof. Necessity. The following algorithm will be used to set up a new connection. If the new connection of weight ω arrives in the same first stage switch as the previous connection, the path searching will start from the link used by the previous connection. If the new connection arrives in another first stage switch, the path searching will start from the first link to the second stage switch next to the switch used by the previous connection.

Step 1 Set up $\lfloor (1 - B + b)/b \rfloor$ connections (I_i, I_j, b), $i \neq j$.

Step 2 Set up connection $(I_i, I_j, (B - b))$.

Step 3 Set up connection (I_i, I_j, b), disconnect $(I_i, I_j, (B - b))$, and set up $\lfloor (1 - B)/b \rfloor$ connections (I_i, I_j, b).

Step 4 Repeat connections and disconnections from steps 2 and 3 up until
$v \lfloor (1 - B + b)/b \rfloor \lfloor [(n-1) \lfloor \beta/b \rfloor + \lfloor (\beta - B)/b \rfloor]/[v \lfloor (1 - B + b)/b \rfloor] \rfloor$
connections (I_i, I_j, b) will be set up in I_i. These connections will make $\lfloor [(n-1) \lfloor \beta/b \rfloor + \lfloor (\beta - B)/b \rfloor]/[v \lfloor (1 - B + b)/b \rfloor] \rfloor$ second stage switches inaccessible by the connection (I_i, I_w, B).

Step 5 Set up $\lfloor (1 - B + b)/b \rfloor$ (I_w, I_z, b), $w \neq z$, $z \neq j$, $w \neq i$.

Step 6 Set up connection $(I_w, I_z, (B - b))$.

Step 7 Set up connection (I_w, I_z, b), disconnect $(I_w, I_z, (B - b))$, and set up $\lfloor (1 - B)/b \rfloor$ connections (I_w, I_z, b).

Step 8 Repeat connections and disconnections from steps 6 and 7 up until
$v \lfloor (1 - B + b)/b \rfloor \lfloor [(n-1) \lfloor \beta/b \rfloor + \lfloor (\beta - B)/b \rfloor]/[v \lfloor (1 - B + b)/b \rfloor] \rfloor$

connections (I_w, I_z, b) will be set up in I_w. These connections will make next $\lfloor[(n-1)\lfloor\beta/b\rfloor + \lfloor(\beta - B)/b\rfloor]/[v\lfloor(1-B+b)/b\rfloor]\rfloor$ second stage switches inaccessible by the connection (I_i, I_w, B).

In the second stage, $2\lfloor[(n-1)\lfloor\beta/b\rfloor + \lfloor(\beta - B)/b\rfloor]/[v\lfloor(1-B+b)/b\rfloor]\rfloor$ switches are inaccessible by the connection (I_i, I_w, B), and one more second stage switch is needed.

Sufficiency. Assume that the new connection is (I_i, I_j, ω). In switch I_i at most $\lfloor\beta/b\rfloor$ connections of weight b can be set up in each of $n-1$ terminals, and $\lfloor(\beta - B)/b\rfloor$ such connections may be set up in the remaining terminal. The total number of connections is $(n-1)\lfloor\beta/b\rfloor + \lfloor(\beta - B)/b\rfloor$. It should be noted, that one connection of weight b_k and b_k/b connections of weight b between the same switches are equivalent when the state of the switching fabric is considered (b_k/b is an integer in the discrete bandwidth case). The interstage link will be inaccessible by the new connection of weight ω if the sum of connections' weights already realized through this link is greater than $1-\omega$. Thus at least $\lfloor(1-\omega)/b\rfloor + 1 = \lfloor(1-\omega+b)/b\rfloor$ connections of weight b are to be set up through this link. These connections may occupy $[(n-1)\lfloor\beta/b\rfloor + \lfloor(\beta-\omega)/b\rfloor]/[v\lfloor(1-\omega+b)/b\rfloor]$ second stage switches in such way that these switches will be inaccessible by the new connection of weight ω from switch I_i. Similar configuration of connections may occupy next $[(n-1)\lfloor\beta/b\rfloor + \lfloor(\beta-\omega)/b\rfloor]/[v\lfloor(1-\omega+b)/b\rfloor]$ second stage switches, which will be inaccessible by the new connection of weight ω from switch I_j. In the worst case these sets of switches are disjoint and one more switch is needed for connection (I_i, I_j, ω). All together we should have at least

$$m \geqslant 2\left\lfloor \frac{(n-1)\left\lfloor\dfrac{\beta}{b}\right\rfloor + \left\lfloor\dfrac{\beta-\omega}{b}\right\rfloor}{v\left\lfloor\dfrac{1-\omega+b}{b}\right\rfloor}\right\rfloor + 1 \tag{4.26}$$

second stage switches. Equation (4.26) should be maximized through all possible values of ω, and this maximum is obtained for $\omega = B$. □

Nonblocking conditions of the $T_{MR}(n, \beta, m, r, v)$ in the continuous bandwidth case were considered in [75]. We will consider now only the switching fabrics with $r > 3$. Nonblocking conditions for $r \leqslant 3$ can be

derived by analogy. The functions defined below are used in this case:

$$P(i;j) = \lim_{\varepsilon \to 0} \left\lfloor \frac{i}{j+\varepsilon} \right\rfloor \tag{4.27}$$

$$= \begin{cases} \left\lfloor \frac{i}{j} \right\rfloor & \text{if } \frac{i}{j} \text{ is not an integer or } \left\lfloor \frac{i}{j} \right\rfloor = 0 \\ \left\lfloor \frac{i}{j} \right\rfloor - 1 & \text{if } \frac{i}{j} \text{ is an integer and} \frac{i}{j} > 0 \\ 0 & \text{if } j = 0 \end{cases},$$

$$R_1(i;j) = \lim_{\varepsilon \to 0} \left(i - (j+\varepsilon) \left\lfloor \frac{i}{j+\varepsilon} \right\rfloor \right) \tag{4.28}$$

$$= \begin{cases} i - j \cdot P(i;j) & \text{for } P(i;j) \neq 0 \\ & \text{and } i - j \cdot P(i;j) > b \\ 0 & \text{for } P(i;j) \neq 0 \\ & \text{and } i - j \cdot P(i;j) \leqslant b \\ i & \text{for } P(i;j) = 0 \end{cases}$$

$$R_2(i;j) = \begin{cases} \lim_{\varepsilon \to 0} \left\lfloor \frac{j+\varepsilon}{R_1(i;j)} \right\rfloor = \left\lfloor \frac{j}{R_1(i;j)} \right\rfloor & \text{for } R_1(i;j) \geqslant b \\ 0 & \text{for } R_1(i;j) < b \end{cases}, \tag{4.29}$$

$$R_3(i;j) = \begin{cases} \frac{i}{j} & \text{for } j \neq 0 \\ 0 & \text{for } j = 0 \end{cases}, \tag{4.30}$$

and

$$R_5(i) = \begin{cases} i & \text{for } i \geqslant b \\ 0 & \text{for } i < b \end{cases}. \tag{4.31}$$

THEOREM 4.13 *$T_{MR}(n, \beta, m, r, v)$ is nonblocking in the strict sense for the continuous bandwidth case, and $r > 3$ if*

$$m \geqslant 2 \left\lfloor \frac{S(B)}{v} \right\rfloor + 1, \tag{4.32}$$

where

$$S(B) = \begin{cases} (n-1)\left\lfloor \dfrac{\beta}{b} \right\rfloor, & \text{for } B \in (1-b,\ \beta]; \\ \dfrac{(n-1)\left\lfloor \dfrac{\beta}{b} \right\rfloor + \left\lfloor \dfrac{\beta - B}{b} \right\rfloor}{2}, & \text{for } B \in \left(1-2b,\ \frac{1}{2}\right] \text{ and } \frac{1}{4} < b < \frac{1}{2}; \\ (n-1)\,P(\beta; 1-B), & \text{for all other } B \text{ and } R_1(\beta; 1-B) < b; \\ (n-1)\,P(\beta; 1-B) + \lfloor R_3(n-1; a)\rfloor + P(\alpha(B); 1-B) & \text{for all other } B \text{ and } b \leqslant R_1(\beta; 1-B) < 2b; \\ (n-1)\,P(\beta; 1-B) + P[(n-1)\,R_1(\beta; 1-B) + R_5(\beta - B); 1-B], & \text{for all other } B \text{ and } R_1(\beta; 1-B) \geqslant 2b; \end{cases} \tag{4.33}$$

$$\alpha(B) = [n - 1 - a\lfloor R_3(n-1; a)\rfloor]\,R_1(\beta; 1-B) + R_5(\beta - B), \tag{4.34}$$

$$a = R_2(\beta;\ 1-B) + 1, \tag{4.35}$$

Proof. Sufficiency. Suppose we want to add a new connection (I_i, I_j, ω), $0 < b \leqslant \omega \leqslant B \leqslant \beta \leqslant 1$. Any interstage link from switch I_i will be inaccessible to the new connection of weight ω, if the sum of connection weights already set up through this link is greater than $1-\omega$. In the worst case this sum should be as small as possible, say $1-\omega+\varepsilon$, where ε is close to but greater than 0. However, when $1-\omega < b$, it is only possible, in the worst case, to set up a connection of weight b through this link. When $1-\omega \geqslant B$ it is not possible that one connection will occupy weight $1-\omega+\varepsilon$. In this case we have to set up at least two connections. When $1-\omega < 2b$, the total weight which makes the interstage link inaccessible to the connection of weight ω is equal to $2b$. Otherwise, it is always possible to set up one or more connections with the total weight $1-\omega+\varepsilon$. There are three cases:

1 $1-\omega < b$,

2 $b \leqslant 1-\omega < 2b$ and $1-\omega \geqslant B$,

3 $1-\omega \geqslant 2b$ or $b \leqslant 1-\omega < 2b$ and $1-\omega < B$.

Case 1, $1-\omega < b$. If this condition is fulfilled, the interstage link is inaccessible to the new connection if it carries one connection of weight b. In the worst case each connection with this weight from switch I_i may be set up through a separate interstage link. In one terminal there may be at most $\lfloor \beta/b \rfloor$ connections of weight b. If β/b is not an integer, there is some free bandwidth in this terminal, but it is less than b and it cannot be used by the next connection. There are $n-1$ such terminals. In terminal n there is a free bandwidth of weight $\beta-\omega$. Since $1-\omega < b$ and $\beta \leqslant 1$, then $\beta - \omega < b$. In this terminal we cannot set up a connection of weight b. In switch I_i there may be at most $(n-1)\lfloor \beta/b \rfloor$ connections of weight b, and they will occupy a set of $\lfloor (n-1)\lfloor \beta/b \rfloor / v \rfloor$ second stage switches in such way that they will be inaccessible to the connection of weight ω. Similarly, in switch I_j there may be at most $(n-1)\lfloor \beta/b \rfloor$ connections of weight b, and they will also occupy a set of $\lfloor (n-1)\lfloor \beta/b \rfloor / v \rfloor$ second stage switches. In the worst case these sets are disjoint and one more switch in the second stage is needed to set up connection (I_i, I_j, ω). So if $1-\omega < b$

$$m \geqslant 2 \left\lfloor \frac{S(\omega)}{v} \right\rfloor + 1 \tag{4.36}$$

second stage switches are needed, where

$$S(\omega) = (n-1) \left\lfloor \frac{\beta}{b} \right\rfloor . \tag{4.37}$$

Case 2, $b \leqslant 1-\omega < 2b$ and $1-\omega \geqslant B$. If these conditions are fulfilled, the interstage link is inaccessible to the new connection of weight ω, if it carries connections of the total weight greater than $1-\omega$. Since $1-\omega \geqslant B$, it is not possible to set up one connection of such weight. So, at least two connections have to be set up in one link. Because $1-\omega < 2b$, the interstage link will be inaccessible to the new connection of weight ω, if it carries two connections of weight b. Similarly as in case 1, we may have $(n-1)\lfloor \beta/b \rfloor$ connections of weight b in $n-1$ terminals of switch I_i, and $\lfloor (\beta-\omega)/b \rfloor$ connections of such a weight in the last terminal of this switch. These connections may occupy a set of at most $\lfloor [(n-1)\lfloor \beta/b \rfloor + \lfloor (\beta-\omega)/b \rfloor]/2v \rfloor$ second stage switches, and these switches are inaccessible to the new connection. Similarly in switch I_j there may be at most $(n-1)\lfloor \beta/b \rfloor + \lfloor (\beta-\omega)/b \rfloor$ connections of weight b, and these connections will occupy next $\lfloor [(n-1)\lfloor \beta/b \rfloor + \lfloor (\beta-\omega)/b \rfloor]/2v \rfloor$ second stage switches. These switches will also be inaccessible to the new connection. In the worst case these sets are disjoint and one more switch in the second stage is needed to set up connection (I_i, I_j, ω). So,

if $b \leqslant 1-\omega < 2b$ and $1-\omega \geqslant B$, we can write that

$$m \geqslant 2\left\lfloor \frac{S(\omega)}{v} \right\rfloor + 1 \tag{4.38}$$

middle stage switches are needed, where

$$S(\omega) = \frac{(n-1)\left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta-\omega}{b} \right\rfloor}{2}. \tag{4.39}$$

Case 3, $1-\omega \geqslant 2b$ or $b \leqslant 1-\omega < 2b$ and $1-\omega < B$. If these conditions are fulfilled, the interstage link is inaccessible to the new connection of weight ω, if it carries connections of the total weight greater than $1-\omega$. If $1-\omega < B$, only one connection of such weight may be set up. In the other case, at least two connections of total weight greater than $1-\omega$ must be set up in the interstage link. In each terminal $\lfloor \beta/(1-\omega) \rfloor$ connections of weight greater than $1-\omega$ may be set up if $\beta/(1-\omega)$ is not an integer, or $\lfloor \beta/(1-\omega) \rfloor - 1$ such connections if $\beta/(1-\omega)$ is an integer greater than 0. The maximum number of connections of weight $1-\omega+\varepsilon$, $\varepsilon \to 0$, that can be set up in one terminal is represented by function $P(\beta; 1-\omega)$. So, at most $(n-1)P(\beta; 1-\omega)$ connections of such weight may be set up in switch I_i. There is still free bandwidth of weight $\beta-\omega$ in terminal n, but since $\beta-\omega \leqslant 1-\omega$, it cannot be used by a connection of weight greater than $1-\omega$.

In each of the $n-1$ terminals there is free bandwidth, and its weight is represented by function $R_1(\beta; 1-\omega) = \beta - (1-\omega)P(\beta; 1-\omega)$. When $R_1(\beta; 1-\omega) < b$, this bandwidth cannot be used by the next connection. This means that in switch I_i we may have at most $(n-1)P(\beta; 1-\omega)$ connections of weight greater than $1-\omega$, and these connections may occupy $\lfloor (n-1)P(\beta; 1-\omega)/v \rfloor$ second stage switches such that they will be inaccessible to the connection of weight ω in switch I_i. This means that, in switch I_i we have $S(\omega) = (n-1)P(\beta; 1-\omega)$, connections of weight greater than $1-\omega$, and these connections occupy $\lfloor S(\omega)/v \rfloor$ second stage switches.

When $b \leqslant R_1(\beta; 1-\omega) < 2b$, the remaining bandwidth in each link can be used by the next connection. Several such connections of weight $R_1(\beta; 1-\omega)$ in one interstage link may lead to the state, in which this link will be inaccessible to the new connection (the minimum number of these connections is denoted by a). The next interstage link will be inaccessible to the new connection, if it carries $a = R_2(\beta; 1-\omega) + 1$ connections of weight $R_1(\beta; 1-\omega)$. This means, that the next $\lfloor R_3(n-1; a) \rfloor$ interstage links from switch I_i will be inaccessible to the new connection. In switch I_i, we have now $n-1-a\lfloor R_3(n-1; a) \rfloor$ terminals with available bandwidth

of weight $R_1(\beta; 1-\omega)$, and one link with available bandwidth of weight $R_5(\beta-\omega)$ (Function $R_5(i)$ indicates whether the bandwidth quantity i is less than b or not, so it determines whether this bandwidth can be used by a connection or not). Connections of such weights may occupy a bandwidth of weight $\alpha(\omega) = [n-1-a\lfloor R_3(n-1;a)\rfloor] R_1(\beta;1-\omega) + R_5(\beta-\omega)$ in an interstage link. If $\alpha(\omega)$ is greater than $1-\omega$, then this interstage link will also be inaccessible to the new connection in switch I_i. Whether this interstage link is accessible or not, can be calculated by function $P(\alpha(\omega); 1-\omega)$. Therefore, we may have $S(\omega) = (n-1)P(\beta; 1-\omega) + \lfloor R_3(n-1;a)\rfloor + P(\alpha(\omega); 1-\omega)$ interstage links in switch I_i, which are inaccessible to the new connection of weight ω. These links will fully occupy $\lfloor S(\omega)/v\rfloor$ second stage switches.

For $R_1(\beta; 1-\omega) \geqslant 2b$ this remaining bandwidth may be divided among more than one connections. However, these connections may occupy no more than $P([(n-1)R_1(\beta;1-\omega)+R_5(1-\omega)]; 1-\omega)$ interstage links. This means that $S(\omega) = (n-1)P(\beta;1-\omega) + P([(n-1)R_1(\beta;1-\omega)+R_5(1-\omega)]; 1-\omega)$ interstage links may be inaccessible to the new connection of weight ω and they will fully occupy $\lfloor S(\omega)/v\rfloor$ second stage switches.

In switch I_j, similarly as for switch I_i, we may have $S(\omega)$ interstage links inaccessible to the new connection, which will occupy $\lfloor S(\omega)/v\rfloor$ second stage switches, where

$$S(\omega) = \begin{cases} (n-1)\,P(\beta;1-\omega), & \text{for } R_1(\beta;1-\omega) < b; \\ (n-1)\,P(\beta;1-\omega) + \lfloor R_3(n-1;a)\rfloor + \\ \quad + P(\alpha(\omega);1-\omega), & \text{for } b \leqslant R_1(\beta;1-\omega) < 2b; \\ (n-1)\,P(\beta;1-\omega) + P[(n-1)\,R_1(\beta;1-\omega) + \\ \quad + R_5(\beta-\omega);1-\omega], & \text{for } R_1(\beta;1-\omega) \geqslant 2b; \end{cases} \tag{4.40}$$

and one more second stage switch is needed for connection (I_i, I_j, ω).

Considering equations (4.36), (4.37), (4.38), (4.39) and (4.40) of all three cases and taking into account that these values must be maximized through all ω we obtain following condition:

$$m \geqslant 2 \max_{b \leqslant \omega \leqslant B} \left\lfloor \frac{S(\omega)}{v} \right\rfloor + 1, \tag{4.41}$$

$$S(\omega) = \begin{cases} (n-1)\left\lfloor \frac{\beta}{b} \right\rfloor, & \text{for case 1;} \\ \frac{(n-1)\left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta-\omega}{b} \right\rfloor}{2}, & \text{for case 2;} \\ (n-1)\,P(\beta; 1-\omega), & \text{for case 3 and } R_1(\beta; 1-\omega) < b; \\ (n-1)\,P(\beta; 1-\omega) + \lfloor R_3(n-1; a) \rfloor + P(\alpha(\omega); 1-\omega), & \text{for case 3 and } b \leqslant R_1(\beta; 1-\omega) < 2b; \\ (n-1)\,P(\beta; 1-\omega) + P[(n-1)\,R_1(\beta; 1-\omega) + R_5(\beta-\omega); 1-\omega], & \text{for case 3 and } R_1(\beta; 1-\omega) \geqslant 2b; \end{cases} \tag{4.42}$$

and the maximum value is reached for $\omega = B$.

For cases 1 and 2 it is possible to show that the given conditions are not only sufficient but also necessary. Necessary conditions for these cases will be proved by showing a set of events leading to the occupancy of the number of the second stage switches given by inequality (4.32). The following path searching algorithm will be used. If the new connection will appear in the same first stage switch as the last connection set up, the path searching will start from the link through which the last connection was set up. When the new connection will appear in the other first stage switch, the path searching will start from the link to the second stage switch next to the last occupied. Let us assume that the new connection is (I_i, I_w, B).

Case 1, $B \in (1-b, 1]$.

Step 1 Set up connection (I_i, I_j, b), $i \neq j$.

Step 2 Set up connection $(I_i, I_j, 1-b)$, set up connection (I_i, I_j, b), disconnect connection $(I_i, I_j, 1-b)$.

Step 3 Repeat Step 2 until $v\lfloor(n-1)\lfloor\beta/b\rfloor/v\rfloor$ connections (I_i, I_j, b) are set up. These connections will occupy $\lfloor(n-1)\lfloor\beta/b\rfloor/v\rfloor$ second stage switches.

Step 4 Repeat Steps 1-3 for connections between switches I_w and I_z, $w \neq z$, $w \neq i$, $z \neq j$. These connections will occupy the next $\lfloor(n-1)\lfloor\beta/b\rfloor/v\rfloor$ second stage switches.

We have $2\lfloor(n-1)\lfloor\beta/b\rfloor/v\rfloor$ second stage switches occupied and these switches are inaccessible to connection (I_i, I_w, B). This connection will occupy the next middle stage switch.

Case 2, $B \in (1-2b, 1/2]$ and $1/4 < b < 1/2$.

Step 1 Set up two connection (I_i, I_j, b), $i \neq j$.

Step 2 Set up connection $(I_i, I_j, 1-2b)$, set up connection (I_i, I_j, b), disconnect connection $(I_i, I_j, 1-2b)$ and set up the next connection (I_i, I_j, b).

Step 3 Repeat Step 2 until $2v\lfloor[(n-1)\lfloor\beta/b\rfloor + \lfloor(\beta-B)/b\rfloor/2v\rfloor$ connections (I_i, I_j, b) are set up. These connections will occupy $\lfloor[(n-1)\lfloor\beta/b\rfloor + \lfloor(\beta-B)/b\rfloor/2v\rfloor$ second stage switches.

Step 4 Repeat Steps 1-3 for connections between switches I_w and I_z, $w \neq z$, $w \neq i$, $z \neq j$. These connections will occupy the next $\lfloor[(n-1)\lfloor\beta/b\rfloor + \lfloor(\beta-B)/b\rfloor/2v\rfloor$ second stage switches.

We have $2\lfloor[(n-1)\lfloor\beta/b\rfloor + \lfloor(\beta-B)/b\rfloor]/2v\rfloor$ second stage switches occupied and inaccessible to connection (I_i, I_w, B). This connection will occupy the next middle stage switch.

In case 3 the necessity can be proved only when $R_1(\beta; 1-\omega) < b$, since this bandwidth cannot be used by the next connections in terminals. The set of events will be very similar to those in cases 1 and 2. For $b \leqslant R_1(\beta; 1-\omega) < 2b$ and $R_1(\beta; 1-\omega) \geqslant 2b$ the conditions given in this theorem constitutes only the upper bounds. □

4.2.3.3 Wide-sense Nonblocking Conditions

Wide-sense nonblocking conditions may be considered when a path searching algorithm based on functional division of the second stage switches is used.

THEOREM 4.14 *$T_{MR}(n, \beta, m, r, v)$ with $r > 3$ is nonblocking in the wide sense for discrete bandwidth case, and under the routing strategy with functional division of second stage switches if and only if: $m \geqslant m_1 + m_2$, where*

$$m_1 = 2\left\lfloor \frac{(n-1)\left\lfloor\dfrac{\beta}{b}\right\rfloor + \left\lfloor\dfrac{\beta-0.5}{b}\right\rfloor}{v\left\lfloor\dfrac{0.5+b}{b}\right\rfloor}\right\rfloor + 1 \tag{4.43}$$

denotes the number of second stage switches which serves connections of weights $\omega \leqslant 0.5$, *and*

$$m_2 = 2\left\lfloor \frac{n-1}{v} \right\rfloor + 1 \tag{4.44}$$

denotes the number of second stage switches which serves connections of weights $\omega > 0.5$.

Proof. Proof of this Theorem is similar to the proof of Theorem 4.10. □

THEOREM 4.15 $T_{MR}(n, \beta, m, r, v)$ *with* $r > 3$ *is nonblocking in the wide sense for continuous bandwidth case, and under the algorithm with functional division of second stage switches if:* $m \geqslant m_1 + m_2$, *where*

$$m_1 \geqslant 2\left\lfloor \frac{S(0.5)}{v} \right\rfloor + 1, \tag{4.45}$$

$$S(0.5) = \begin{cases} \dfrac{(n-1)\left\lfloor \dfrac{\beta}{b} \right\rfloor + \left\lfloor \dfrac{\beta - 0.5}{b} \right\rfloor}{2}, & \text{for } b > 0.25; \\ (n-1)\,P(\beta; 0.5), & \text{for } b < 0.25 \text{ and } R_1(\beta; 0.5) < b; \\ (n-1)\,P(\beta; 0.5) + \\ \quad + \lfloor R_3(n-1; a) \rfloor + \\ \quad + P(\alpha(0.5); 0.5), & \text{for } b < 0.25 \text{ and } b \leqslant R_1(\beta; 0.5) < 2b; \\ (n-1)\,P(\beta; 1-0.5) + \\ \quad + P[(n-1)\,R_1(\beta; 0.5) + \\ \quad + R_5(\beta - 0.5); 0.5], & \text{for } b < 0.25 \text{ and } R_1(\beta; 0.5) \geqslant 2b; \end{cases} \tag{4.46}$$

denotes the number of second stage switches which serves connections of weights $\omega \leqslant 0.5$, *and*

$$m_2 = 2\left\lfloor \frac{n-1}{v} \right\rfloor + 1 \tag{4.47}$$

denotes the number of second stage switches which serves connections of weights $\omega > 0.5$.

Proof. Proof of this Theorem is also similar to the proof of Theorem 4.10. □

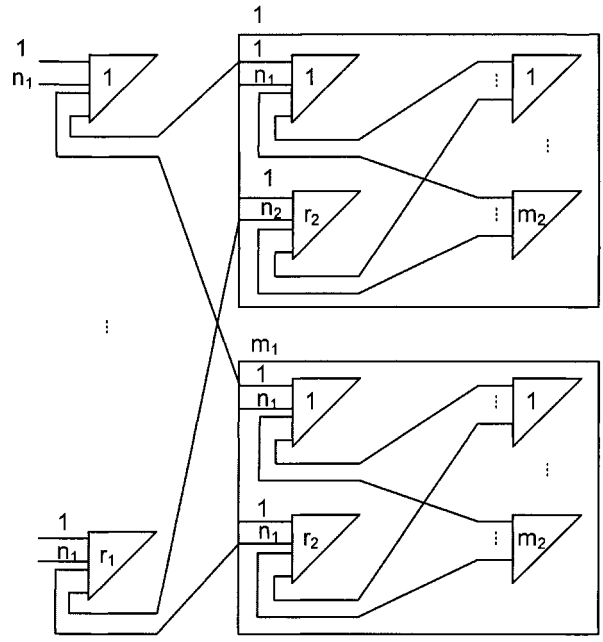


Figure 4.8. The one-sided three-stage switching fabric - structure 1

4.3 One-sided Switching Fabrics With More Than Two Stages

Switching networks of greater number of stages may be obtained by replacing second stage switches in the nonblocking network with two-stage nonblocking networks of the capacity equal to the second stage switches of the main network. In this way one-sided switching networks with three, four, and more stages can be built. The structure of such constructed one-sided three-stage switching fabric, called structure 1, is shown in Fig. 4.8. In this switching fabric m_1 second stage switches of $T_{SD}(n_1, m_1, r_1, 1)$ are replaced with $T_{SD}(n_2, m_2, r_2, 1)$. The capacity of this switching fabric is $N = r_1 n_1$, $r_1 = r_2 n_2$. When $r_1 > 3$ and $r_2 > 3$, this switching fabric is strict-sense nonblocking if and only if $m_1 \geqslant 2n_1 - 1$, and $m_2 \geqslant 2n_2 - 1$.

Other structures of one-sided three-stage switching fabric, called structure 2 and structure 3, are shown in Fig. 4.9 and Fig. 4.10, respectively. Structure 2 is obtained by replacing the first stage switches of $T_{SD}(n_1, m_1, r_1, 1)$ with $T_{SD}(n_2, m_2, r_2, 1)$. In structure 3 another approach is used in connecting the second and the third stage switches between themselves. These three structures are isomorphic, i.e., one

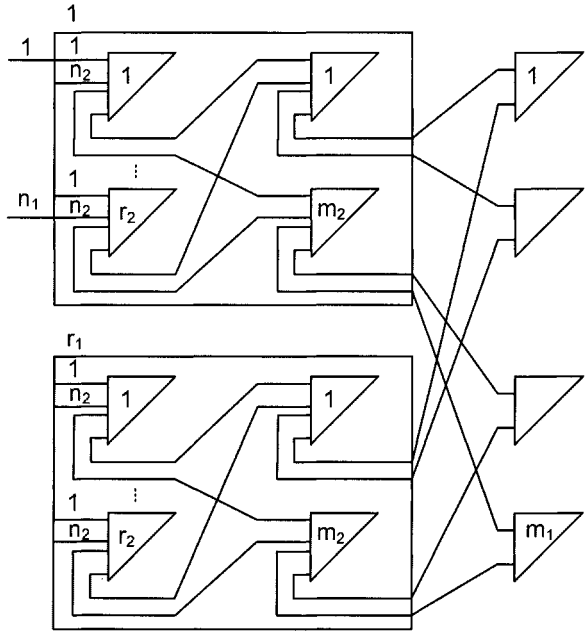


Figure 4.9. The one-sided three-stage switching fabric - structure 2

can be obtained from the other by renumbering the switches in stages. Respective theorems and functions which renumber the switches were given by Jajszczyk [64].

Structures considered above refers to space-division switching fabrics, but they are of course true also in time-division switching and multirate switching.

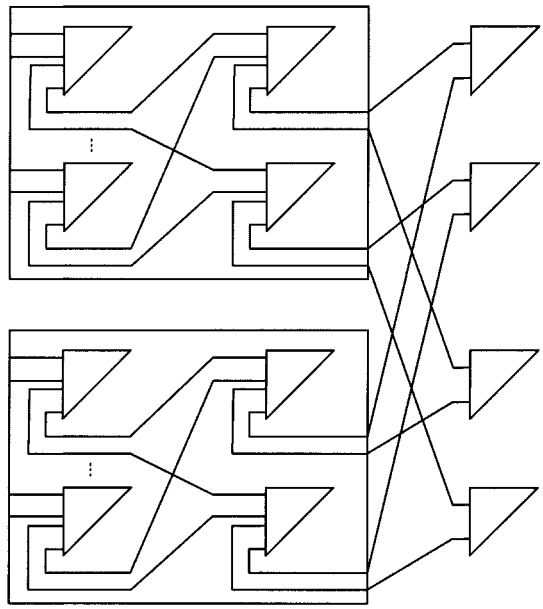


Figure 4.10. The one-sided three-stage switching fabric - structure 3

Chapter 5

THREE-STAGE SWITCHING FABRICS

5.1 Space-division Switching

5.1.1 Switching Fabric Architecture

The most popular architecture of two-sided three-stage switching fabric is the Clos switching fabric proposed by C. Clos [23]. The general architecture of the space-division switching fabric is shown in Fig. 5.1. It consists of r_1 switches in the first stage (called the first stage or input switches), r_2 switches in the third stage (called the third stage or output switches), and m switches in the second stage (called also the central or the middle stage switches). Each input switch has n_1 inputs and mv outputs. Switches in the third stage are of capacity $mv \times n_2$, and each of the second stage switches has the capacity of $vr_1 \times vr_2$. This switching fabric is fully connected, i.e., each switch of the previous stage is connected with each switch of the next stage. Switches in the first, the second and the third stages are numbered from 1 to r_1, from 1 to m, and from 1 to r_2, respectively. The capacity of this switching fabric is $N_1 \times N_2$, where $N_1 = n_1 r_1$, $N_2 = n_2 r_2$. In the Clos switching fabric switches in adjacent stages are connected by means of one link ($v = 1$). A v-dilated three-stage switching fabric will be denoted by $C_{SD}(n_1, r_1, m, n_2, r_2, v)$. This switching fabric is also referred to as the asymmetrical three-stage Clos switching fabric. When $n_1 = n_2 = n$ and $r_1 = r_2 = r$ the switching fabric is denoted by $C_{SD}(n, r, m, v)$ and is called symmetrical.

Switching fabrics with greater number of stages can be constructed using three-stage switching fabric as a basic construction. Clos proposed construction of the five-stage switching fabric by replacing each middle stage switch of the three-stage switching fabric by another three-stage

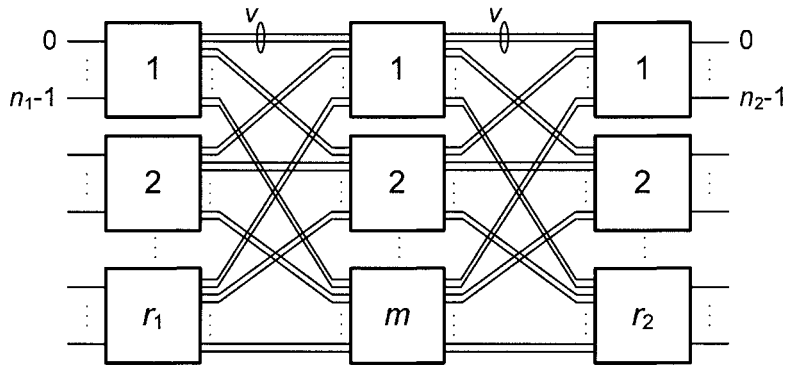


Figure 5.1. A two-sided space-division three-stage switching fabric

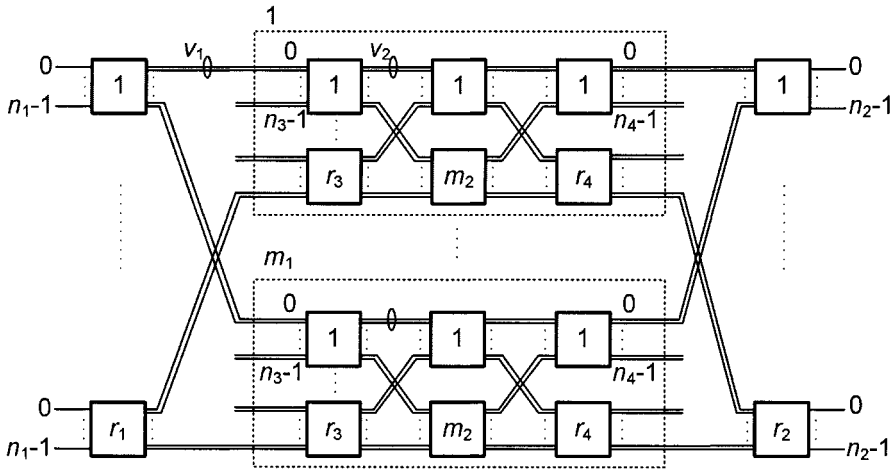


Figure 5.2. The two-sided space-division five-stage switching fabric

switching fabric. The example of five-stage switching fabric is shown in Fig. 5.2. This approach can be further used to construct switching fabrics with an odd number of stages. It should be noted that when switching fabrics on each hierarchical level has certain combinatorial properties, than the whole switching fabric also has these properties. For instance, when switching fabrics in all hierarchical levels are strict-sense nonblocking, than the whole switching fabric is also nonblocking. Therefore, in this chapter only three-stage switching fabrics will be considered.

5.1.2 Path Searching Algorithms

When a new call arrives at the switching fabric, a connection is to be set up between an input and an output terminals. Let $\langle x, y \rangle$ be a new connection from input terminal x of the first stage switch I_i to output terminal y of the third stage switch O_j, $0 \leqslant x \leqslant N_1 - 1$, $0 \leqslant y \leqslant N_2 - 1$, $1 \leqslant i \leqslant r_1$, and $1 \leqslant j \leqslant r_2$. The connecting path consists of input terminal x, switch I_i, inter-stage link from switch I_i to the center stage switch M_k, switch M_k, inter-stage link from switch M_k to switch O_j, switch O_j, and output terminal y. Since switches are nonblocking, often it is sufficient to say that the connection is to be set up between switches I_i and O_j. Such connection will be denoted by (I_i, O_j).

To set up the new connection the center stage switch M_k with free links to switches I_i and O_j is to be found. This switch M_k is called free or available for the new connection. Several path searching algorithms were considered so far for finding an available center stage switch. Below, a general description of these algorithms will be given, while the Packing algorithm, also called the Beneš algorithm, will be described in more detail.

ALGORITHM 5.1 *Random (RAN)*

Check center stage switches randomly and set up the connection through the first available switch.

ALGORITHM 5.2 *Sequential (SEQ)*

Check center stage switches sequentially starting form the center stage switch M_k, $1 \leqslant k \leqslant m$ and choose the first available switch.

ALGORITHM 5.3 *Minimum index (MINIX)*

This algorithm is the same as sequential, but $k = 1$.

ALGORITHM 5.4 *Quasi-random (Q-RAN)*

This algorithm is the same as sequential, but $k = l+1$, where M_l denotes the switch used to route the last request ($k = 1$ for $l = m$). The Q-RAN is also called *cyclic dynamic* (CD) or *round-robin* (RR).

ALGORITHM 5.5 *Cyclic static (CS)*

This algorithm is the same as Q-RAN, but $k = l$, i.e., we start to check center stage switches from the the switch used to route the last request.

ALGORITHM 5.6 *Save the unused (STU)*

Do not route a new connection through any empty center stage switch unless there is no choice.

ALGORITHM 5.7 *Packing (PACK)*

Route a new connection through the busiest but available center stage switch.

Step 1 Sort center stage switches in vector **L** according to the number of connections set up through these switches

Step 2 Take the first switch from **L**. Denote it by M_k

Step 3 Check if links from I_i to M_k and from M_k to O_j are accessible for connection (I_i, O_j).

Step 4 If these links are accessible then set up the connection through switch M_k and algorithm ends. Otherwise go to step 5.

Step 5 Repeat steps 3 and 4 for next switches in **L** until all center stage switches are checked.

Step 6 If all center stage switches are checked and connection is not set up, report a blocking state.

In all these algorithms when all center stage switches were checked and the switch for the new connection was not found, the connection is blocked. In recursively constructed multi-stage switching fabrics these algorithms are used for each three-stage switching fabric on respective hierarchical level.

Different routing strategies may be used in three-stage switching fabrics to set up multicast connections. These strategies depend on the capabilities of switches used for constructing the three-stage switching fabric. For instance, to reduce the cost of switches in some stages switches with no fan-out capability may be used. Some possible strategies are:

- *Any-split strategy*: connecting path to output terminals may be spread in any number of switches and in any stage. In this strategy switches in all sections have fan-out capability.
- *No-split restriction strategy*: connecting path to output terminals of the same third stage switch is spread only in this switch, i.e., it must use only one connecting path to connect output terminals on the same third stage switch. In this strategy switches in all sections have fan-out capability.

- *1-split restriction strategy*: Connecting path cannot be spread to more than one output of the first stage switches. In this strategy switches in the first stage may have no fan-out capability.

- *2-split restriction strategy*: Connecting path cannot be spread to more than one output of the second stage switches. In this strategy switches in the second stage may have no fan-out capability.

- *3-split restriction strategy*: Connecting path cannot be spread to more than one outputs of the third stage switches. In this strategy switches in the third stage may have no fan-out capability.

- *p-limited no-split restriction strategy*: connecting path to output terminals of the same third stage switch is spread only in this switch and the connection path can use at most p middle stage switches.

An example of a multicast connection set up using these strategies is shown in Figs. 5.3a, 5.3b, 5.3c, 5.3d, and 5.3e. In the case of any-split strategy (Fig. 5.3a) connection is spread in switches of all stages. It can be seen in switch 1 of the third stage, two inputs are used by this connection and one of this input is spread to two outputs of this switch. When no-split strategy is used only one input of each third stage switches is used by the connection and it is spread to all requested outputs of these switches. The connecting path set up using 1-split strategy is shown in Fig. 5.3b. As can be seen connection is not spread in switch 1 of stage 1, i.e., in this switch only one input and one output is used by the connection. Connecting path when 2-split and 3-split strategies were used are shown in Figs. 5.3c and 5.3d, respectively. Finally, p-limited no-split strategy when $p = 2$ is shown in Fig. 5.3e.

Additionally, when the switching fabric is v-dilated, two approaches may be used to set up multicast connections, depending on the number of links used by a multicast connection in a set of v links [108]:

- *Duplication routing*: two or more inputs of the same switch may be used by the same multicast connection.

- *Non-duplication routing*: each multicast connection uses at most one input of each switch.

Four examples of duplication routing are shown in Figs. 5.4a, 5.4b, 5.4c, and 5.4d. In the first example (Fig. 5.4a) the multicast connection uses two inputs of switch 2 in the second and in the third stage. In the example of Fig. 5.4b, the multicast connection uses two inputs of switch 3 in the middle stage. The third example (Fig. 5.4c) shows multicast connection which uses two inputs of switch 3 in the third stage. Finally,

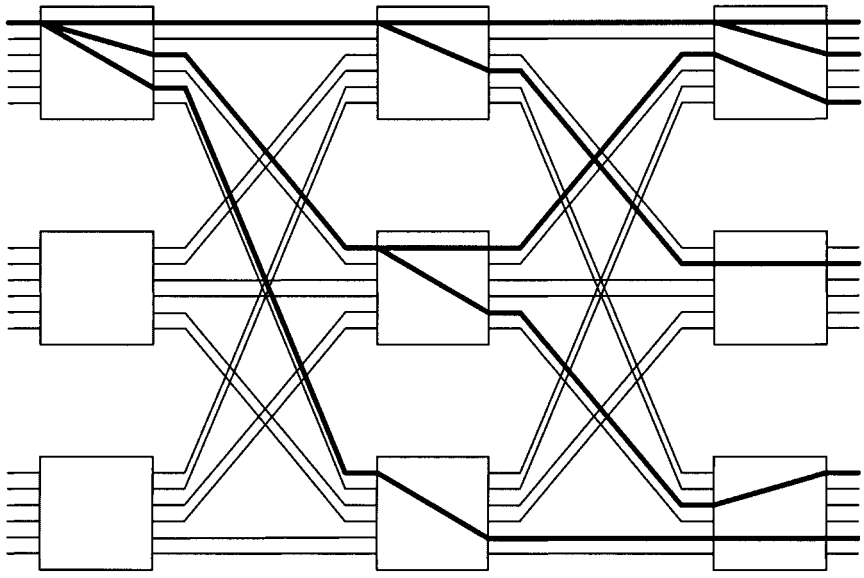


Figure 5.3a. Any-split strategy

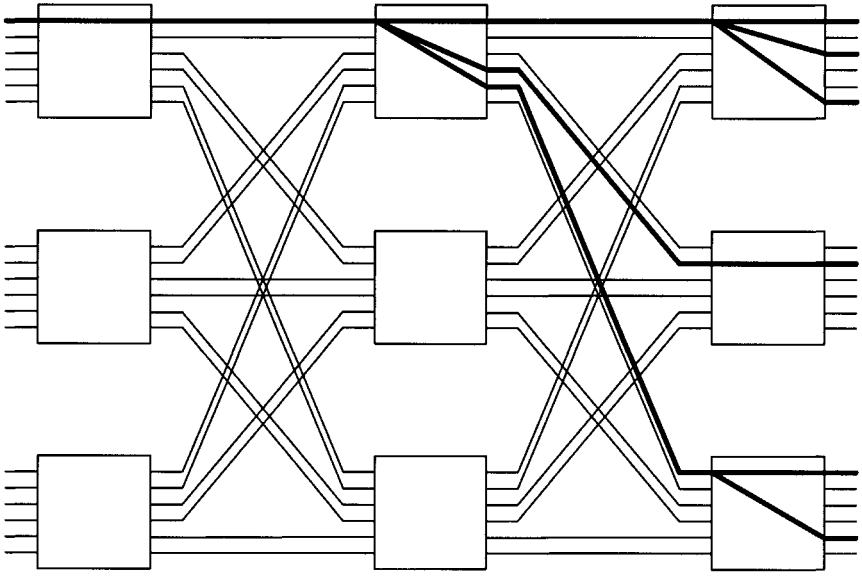


Figure 5.3b. 1-split restriction strategy

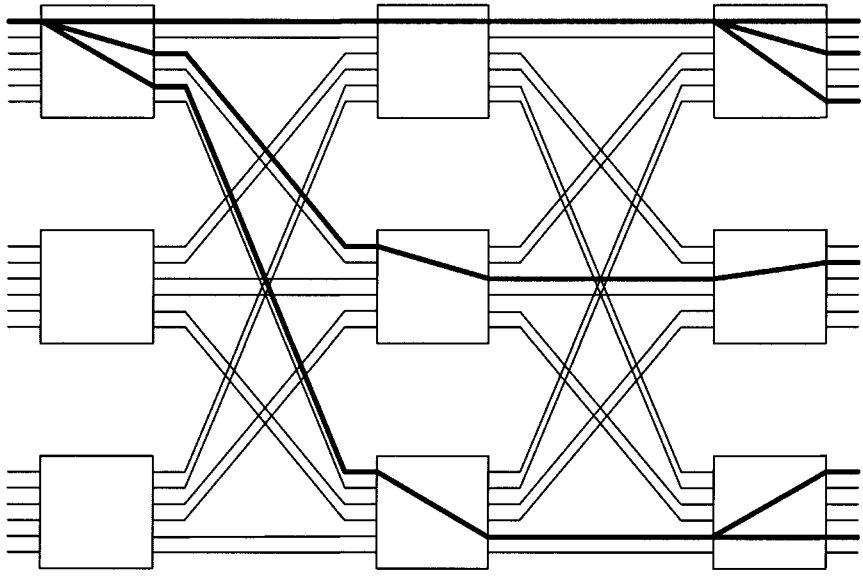


Figure 5.3c. 2-split restriction strategy

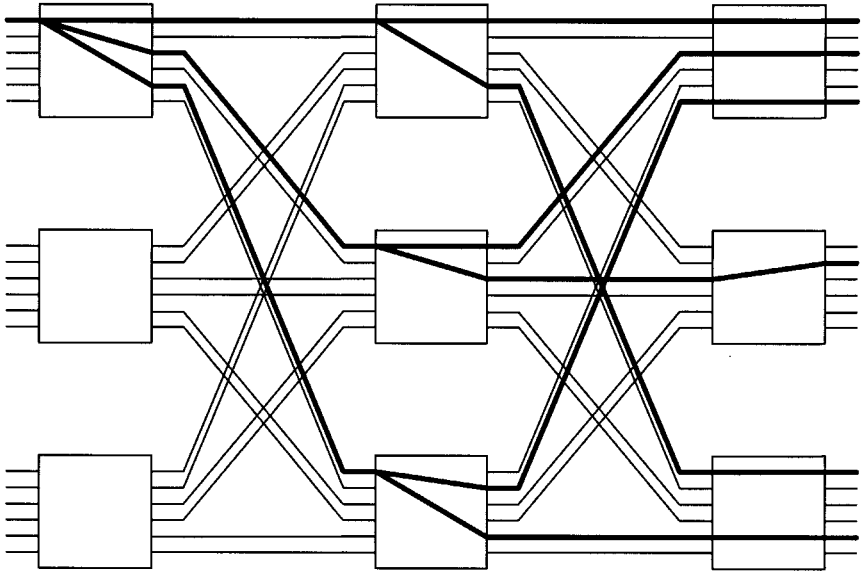


Figure 5.3d. 3-split restriction strategy

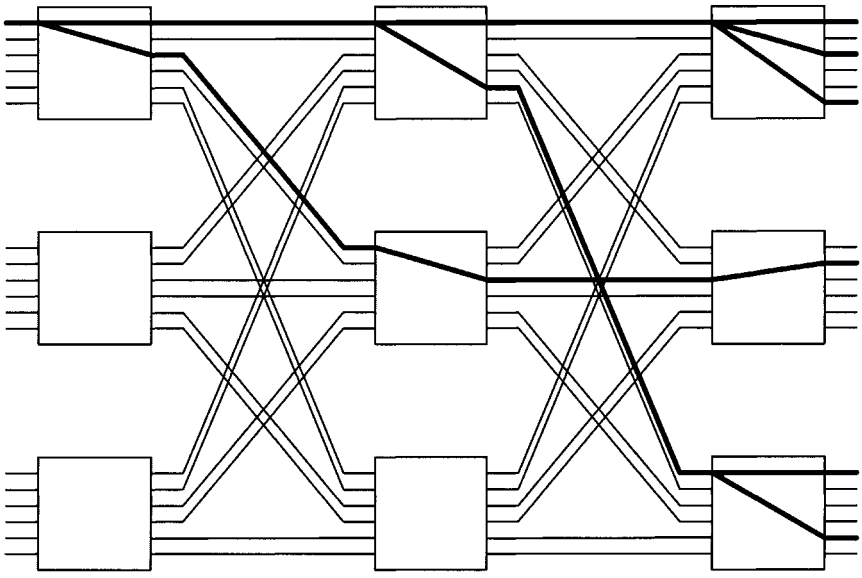


Figure 5.3e. 2-limited no-split restriction strategy

the multicast connection presented in Fig. 5.4d also uses duplication routing but no-split strategy is preserved, i.e., it is only one connecting path to output terminals at the same third stage switch. The same multicast connections set up using non-duplication routing are shown in Figs. 5.5a, 5.5b, 5.5c, and 5.5d, respectively. The connection shown in Fig. 5.4d is set up using 2-split restriction strategy with duplication routing since two links are used between switches 1 of stages 1 and 2.

5.1.3 Strict-sense Nonblocking Conditions

5.1.3.1 Unicast Connections

The sufficient conditions under which $C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ is strictly nonblocking were given by C. Clos [23]. He did not prove the necessity, which was later given by other authors [113]. The conditions are given in the following theorem:

THEOREM 5.1 *$C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ is nonblocking in the strict sense if and only if*

$$m \geqslant \min\{n_1 + n_2 - 1; n_1 r_1; n_2 r_2\} \tag{5.1}$$

Proof. Sufficiency can be proved by showing the worst state in the switching fabric. In the first stage switch I_i there may be at most $n_1 - 1$ connections to outputs in $r_2 - 1$ switches of the third stage (others than

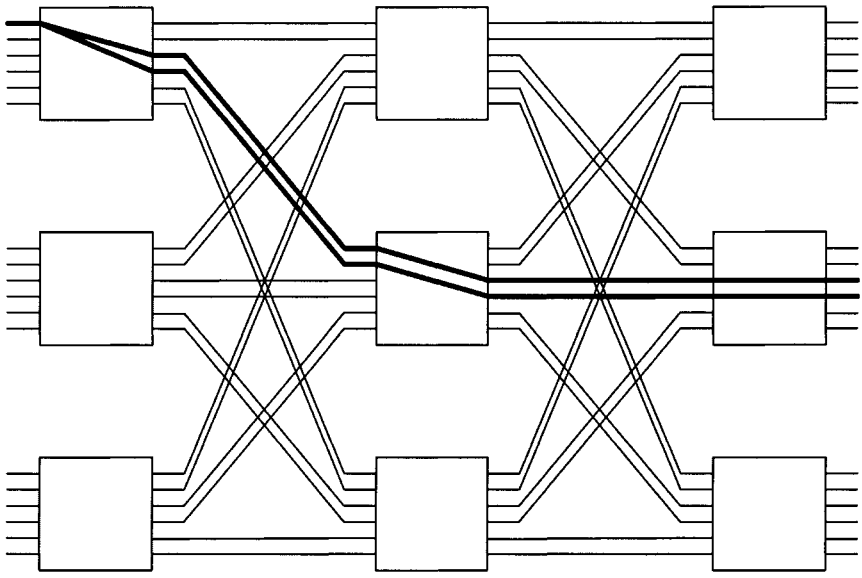


Figure 5.4a. Duplication routing - example 1

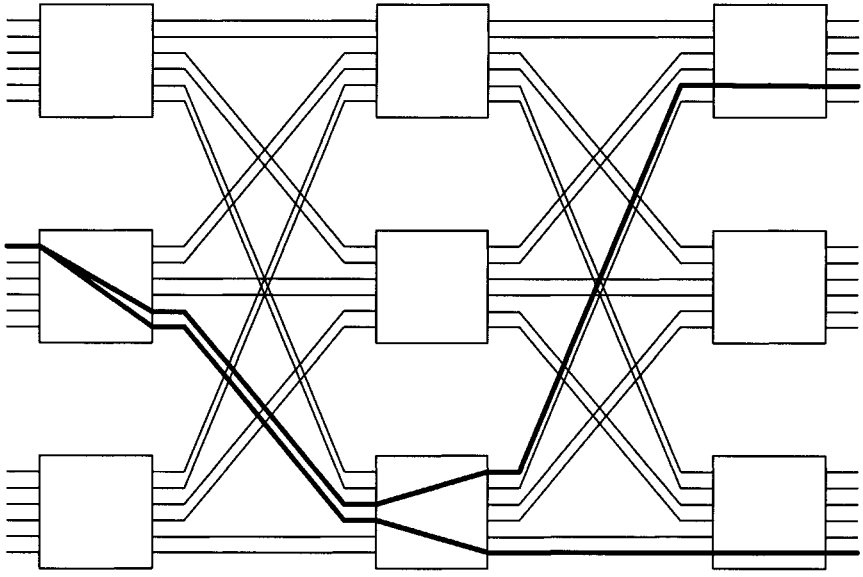


Figure 5.4b. Duplication routing - example 2

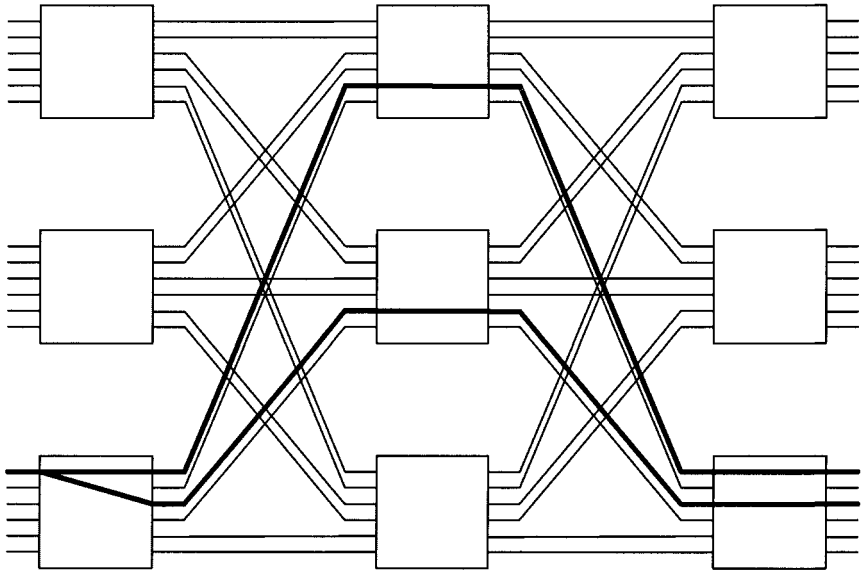


Figure 5.4c. Duplication routing - example 3

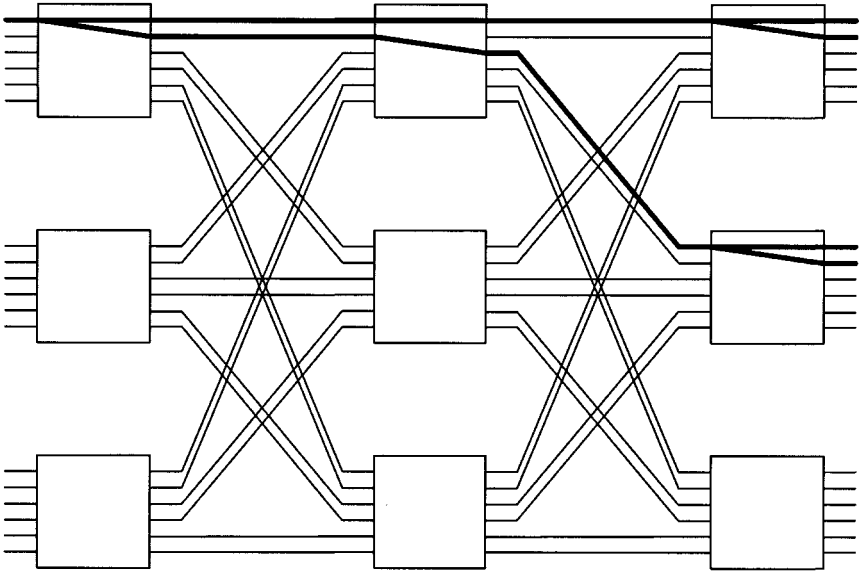


Figure 5.4d. Duplication routing - example 4

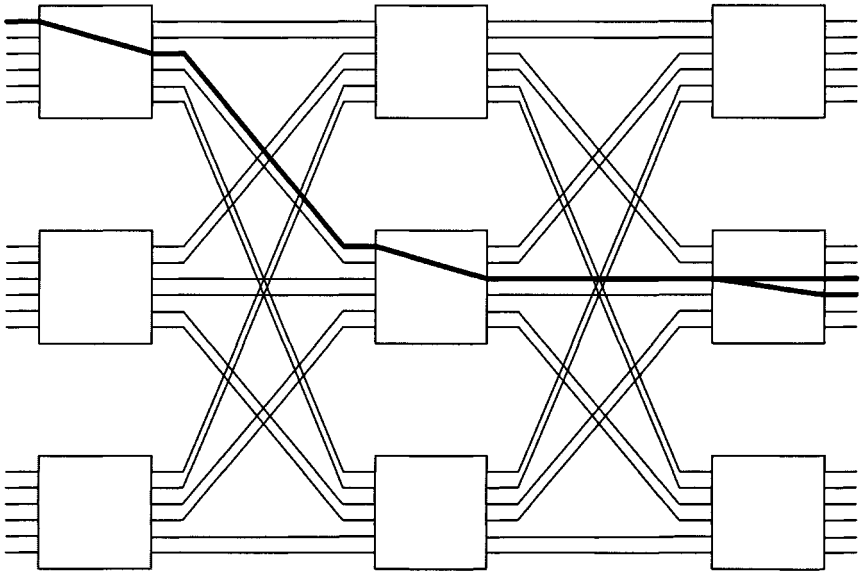


Figure 5.5a. Non-duplication routing - example 1

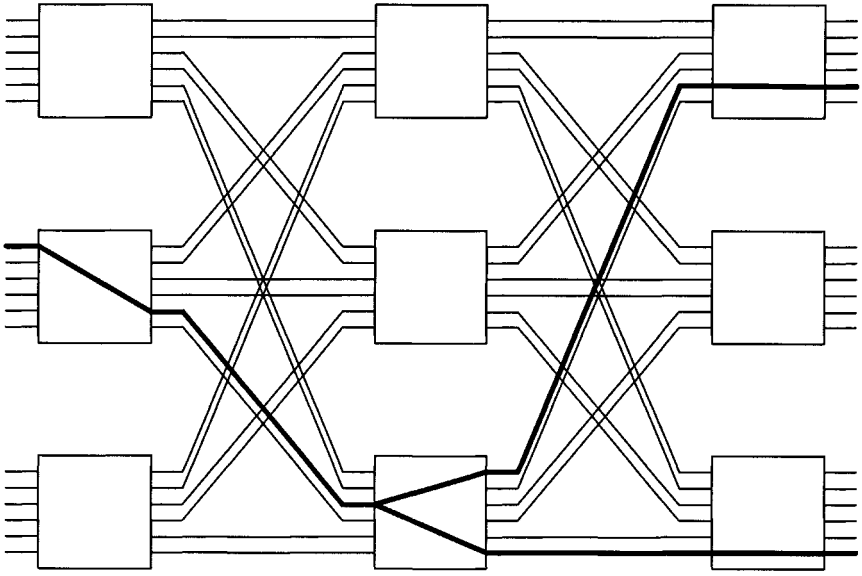


Figure 5.5b. Non-duplication routing - example 2

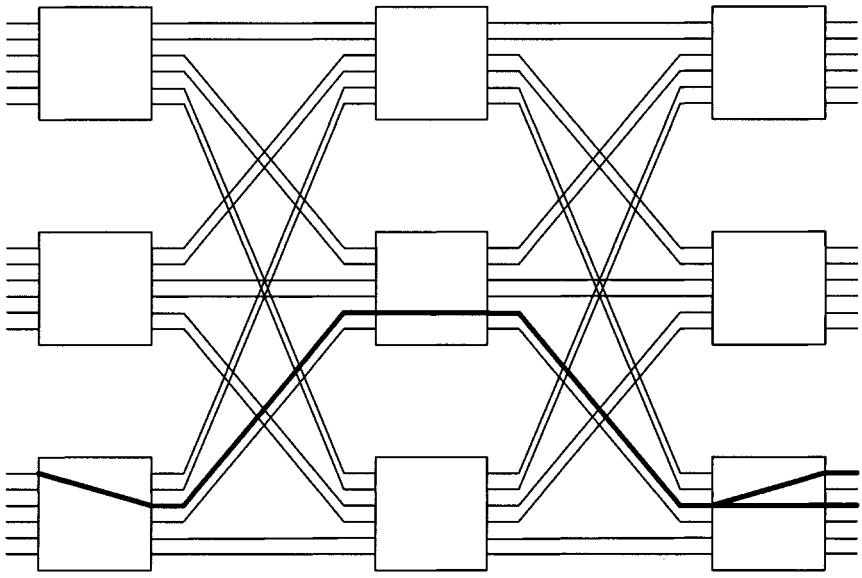


Figure 5.5c. Non-duplication routing - example 3

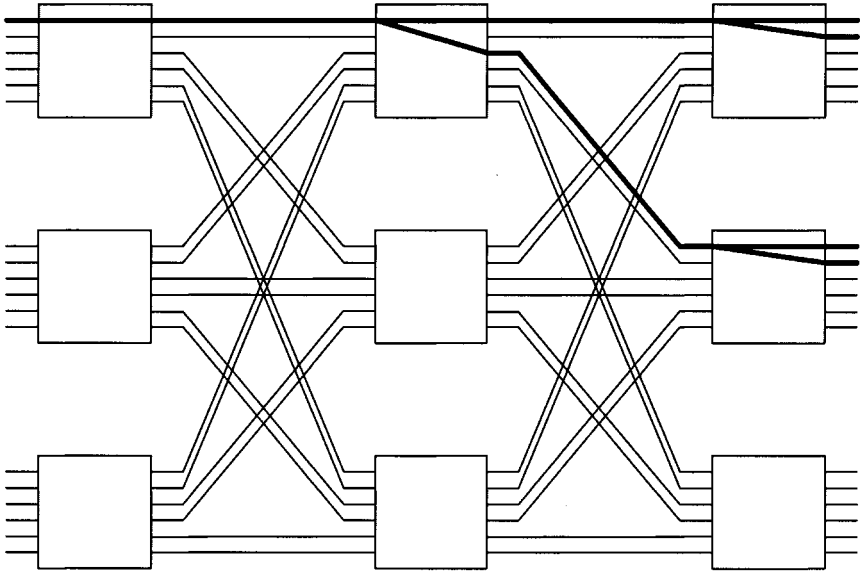


Figure 5.5d. Non-duplication routing - example 4

switch O_j), but no more than $(r_2-1)n_2$ such connections can be set up. These connections will occupy $a = \min\{n_1-1, (r_2-1)n_2\}$ middle stage switches. Similarly, to the third stage switch O_j there may be n_2-1 connections from the first stages switches (others than switch I_i), but no more than $(r_1-1)n_1$ such connections can be set up. These connections may occupy another set of $b = \min\{n_2-1, (r_1-1)n_1\}$ middle stage switches. In the worst case these sets of switches are disjoint and one more switch is needed to set up the connection (I_i, O_j). This state is shown in Fig. 5.6. We have:

$$m \geqslant \min\{n_1-1; (r_2-1)\,n_2\} + \min\{n_2-1; (r_1-1)\,n_1\} + 1. \quad (5.2)$$

When n_1-1 and n_2-1 are minima we have

$$m \geqslant m_1 = n_1 + n_2 - 1. \quad (5.3)$$

When n_1-1 and $(r_1-1)n_1$ are minima we have

$$m \geqslant m_2 = n_1 - 1 + (r_1-1)\,n_1 + 1 = n_1 r_1. \quad (5.4)$$

When $(r_2-1)n_2$ and n_2-1 are minima we have

$$m \geqslant m_3 = (r_2-1)\,n_2 + n_2 - 1 + 1 = n_2 r_2. \quad (5.5)$$

It should be noted, that both $(r_2-1)\,n_2$ and $(r_1-1)\,n_1$ cannot be minima at the same time so finally we obtain:

$$m \geqslant \min\{m_1; m_2; m_3\} = \min\{n_1 + n_2 - 1; r_1 n_1; r_2 n_2\}. \quad (5.6)$$

Necessity can be proved by showing the set of events leading to the blocking state when less switches are used in the middle stage. Since the network is strict-sense nonblocking it should be nonblocking for any algorithm used. Let us assume that the quasi-random algorithm is used, there is not any connection in the switching fabric at the beginning, and path searching for the first connection will start from the center stage switch M_1. The following set of events lead to the blocking state:

Step 1 Set up $a = \min\{n_1-1; (r_2-1)\,n_2\}$ connections from switch I_1 to switches O_j, $2 \leqslant j \leqslant r_2$. These connections will occupy switches M_1 to M_a.

Step 2 Set up $b = \min\{n_2-1; (r_1-1)\,n_1\}$ connections from switches I_i, $2 \leqslant i \leqslant r_1$ to switch O_1. These connections will occupy switches numbered form M_{a+1} to M_{a+b}.

Step 3 Connection (I_1, O_1) is blocked and one more switch is needed.

□

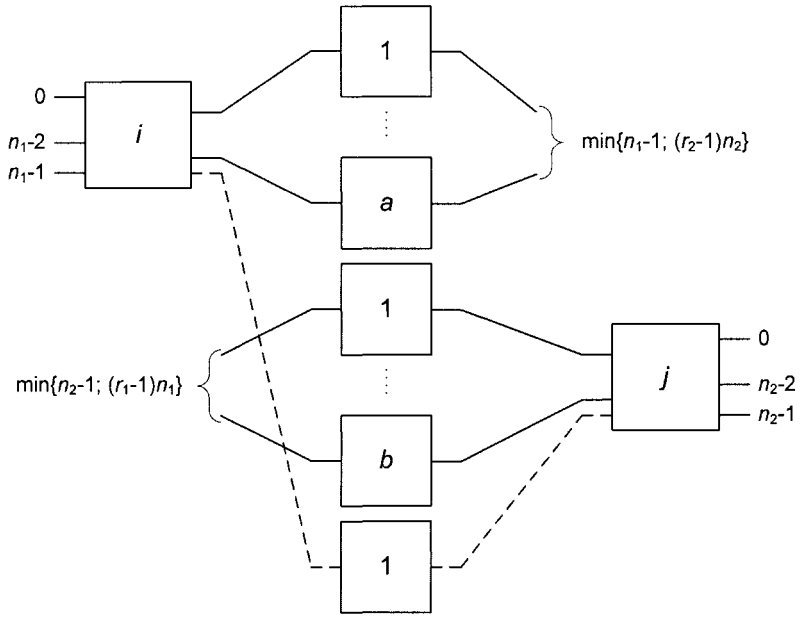


Figure 5.6. The worst state of $C_{SD}(n_1, r_1, m, n_2, r_2, 1)$

5.1.3.2 Multicast Connections

Multicast Clos switching fabrics were considered by many authors, however most of the results deal with the wide-sense nonblocking conditions, since it is easier to propose efficient routing strategies which leads to fewer middle stage switches required. The first result for strictly and rearrangeable nonblocking three stage Clos networks were given by Masson and Jordan [112]. However, they assume that a multicast connection to outputs of the same third stage switch will be only set up in this switch (no-split strategy). Therefore, their results are in fact wide-sense nonblocking conditions. Strict-sense nonblocking conditions for multicast connections have rather theoretical value, since wide-sense nonblocking conditions for different routing strategies were given, and wide-sense nonblocking switching fabrics are less costly than strict-sense nonblocking ones. We will give here without proofs main results given in the literature. Also blocking characteristics of Clos switching fabric with multicast connections were considered in some papers [174], [155].

Switching fabrics with closed-end q-cast connections, $1 \leqslant q \leqslant q_2$ where considered by Hwang and Liew. They proved following theorem [57, 56]:

THEOREM 5.2 *$C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ is nonblocking in the strict sense for q_2-connections if and only if*

$$m \geqslant \min\{(N_1 - 1)\, q_2 + 1; (n_1 - 1)\, q_2 + n_2; N_2\}. \tag{5.7}$$

This result was generalized by Listanti and Veltri to v-dilated switching fabric [108].

THEOREM 5.3 *$C_{SD}(n_1, r_1, m, n_2, r_2, v)$ is nonblocking in the strict sense for q_2-connections if and only if*

$$\begin{aligned} m \quad \geqslant \quad & \left\lfloor \frac{\min\{(n_1 - 1)\, q_2; (r_2 - 1)\, n_2\}}{v} \right\rfloor + \\ & + \left\lfloor \frac{\min\{n_2 - 1; (r_1 - 1)\, n_1 q_2\}}{v} \right\rfloor + 1. \end{aligned} \tag{5.8}$$

They also showed that this theorem is true for both open-end and closed-end connections.

Strict-sense nonblocking conditions were also generalized to switching fabrics with q-cast connections, where $1 \leqslant q_1 \leqslant q \leqslant q_2 \leqslant N_2$ by Giacomazzi and Trecordi [45], who gave sufficient conditions. Necessary and sufficient conditions were given by Pattavina and Tesei [139]:

THEOREM 5.4 *$C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ is nonblocking in the strict sense for q-connections, where $1 \leqslant q_1 \leqslant q \leqslant q_2 \leqslant N_2$ if and only if*

$$\begin{aligned} m \quad \geqslant \quad & \min\{(n_1 - 1)\, q_2 + n_2; (N_1 - 1)\, q_2 + 1; \\ & N_2 - q_1 + 1; \left(\left\lfloor \frac{N_2}{q_1} \right\rfloor - 1\right) q_2 + 1\Big\}. \end{aligned} \tag{5.9}$$

In above theorems it was assumed that switches in all stages have fan-out capability. If this is not true, i.e., switches in one stage have no fan-out capability, nonblocking conditions are different. However, when switches in stage i, $1 \leqslant i \leqslant 3$ have no fan-out capability, it corresponds to i-split routing strategy and may be considered as wide sense nonblocking conditions.

5.1.4 Wide-sense Nonblocking Conditions

5.1.4.1 Unicast Connections

Wide-sense nonblocking conditions are always associated with a particular algorithm or a class of algorithms used to route a new connection. Several authors give different lower bounds for $C_{SD}(n, r, m, 1)$ which show that no algorithms given in 5.1.2 leads to the savings in the number of the center stage switches. Up until now, the only result is due to Beneš [9], who proved the following theorem:

THEOREM 5.5 *$C_{SD}(n, m, 2, 1)$ is nonblocking in the wide sense when save the unused or packing algorithm is used if:*

$$m \geqslant \left\lfloor \frac{3n}{2} \right\rfloor \tag{5.10}$$

Proof [56]. It is sufficient to prove this theorem for save the unused algorithm, since packing algorithm also saves the unused center stage switches when it is unnecessary. Let the switching fabric is at state S, and let $\mathbb{S}$ denote a set of center stage switches carrying at least one connection in this state. We may only have four types of connections, namely: (I_1, O_1), (I_1, O_2), (I_2, O_1), and (I_2, O_2). Let $\mathbb{S}_i$, $1 \leqslant i \leqslant 6$ denote sets of center stage switches carrying following connections:
$\mathbb{S}_1$ carries connections (I_1, O_1),
$\mathbb{S}_2$ carries connections (I_1, O_2),
$\mathbb{S}_3$ carries connections (I_2, O_1),
$\mathbb{S}_4$ carries connections (I_2, O_2),
$\mathbb{S}_5$ carries connections (I_1, O_1) and (I_2, O_2),
$\mathbb{S}_6$ carries connections (I_1, O_2) and (I_2, O_1).
We shall show by induction that following inequalities holds:

$$\sum_{i=1}^{6} |\mathbb{S}_i| \leqslant \left\lfloor \frac{3n}{2} \right\rfloor, \tag{5.11}$$

$$|\mathbb{S}_1| + |\mathbb{S}_4| + |\mathbb{S}_5| \leqslant n, \tag{5.12}$$

$$|\mathbb{S}_2| + |\mathbb{S}_3| + |\mathbb{S}_6| \leqslant n, \tag{5.13}$$

when starting from the empty state. The first inequality means that at most $\lfloor 3n/2 \rfloor$ middle stage switches are occupied in state S. The second (third) inequality means that if there are connections (I_1, O_1) and (I_2, O_2) ((I_1, O_2) and (I_2, O_1)), then they will be paired in middle stage switches. In other words, we can say that if there are two connections (I_1, O_1) and (I_2, O_2) ((I_1, O_2) and (I_2, O_1)), they will be set up through the same middle stage switch. It is obvious, that these inequalities are true in empty state. Let S change to S' by adding or removing one connection. Inequalities (5.11), (5.12), and (5.13) should be true for S' if they are true for S. When one of existing connections is disconnected, it is obvious that these inequalities are true. Assume now, that the new connection is (I_2, O_2). It can be set up only through one of empty middle stage switches or one of switches in $\mathbb{S}_1$. When $\mathbb{S}_1$ is not empty, one of its switches is to be used. Thus the selected switch will now carry connections (I_1, O_1) and (I_2, O_2), so it is moved to $\mathbb{S}_5$. We have

$|\mathbb{S}'_1| = |\mathbb{S}_1| - 1$ and $|\mathbb{S}'_5| = |\mathbb{S}_5| + 1$, so inequalities (5.11), (5.12), and (5.13) are still true. When $\mathbb{S}_1$ is empty we have:

$$|\mathbb{S}_3| + |\mathbb{S}_4| + |\mathbb{S}_5| + |\mathbb{S}_6| \leqslant n - 1, \tag{5.14}$$

since at most $n - 1$ connections could be already set up from I_2,

$$|\mathbb{S}_2| + |\mathbb{S}_4| + |\mathbb{S}_5| + |\mathbb{S}_6| \leqslant n - 1, \tag{5.15}$$

since at most $n - 1$ connections could be already set up to switch O_2, and from the induction hypothesis (5.13) we have

$$|\mathbb{S}_2| + |\mathbb{S}_3| \leqslant n. \tag{5.16}$$

After adding (5.14), (5.15), and (5.16) we obtain

$$2\sum_{i=1}^{6} |\mathbb{S}_i| \leqslant 3n - 2, \tag{5.17}$$

or

$$\sum_{i=1}^{6} |\mathbb{S}_i| \leqslant \left\lfloor \frac{3n}{2} \right\rfloor - 1, \tag{5.18}$$

This means that in S' obtained from S by adding connection (I_2, O_2) inequality (5.11) is still true and this connection will be routed through an empty switch in the middle stage. Inequalities (5.12) and (5.13) also holds because

$$|\mathbb{S}'_1| + |\mathbb{S}'_4| + |\mathbb{S}'_5| = 1 + |\mathbb{S}_1| + |\mathbb{S}_4| + |\mathbb{S}_5| = 1 + |\mathbb{S}_4| + |\mathbb{S}_5| \leqslant n, \tag{5.19}$$

since switches in $\mathbb{S}_4$ and $\mathbb{S}_5$ carries connections form switch I_2.
Similar considerations can be done when other type of connection is to be added. □

The $C_{SD}(n, 2, \lfloor 3n/2 \rfloor)$ is not practical, since the number of crosspoints is always greater than in the crossbar switch of the same capacity. In 1979 Melas and Milewski showed that for sequential routing the minimum number of the center stage switches is the same as for the strict sense nonblocking case [113].

THEOREM 5.6 *$C_{SD}(n, r, m, 1)$ is wide-sense nonblocking under sequential routing and minimum index routing for $r \geqslant 2$ if and only if: $m \geqslant 2n - 1$.*

Proof. Similarly as in Theorem 4.3 we will consider only the minimum index algorithm. It is sufficient to show the necessary condition for $r = 2$. The set of events:

Step 1 Set up two connections $(I_1,\ O_1)$ through switches M_1 and M_2. Disconnect connection $(I_1,\ O_1)$ realized through switch M_1 and set up connection $(I_2,\ O_2)$. This connection will use switch M_1. The state of the switching fabric is:

connection	the second stage switch
(I_2, O_2)	M_1
(I_1, O_1)	M_2

Step 2 Set up two connections $(I_1,\ O_2)$ through middle stage switches M_3 and M_4. Disconnect connection $(I_1,\ O_2)$ realized through switch M_3 and set up connection $(M_2,\ I_1)$. This connection will use switch M_3. Disconnect connection $(I_1,\ O_1)$ from switch M_2 and set up connection $(I_1,\ O_2)$ through this switch. Disconnect $(I_2,\ O_2)$ from switch M_1 and set up $(I_2,\ O_1)$ through this switch. The state of the switching fabric is changed to:

connection	the second stage switch
(I_2, O_1)	M_1
(I_1, O_2)	M_2
(I_2, O_1)	M_3
(I_1, O_2)	M_4

Assume that after step $i-1$ following state is in the switching fabric:

connection	the second stage switch
(I_2, O_1)	M_1
(I_1, O_2)	M_2
(I_2, O_1)	M_3
(I_1, O_2)	M_4
...	...
(I_2, O_1)	M_{2i-3}
(I_1, O_2)	M_{2i-2}

Step *i* Set up two connections (I_1, O_1) through middle stage switches M_{2i-1} and M_{2i}. Disconnect connection (I_1, O_1) from switch M_{2i-1} and set up connection (I_2, O_2). This connection will use switch M_{2i-1}. Then do:

Disconnect (I_1, O_2) from M_{2i-2} and set up (I_1, O_1) through M_{2i-2}.
Disconnect (I_2, O_1) from M_{2i-3} and set up (I_2, O_2) through M_{2i-3}.
Disconnect (I_1, O_2) from M_{2i-4} and set up (I_1, O_1) through M_{2i-4}.
Disconnect (I_2, O_1) from M_{2i-5} and set up (I_2, O_2) through M_{2i-5}.
...
Disconnect (I_1, O_2) from M_4 and set up (I_1, O_1) through M_4.
Disconnect (I_2, O_1) from M_3 and set up (I_2, O_2) through M_3.
Disconnect (I_1, O_2) from M_2 and set up (I_1, O_1) through M_2.

Disconnect (I_2, O_1) from M_1 and set up (I_2, O_2) through M_1. The state of the switching fabric is changed to:

connection	the second stage switch
(I_2, O_2)	M_1
(I_1, O_1)	M_2
(I_2, O_2)	M_3
(I_1, O_1)	M_4
...	...
(I_2, O_2)	M_{2i-3}
(I_1, O_1)	M_{2i-2}
(I_2, O_2)	M_{2i-1}
(I_1, O_1)	M_{2i}

During execution of step i no more than $i+1$ connections were set up in any of the outer stage switches. Step i can be executed until $i = n-1$. After step $n-1$ we have $n-1$ connections (I_1, O_1) and $n-1$ connections (I_2, O_2) (or (I_1, O_2) and (I_2, O_1), depending on n being odd or even) which are set up through $2n-2$ different second stage switches. Connection (I_1, O_2) has to be set up through the switch $2n-1$. □

In the same paper Melas and Milewski proved that for save the unused routing $C_{SD}(n, r, m, 1)$ with $r \geqslant 2$ requires $2n-1$ middle stage switches to be nonblocking if $n \leqslant 2^{r-1}$ and $2n-2$ switches if $2^{r-1} < n \leqslant 2^r$. They also showed that if $r > 2$ and the packing routing which always uses the most heavily loaded middle stage switch $C_{SD}(n, r, m, 1)$ also require $2n-1$ switches in the middle stage when $2^{r-1} < n \leqslant 2^r$. The proof starts from the initial state in which there are n connections $(I_1,\ O_1)$, n connections $(I_2,\ O_2)$, ..., n connections $(I_r,\ O_r)$. This state is called the state, n-uniform [56]. Du, Fishburn, Gao, and Hwang [35, 56] extended this result for a class of algorithms for which the switch fabric can reached the n-uniform state. They showed, that if for any algorithm $C_{SD}(n, r, m, 1)$ enters the n-uniform state, then it is wide-sense nonblocking under this algorithm only if $m \geqslant 2n - \lceil n/2^{r-1} \rceil$. The wide-sense nonblocking conditions for $C_{SD}(n, r, m, 1)$ where also considered by Yang and Wang [175]. They proved that at least $\lfloor n\,(2 - 1/F_{2r-1}) \rfloor$ middle stage switches are necessary under packing strategy, where F_{2r-1} is the Fibonacci number.

For packing strategy more precise rules are needed to determine which out of several equally loaded center stage switches should be used. When the switch recently used to set up the previous connection is used, the following theorem holds.

THEOREM 5.7 *$C_{SD}(n, r, m, 1)$ is wide-sense nonblocking under packing strategy for $r \geqslant 3$ if and only if $m \geqslant 2n-1$.*

Proof. It is sufficient to prove the "only if" part for $r = 3$. For such switching fabric it is possible to show that there is a sequence of connections and disconnections which lead to the blocking state when $m = 2n - 2$. Following events fulfill this condition.

Step 1 Set up n connections (I_1, O_1). These connections will occupy switches M_1 do M_n.

Step 2 Set up connection (I_2, O_2). This connection will occupy switch M_n

Step 3 Disconnect connection (I_1, O_1) which was set up through switch M_n.

Step 4 Repeat steps 4.1 - 4.5 $n - 1$ times.

- **Step 4.1** Set up connection (I_1, O_2). This connection will have to occupy switch M_{n+i}.
- **Step 4.2** Set up connection (I_3, O_3). Since all center stage switches have occupancy equal to 1, this connection will be set up through switch M_{n+i}.
- **Step 4.3** Disconnect connection (I_1, O_2).
- **Step 4.4** Set up connection (I_2, O_2). Since all center stage switches have occupancy equal to 1, this connection will be set up through switch M_{n+i}.
- **Step 4.5** Disconnect connection (I_3, O_3).

After these steps we have $n - 1$ connections (I_1, O_1) set up through $n - 1$ switches and $n - 1$ connection (I_2, O_2) set up through other $n - 1$ switches. Connection (I_1, O_2) is blocked and will have to occupy the center stage switch numbered M_{2n-1}.

For $r \geqslant (n-1)\begin{pmatrix} 2n-2 \\ n-1 \end{pmatrix} + 1$, Tsai, Wang, and Hwang showed that no algorithm exists which leads to the reduction in the number of required center stage switches [160, 161].

5.1.4.2 Multicast Connections

Wide-sense nonblocking conditions for $C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ where derived in some papers depending of different strategies used for setting up multicast connections. The first sufficient condition where given by Masson and Jordan [112]. They consider the switching fabric as strict-sense nonblocking, however, they assumed the no-split restricted strategy in their considerations. Necessary and sufficient conditions were given by Hwang and Liaw [56, 57]:

THEOREM 5.8 *$C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ is wide-sense nonblocking for open-end and closed-end q-cast connections, $1 \leqslant q \leqslant q_2 \leqslant N_2$ under no-split restriction strategy if and only if:*

$$\begin{aligned} m \geqslant & \min\{N_1 + (n_1 - 1)(r_2 - 1); (n_1 - 1)(r_2 - 1) + n_2; \\ & N_1 + (n_1 - 1)(q_2 - 1); (n_1 - 1)q_2 + n_2; N_2\}. \end{aligned} \tag{5.20}$$

The conditions for switching fabrics under the 1-split restriction strategy may be derived from conditions for multiconnection switching fabrics considered by Hwang and Jajszczyk [51]. Strategy 2 for multiconnections set up of the cited paper correspond to the 1-split restriction strategy for multicast connections. We can therefore write:

THEOREM 5.9 *$C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ is wide-sense nonblocking for open-end and closed-end q-cast connections, $1 \leqslant q \leqslant q_2 \leqslant N_2$ under 1-split restriction strategy if and only if:*

$$\begin{aligned} m \geqslant & \max_{1 \leqslant l \leqslant q_2} \{\min\{n_1 - 1 + l(n_2 - 1) + 1; n_1 r_1; \\ & n_2(r_2 - l) + l(n_2 - 1) + 1\}\}. \end{aligned} \tag{5.21}$$

Nonblocking conditions of $C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ under 2-split and 3-split restriction strategies were given by Hwang [56]:

THEOREM 5.10 *$C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ is wide-sense nonblocking for open-end and closed-end q-cast connections, $1 \leqslant q \leqslant q_2 \leqslant N_2$ under 2-split restriction strategy if and only if:*

$$\begin{aligned} m \geqslant & \min\{(N_1 - 1)q_2 + \min\{q_2; r_2\}; N_2; \\ & (n_1 - 1)q_2 + n_2 - 1 + \min\{q_2; r_2\}\}. \end{aligned} \tag{5.22}$$

THEOREM 5.11 *$C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ is wide-sense nonblocking for closed-end q-cast connections, $1 \leqslant q \leqslant q_2 \leqslant N_2$ under 3-split restriction strategy if and only if:*

$$m \geqslant \min\{N_1 q_2; (n_1 - 1)q_2 + n_2; N_2;\}. \tag{5.23}$$

The p-limited no-split restriction strategy was proposed by Yang and Masson [173]. They proved the following theorem:

THEOREM 5.12 *$C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ is wide-sense nonblocking for closed-end q-cast connections, $1 \leqslant q \leqslant q_2 \leqslant r_2$ under p-limited no-split restriction strategy if:*

$$m \geqslant (n_1 - 1)p + (n_2 - 1)q_2^{1/p}. \tag{5.24}$$

Another upper bound for $C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ with multicast connections under p-limited no-split restriction strategy was considered in [28, 26]. Let us assume that the new q-cast connection $\langle x, \mathbb{Y}\rangle$, $1 \leqslant q \leqslant q_2 \leqslant r_2$, is to be added. Since p-limited no-split restriction strategy is considered, i.e., connecting paths to outputs of the same third stage switch is spread only in this switch, we may assume that $\mathbb{Y} = \{y_1, y_2, \ldots, y_q\}$, $y_w \in \mathbb{O}_{j_w}$, $1 \leqslant w \leqslant q$, $j_w \neq j_z$ for $z \neq w$, where $\mathbb{O}_{j_w}$ denotes the set of outputs of switch O_{j_w}. For space division switching, in any switch O_{j_w} $n_2 - 1$ outputs may be already occupied and they will use at most $n_2 - 1$ inputs of this switch, connected to middle stage switches. Similarly, at switch I_i containing input terminal x $n_1 - 1$ inputs may be already occupied.

The state matrix $\mathbf{C}$ will be used to represent the state of links between m_2 middle stage switches and q output stage switches. The matrix $\mathbf{C}$ is of size $q \times n_2$. Each row represents one of q third stage switches and each column represents one of n_2 outputs of these switches. The matrix is defined as follows:

DEFINITION 5.13

$$\mathbf{C} = \left[c_{j,a} = \begin{cases} k, & \text{if a connection to output } a \text{ of } O_j \text{ is set} \\ & \text{up through switch } M_k, 1 \leqslant k \leqslant m_2; \\ 0, & \text{if there is no such connection.} \end{cases} \right] \tag{5.25}$$

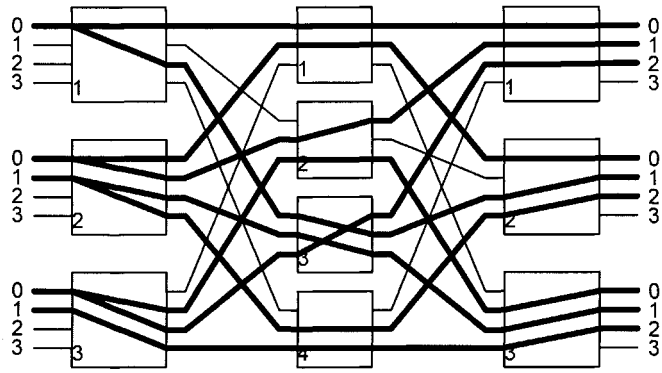
The state matrix $\mathbf{C}$ has following properties:

1 It contains only m_2 different numbers (since $1 \leqslant k \leqslant m_2$);

2 Any row of the matrix $\mathbf{C}$ has to contain no more than n_2 different numbers since there is only one link between the given output stage switch and any of the middle stage switches.

An example of matrix $\mathbf{C}$ for the switching fabric with $n_2 = 4$, $q = 3$, and $p = 2$, is presented in Fig. 5.7. Element $c_{1,0} = 1$ means that the connection to output 0 of switch O_1 is set up through switch M_1.

Since outputs of the third stage switches which take part in a multicast connection are not important, let $(I_i, \mathbb{Y})$ denote a new multicast connection, where $\mathbb{Y} = \{y_1, y_2, \ldots, y_q\}$, $y_j \in O_j$, i.e., any input terminal in I_i is to be connected with one terminal in each of switches $O_1, O_2, \ldots, O_q$. This connection is to be set up through no more than p middle stage switches. In each of q switches $n_2 - 1$ connections are already set up. Without loss of generality we can assume that these connections occupy first $n_2 - 1$ outputs of each switch. In this case different numbers of middle stage switches are placed in each row of the first $n_2 - 1$ columns of matrix $\mathbf{C}$ (see Fig. 5.7). To set the new connection we have to choose



q= n_2=	0	1	2	3
1	1	2	3	0
2	1	3	4	0
3	2	3	4	0

Figure 5.7. An example of matrix **C** for the switching fabric with $n_2 = 4$, $q = 3$, and $p = 2$

p different numbers from m_2 and placed them in the last column of the state matrix. For instance, if in the example of Fig. 5.7 $m_2 = 4$ and $p = 2$, we cannot set up a connection to switches O_1, O_2, and O_3 through any two of m_2 switches. One more switch is needed in the middle stage to set up this connection. When it is not possible to find p out of m_2 numbers such that they could be placed in the state matrix, then it is the blocking state of the switching fabric (or state matrix **C**). It is obvious, that if we have p numbers chosen, and one row of matrix **C** contains all these p numbers, then the matrix is in the blocking state for these p numbers.

DEFINITION 5.14 *Let $\mathbb{S}_j$ be the set of subsets where each subset contains p numbers from all numbers placed in row j of the matrix* **C**.

THEOREM 5.15 *If $\bigcup_{1 \leqslant j \leqslant q} \mathbb{S}_j$ contains all possible combinations p from m_2, then the state matrix is in the blocking state.*

Proof. Let us assume that we have $k_1, k_2, \ldots, k_p$ numbers, $1 \leqslant k_w \leqslant m_2$, $1 \leqslant w \leqslant p$. If $\bigcup_{1 \leqslant j \leqslant q} \mathbb{S}_j$ contains all possible combinations it means that it

contains also combination $k_1, k_2, \ldots, k_p$. Therefore numbers $k_1, k_2, \ldots, k_p$ are placed in one of the rows of matrix $\mathbf{C}$, and any of these numbers cannot be placed in this row. So matrix $\mathbf{C}$ is in the blocking state. □

The maximum number of m_2 for which state matrix $\mathbf{C}$ can be in the blocking state is determine by the following theorem:

THEOREM 5.16 *State matrix* $\mathbf{C}$ *can be in a blocking state if:*

$$m_2 \leqslant \max_z \left\{ \binom{z}{p} \leqslant q \times \binom{n_2 - 1}{p} \right\}, \tag{5.26}$$

where z *is an integer.*

Proof. Since $|\mathbb{S}_j| \leqslant \binom{n_2 - 1}{p}$, then $\left| \bigcup_{1 \leqslant j \leqslant q} \mathbf{S}_j \right| \leqslant q \times \binom{n_2 - 1}{p}$. It means that in matrix $\mathbf{C}$ no more than $q \times \binom{n_2 - 1}{p}$ different combinations p from z can be realized. If $\binom{z}{p} > q \times \binom{n_2 - 1}{p}$ then we can find such combination of p numbers which is not realized in matrix $\mathbf{C}$. According to Theorem 5.15 state matrix $\mathbf{C}$ can be in the blocking state for any z for which

$$\binom{z}{p} \leqslant q \times \binom{n_2 - 1}{p} \tag{5.27}$$

The maximum integer z for which (5.27) is true constitutes the maximum number of middle stage switches for which state matrix $\mathbf{C}$ can be in the blocking state. □

The upper bound for non-blocking operation of $C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ is given by the following theorem:

THEOREM 5.17 $C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ *is wide-sense nonblocking for closed-end* q*-cast connections,* $1 \leqslant q \leqslant q_2 \leqslant r_2$ *under p-limited no-split restriction strategy if*

$$m \geqslant \max_{1 \leqslant j \leqslant q_2} \left\{ \min_{1 \leqslant p \leqslant \min\{j; n_2 - 1\}} \{\min\{A_1; B_1\} + \min\{A_2; B_2\} + 1\} \right\}, \tag{5.28}$$

where

$$\begin{aligned} A_1 &= p \times (n_1 - 1), \\ B_1 &= (r_2 - p) \times n_2, \\ A_2 &= \max_z \left\{ \binom{z}{p} \leqslant j \times \binom{n_2 - 1}{p} \right\}, \\ B_2 &= p \times (r_1 - 1) \times n_1. \end{aligned}$$

Proof. Let us assume that we want to set up a new multicast connection $(I_i, \mathbb{Y})$, $\mathbb{Y} = \{O_1, O_2, \ldots, O_j\}$, $j \leqslant q_2$. In switch I_i $n_1 - 1$ connections may be set up and these connections may occupy at most $A_1 = p \times (n_1 - 1)$ middle stage switches, provided, that in the remaining third stage switches there are enough outputs to accept these connections (the discussed state is shown in Fig. 5.8). In the third stage we have $r_2 - p$ such switches, each of them can accept n_2 connections. It means that not more than $B_1 = (r_2 - p) \times n_2$ middle stage switches are needed. So no more than $m_1 = \min\{A_1; B_1\}$ switches will be inaccessible for the new connection. On the other hand in each out of j third stage switches we may already have at most $n_2 - 1$ connections set up. These connections, according to Theorem 5.16 may occupy $A_2 = \max_z \left\{ \binom{z}{p} \leqslant j \times \binom{n_2 - 1}{p} \right\}$ middle stage switches in such a way that through any set of p switches out of A_2 it is not possible to set up the new connection. However, similarly as in the first stage switch, connections to $n_2 - 1$ outputs of j third stage switches can be set up if there are enough inputs in the first stage switches except switch I_i. So no more than $(r_1 - 1) \times n_1$ connections can be set up in the remaining first stage switches and they will occupy not more than $B_2 = p \times (r_1 - 1) \times n_1$ middle stage switches. So we have $m_2 = \min\{A_2; B_2\}$. In the worst case these sets of m_1 and m_2 switches are disjoint and one more switch is needed to set up the new connection. So $m \geqslant m_1 + m_2 + 1$. In the switching fabric a multicast connection may be set up to j different third stage switches, $1 \leqslant j \leqslant q_2 \leqslant r_2$. For given j, the number p is to be found, $1 \leqslant p \leqslant j$, for which the smallest number of center stage switches will be engaged by the connection. Therefore, this number m must be maximized through all j and minimized through all p. □

For $q_2 = 1$ we have the space-division switching fabric with point-to-point connections. In this case $A_1 = n_1 - 1$, $B_1 = n_2(r_2 - 1)$, $A_2 = n_2 - 1$, and $B_2 = n_1(r_1 - 1)$, so $m \geqslant \min\{n_1 - 1; n_2(r_2 - 1)\} + \min\{n_2 - 1; n_1(r_1 - 1)\} + 1$ and we obtain conditions given in Theorem 5.1.

Comparison of the required number of center stage switches for different routing strategies is given in Table 5.1. In this table it is assumed

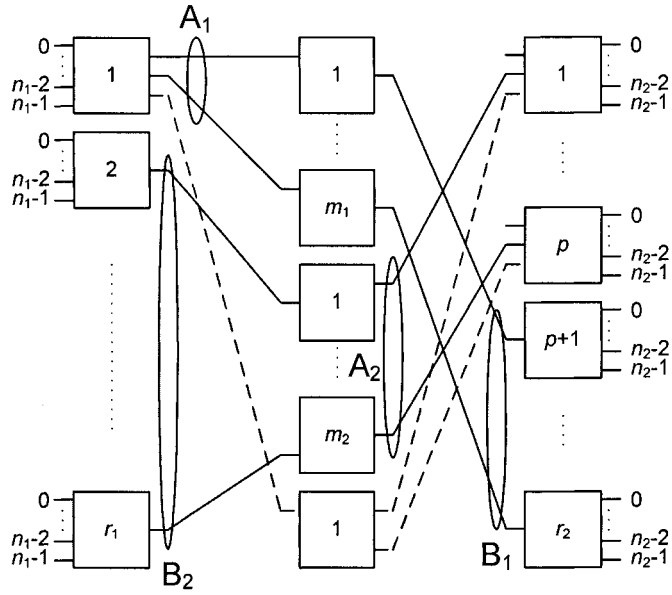


Figure 5.8. The worst scenario for $C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ with multicast connections and p-limited no-split restriction strategy

that $q_2 = N_2$ for SNB and i-split restricted strategy and $q_2 = r_2$ for no-split restriction and p-limited strategies (it is justified since connections to the same outputs of any third stage switch are spread in this switch). Two values for p-limited strategy are given, column p-limited 1 contains values calculated according to Theorem 5.12, while values in column p-limited 2 were calculated using Theorem 5.17. It can be seen that p-limited strategy requires the lowest number of center stage switches and the best upper bound is given in Theorem 5.17. Strategies 2-split restriction and 3-split restriction requires the same number of switches as the strict-sense nonblocking switching fabric of the same capacity. Similarly, switching fabrics with 1-split restriction and no-split restriction strategies require the same number of center stage switches.

The number of center stage switches versus n and r in $C_{SD}(n, r, m, 1)$ for upper bounds determined by Theorems 5.12 and 5.17 are plotted in Figures 5.9 and 5.10, respectively. It can be seen that Theorem 5.17 gives better upper bounds than the previous one. This is also true for asymmetrical switching fabrics. For instance for $C_{SD}(4, 4, m, 4, 64, 1)$ and $q_2 = 64$ the first upper bound gives following results: $p = 3$ and $m \geqslant 22$. According to Theorem 5.17 we have $m \geqslant \min\{3; 252\} + min\{60; 12\} + 1 = 16$ and this value is obtained for $p = 1$ and $j = 20$.

		SNB	1-split	2-split	3-split	no-split	p-limited 1	p-limited 2
r=8	n=4	32	25	32	32	25	15	14
	8	64	57	64	64	57	34	30
	16	128	121	128	128	121	73	63
	32	256	249	256	256	249	150	131
r=16	n=4	64	49	64	64	49	17	15
	8	128	113	128	128	113	39	38
	16	256	241	256	256	241	83	79
	32	512	497	512	512	497	172	163
r=32	n=4	128	97	128	128	97	19	16
	8	256	225	256	256	225	44	41
	16	512	481	512	512	481	93	91
	32	1024	993	1024	1024	993	192	184

Table 5.1. Number of center stage switches in $C_{SD}(n, r, m, 1)$ under different routing strategies

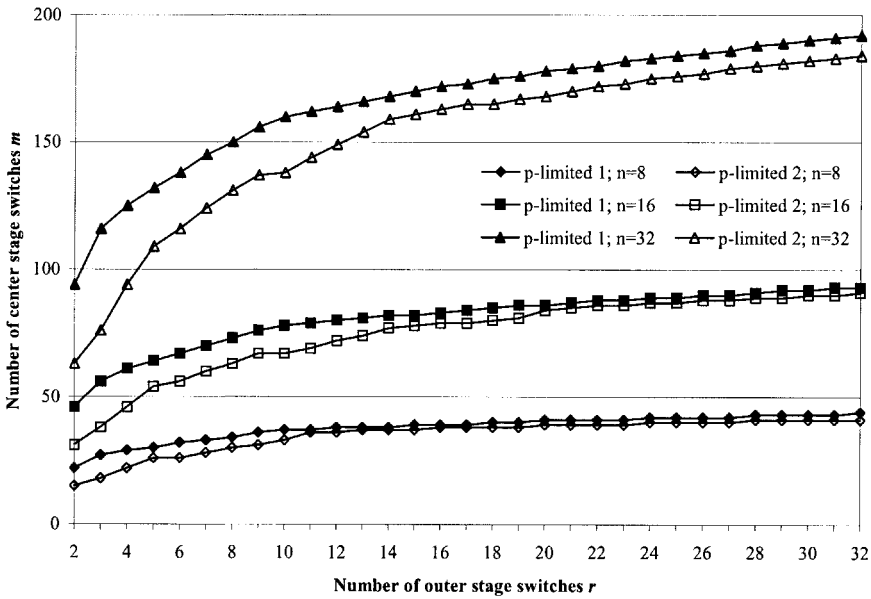


Figure 5.9. Number of center stage switches versus r in $C_{SD}(n, r, m, 1)$ with p-limited no-split restriction strategy

In this case $A_2 = 60$ and $B_2 = 12$. It means that 60 numbers are needed to block the state matrix $\mathbf{C}$. So at least 60 connections should be set up in the remaining first stage switches. But there are 3 such switches and only 12 connections that could be set up in these switches ($B_2 = 12$).

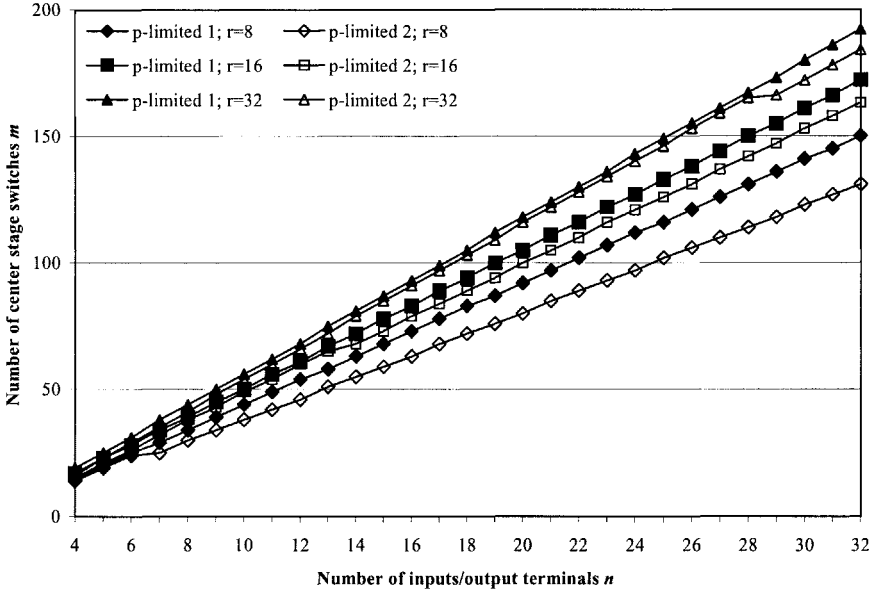


Figure 5.10. Number of center stage switches versus n in $C_{SD}(n, r, m, 1)$ with p-limited no-split restriction strategy

5.1.5 Rearrangeable Switching Fabrics

5.1.5.1 Unicast Connections

The condition under which the symmetrical three-stage two-sided space-division switching fabric is rearrangeable was given by Slepian [151] and was later formally proved by Duguid [40]. The respective theorem is known as Slepian-Duguid theorem.

THEOREM 5.18 *$C_{SD}(n, r, m, 1)$ is rearrangeable if and only if $m \geqslant n$.*

Proof. Necessity is obvious, since we have at least n switches in the center stage to set up connections from n inputs in any of the first stage switches. Sufficiency can be proved using Hall's theorem on distinct representatives [46]. We will show that every maximal assignment between input and output terminals can be realized by a state of the switching fabric. The maximal assignment is equivalent to a permutation

$$\Pi = \begin{pmatrix} 0 & 1 & 2 & \cdots & N-1 \\ \pi(0) & \pi(1) & \pi(2) & \cdots & \pi(N-1) \end{pmatrix}. \tag{5.29}$$

For the first stage switch I_i, $1 \leqslant i \leqslant r$, we assign the set $\mathbb{A}_i$, which contains the number of the third stages switches to which input terminals

incoming to the switch I_i are to be connected. The sum of any k sets $\mathbb{A}_{p_1}, \mathbb{A}_{p_2}, \ldots, \mathbb{A}_{p_k}$ is the set of the numbers of which third stage switches $p_1, p_2, \ldots, p_k$ should be connected. If this set contains fewer than k elements, then k first stage switches would be connected to fewer than k third stage switches. Hence, by connecting inputs of k first stage switches to output terminals, less than kn connections would be established. This contradicts the assumption that each input terminal is to be connected to one output terminal, and vice versa. Therefore the sum of k sets $\mathbb{A}_i$ does not contain fewer than k different elements and, by Hall's theorem concerning distinct representatives of subsets, for any permutation Π it is always possible to choose r input terminals incoming to different first stage switches, such that these terminals are to be connected to output terminals outgoing from different third stage switches. Such subassignment can always by realized using one center stage switch. Applying the same procedure for the rest of the network $n-1$ times, we can realize every maximal assignment using n middle stage switches. □

In the case of asymmetrical switching fabric the respective theorem is as follows:

THEOREM 5.19 *$C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ is rearrangeable if and only if* $m \geqslant \max\{n_1, n_2\}$.

Proof. A maximal assignment can be represented by a bipartite graph of maximum degree $d \leqslant \max\{n_1, n_2\}$. Such a graph can be colored using d colors and connections colored with the same color can always be realized using one center stage switch. □

5.1.5.2 Multicast Connections

Rearrangeable $C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ with multicast connections were first considered by Masson and Jordan [112]. They gave the conditions under which this switching fabric is rearrangeable under no-split restriction and 2-split restriction strategy. The latter case was strengthened by Hwang [52, 56] in the following theorem:

THEOREM 5.20 *The $C_{SD}(n_1, r_1, m, n_2, r_2, 1)$ is rearrangeable for q-cast connections, $1 \leqslant q \leqslant q_2 \leqslant N_2$ under 2-split restriction strategy if and only if:*

$$m \geqslant \max\{\min\{n_1 q_2; n_2 r_2\}; \min\{n_2; n_1 r_1\}\}. \tag{5.30}$$

The case when 1-split restriction and no-split restriction strategies are used was considered by Kirkpatrick, Klawe, and Peppenger [88]. Recently Jajszczyk considered $C_{SD}(n, r, m, 1)$ with multicast connections

under 1-split and 3-split restriction strategy. It means that connections are spread out only in switches of the center stage. This approach may be practical in large-scale cross-connects systems [66]. The assumed restrictions result in that each multicast connection is restricted to be connected to only one output terminal in each switch of the third stage, but this is not a real constraint for some cross-connect applications. The respective theorem is as follows:

THEOREM 5.21 *$C_{SD}(n,r,m,1)$ is rearrangeable for q-cast connections, $1 \leqslant q \leqslant q_2 \leqslant r$ under 1-split and 3-split restriction strategy if and only if:*

$$m \geqslant 2n - \left\lceil \frac{n}{q_2} \right\rceil . \tag{5.31}$$

Different strategies which may be used for setting up multicast connections result in a different number of middle stage switches required for wide-sense nonblocking and rearrangeable nonblocking operation. Let us compare the $C_{SD}(n,r,m,1)$ wide-sense and rearrangeable nonblocking broadcast switching fabrics [27]. For 2-split restriction strategy rearrangeable conditions are given in Theorem 5.20 and for 1-split restriction strategy wide-sense nonblocking conditions are given in Theorem 5.9. When more than one output port of the same output switch take part in the connection, then these outputs are connected by fan-out connection in the output switch, i.e., $q_2 = r$. From (5.30) we obtain that $C_{SD}(n,r,m,1)$ is rearrangeable for r-cast connections if

$$m \geqslant nr. \tag{5.32}$$

From (5.21) we obtain that $C_{SD}(n,r,m,1)$ is nonblocking for r-cast connections under 1-split restriction strategy if

$$m \geqslant \max_{1 \leqslant l \leqslant r} \{\min\{n + ln - l; nr; nr - l + 1\}\} . \tag{5.33}$$

For any l, $1 \leqslant l \leqslant r$, $rn \geqslant nr - l + 1$ so we obtain

$$m \geqslant \max_{1 \leqslant l \leqslant r} \{\min\{n + ln - l; nr - l + 1\}\} . \tag{5.34}$$

A comparison between formula (5.32) and (5.34) leads to the conclusion that wide-sense nonblocking $C_{SD}(n,r,m,1)$ under 1-split strategy contains always less switches in the middle stage then the rearrangeable switching fabric of the same capacity and under 2-split restricted strategy. This difference is equal to $r-1$. It is worth mention that from (5.34) for $r = 2$ we obtain $m \geqslant 2n - 1$. It means, that $C_{SD}(n,2,2n-1,1)$ is strict-sense nonblocking for point-to-point connections and wide-sense

		WNB 1-split	WNB p-limited	RNB 2-split	RNB 1 and 3-split
r=8	n=4	25	14	32	7
	8	57	30	64	15
	16	121	63	128	30
	32	249	131	256	60
r=16	n=4	49	15	64	7
	8	113	38	128	15
	16	241	79	256	31
	32	497	163	512	62
r=32	n=4	97	16	128	7
	8	225	41	256	15
	16	481	91	512	31
	32	993		1024	63

Table 5.2. Comparison of WNB and RNB $C_{SD}(n, r, m, 1)$ under different routing strategies

nonblocking for broadcast connections when 1-split restriction strategy is used. In Table 5.2 number of middle stage switches for rearrangeable nonblocking switching fabrics under 2-split restricted and 1-split and 3-split restricted strategies and wide sense nonblocking switching fabrics with 1-split and p-limited no-split strategies are compared.

5.1.6 Rearrangeable Algorithms

5.1.6.1 Single Connections

When the path searching algorithm fails, the new connection is blocked. To unblock it, a rearrangement algorithm is to be used. Slepian [151] and Paull [140] proposed several rearrangement algorithms. All of them use Paull's matrix $\mathbf{M}$ of size $r \times r$ for representing the state of the switching fabric. The Paull's matrix is also called a state matrix. Each row (column) in $\mathbf{M}$ corresponds to one first stage (third stage) switch. An entry $\mathbf{M}[i; j] = A$ means that the connection (I_i, O_j) is set up through the center stage switch M_A. It should be noted that there can be more connecting paths (but not more than n) between given switches I_i and O_j, so an entry $\mathbf{M}[i; j]$ may contain more than one center stage switch. The example of $C_{SD}(4, 4, 4, 1)$ switching fabric and its state matrix are shown in Figs. 5.11a and 5.11b, respectively. In this switching fabric the following connections are set up: $\langle 0, 0\rangle$ (i.e., (I_1, O_1)), $\langle 4, 8\rangle$ $((I_2, O_3))$, $\langle 12, 4\rangle$ $((I_4, O_2))$, through switch M_1, connections $\langle 1, 1\rangle$ $((I_1, O_1))$ and $\langle 5, 9\rangle$ $((I_2, O_3))$, through switch M_2, $\langle 10, 6\rangle$ $((I_3, O_2))$ and $\langle 14, 14\rangle$ $((I_4, O_4))$

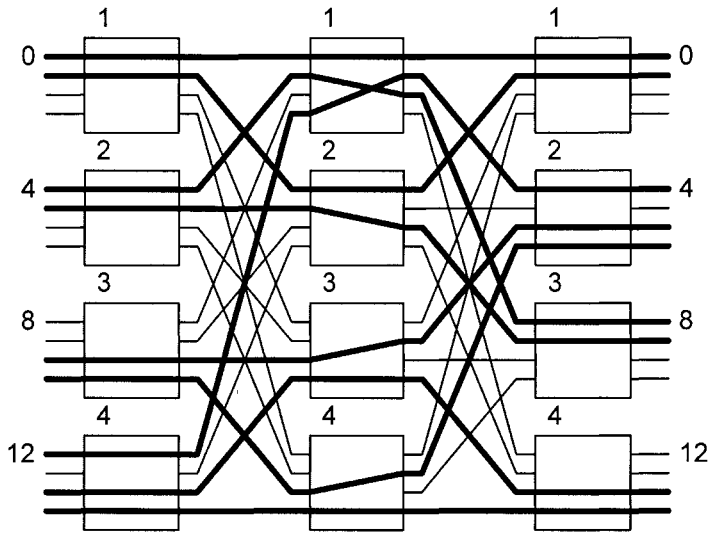


Figure 5.11a. A state of $C_{SD}(4,4,4,1)$ - an example

	1	2	3	4
1	1,2			
2			1,2	
3		3,4		
4		1		3,4

Figure 5.11b. The state matrix of $C_{SD}(4,4,4,1)$

through switch M_3, and $\langle 11,7\rangle$ $((I_3,O_2))$ and $\langle 15,15\rangle$ $((I_4,O_4))$ through switch M_4. The Paull's matrix has the following properties:

1 In each row there can be no more than n different elements, since each first stage switch has n inputs and only one link to every center stage switch.

2 In each column there can be no more than n different elements, since each third stage switch has n outputs and only one link from every center stage switch.

Let us assume that connection (I_i, O_j) is blocked. The input data for the algorithm are input switch I_i, output switch O_j, and Paull's matrix $\mathbf{M}$ representing the current state of the switching fabric. The algorithm is as follows [140]:

ALGORITHM 5.8 *Paull*

Step 1 Find the switch number not present in column j; denote it by A.

Step 2 Find the switch number not present in row i; denote it by B.

Step 3 Set up (I_i, O_j) through M_A, i.e., $\mathbf{M}[i; j] := A$; $i_1 := i$; $j_1 := j$.

Step 4 In row i_1 find such column j' that $\mathbf{M}[i_1; j'] = A$; if there is such j', then replace $\mathbf{M}[i_1; j'] = A$ with $\mathbf{M}[i_1; j'] = B$, set $j_1 := j'$ and go to step 5, otherwise the call is unblocked and the algorithm ends.

Step 5 In column j_1 find such row i' that $\mathbf{M}[i'; j_1] = B$; if there is such i', then replace $\mathbf{M}[i'; j_1] = B$ with $\mathbf{M}[i'; j_1] = A$, set $i_1 := i'$ and go to step 4, otherwise the call is unblocked and the algorithm ends.

First we have to determine which switches will be used for rearrangements. We can choose only center stage switches which have the free link to switch O_j or from switch I_i. There is no center stage switch with free links to both I_i and O_j, since the connection (I_i, O_j) is blocked. In the first step we find switch M_A which has the free link to switch O_j. Switch M_B found in the second step has the free link from switch I_i. We have chosen two switches, M_A and M_B, which will be used to rearrange the state of the switch fabric. Now we take randomly one of these switches to set up the new connection, say this is the center stage switch M_A (step 3). The example of Paull's matrix is given in Fig. 5.12a. Connection (I_i, O_j) is blocked (it is assumed that $i = 1$ and $j = 1$) and connections set up through switches M_A and M_B are shown. Connection (I_i, O_j) is set up through switch M_A ($\mathbf{M}[1; 1] := A$ in Fig. 5.12b). Switch M_A is the switch which was not present in j, but it was already present in i. So we have now A present two times in i. Variables i_1 and o_1 represent the currently considered row and column, respectively. At the beginning these two variables have values i and j. In step 4 we look for such column j' in row $i_1 = 1$ that $\mathbf{M}[1, j'] = A$. In the matrix of Fig. 5.12a, A is in column 4 of row $i_1 = 1$, so we replace $\mathbf{M}[1, 4] = A$ with $\mathbf{M}[1, 4] = B$ and set $j_1 = j' = 4$ (i.e., move $(I_{i_1}, O_{j'})$ from M_A to M_B, and set j' as the currently considered column). After this replacement we have to check if B is present in another row of column 4, so we go to step 5. In the matrix of Fig. 5.12a, switch B is in row $i' = 2$ of column $j_1 = 4$. So

the new value of i_1 is now 2, we replace $\mathbf{M}[2,4] = B$ with $\mathbf{M}[2,4] = A$ and we have to check if there is another column in row 2 containing A. Therefore we go back to step 4 (remember that after update we have now $i_1 = 2$ and $j_1 = 3$). The chain of rearrangements in the matrix is:
$\mathbf{M}[1;1] := A$, $i_1 := 1$, $j_1 := 1$;
$j' = 4$, $\mathbf{M}[1,4] := B$, $j_1 := 4$;
$i' = 2$, $\mathbf{M}[2,4] := A$, $i_1 := 2$;
$j' = 2$, $\mathbf{M}[2,2] := B$, $j_1 := 2$;
$i' = 8$, $\mathbf{M}[8,2] := A$, $i_1 := 8$;
$j' = 5$, $\mathbf{M}[8,5] := B$, $j_1 := 5$;
$i' = 4$, $\mathbf{M}[4,5] := A$, $i_1 := 4$;
$j' = 7$, $\mathbf{M}[4,7] := B$, $j_1 := 7$;
$i' = 7$, $\mathbf{M}[7,7] := A$, $i_1 := 7$;
$j' = 6$, $\mathbf{M}[7,6] := B$, $j_1 := 6$;
$i' = 6$, $\mathbf{M}[6,6] := A$, $i_1 := 6$;
$j' = 8$, $\mathbf{M}[6,8] := B$, $j_1 := 8$;
$i' = 5$, $\mathbf{M}[5,8] := A$, $i_1 := 5$;

In row $i_1 = 5$ switch M_A is not present so the algorithm ends. The state matrix after rearrangements is shown in Fig. 5.12b. The algorithm always ends with success, i.e., in step 4 or 5 there will be no considered switch in respective row or column. In the state matrix of Fig. 5.12a the algorithm ends on position $i_1 = 5$ and $j_1 = 8$. There is no another entry A in row 5. And when we look on the state matrix we can see that there is no such column where this switch could be. It cannot be in column $i = 1$, since in step 2 we chose A because it was not present in this column. It cannot also be in other columns (i.e., 2 to 8) since it is already present in other rows of these columns, and according to state matrix properties, each symbol can be present in each column and in each row only once. More formal proof can be found in [140].

In the algorithm *Paull* switches M_A and M_B were chosen for rearrangements randomly, and switch M_A was chosen for the new connection also randomly. Paull proposed different modifications, which resulted in a lower number of rearrangements needed to unblock the new call. For instance, we can set up the new call through switch M_B (not M_A) or we can check all possible switches and finally use switches, which results in the lowest number of rearrangements [140]. If in the considered matrix we will use switch M_B for the new connection, the chain of rearrangements is:
$\mathbf{M}[1;1] := B$, $i_1 := 1$, $j_1 := 1$;
$i' = 3$, $\mathbf{M}[3,1] := A$, $i_1 := 3$;
$j' = 3$, $\mathbf{M}[3,3] := B$, $j_1 := 3$;
so only 2 connections are to be rearranged instead of 12. It was proved

	1	2	3	4	5	6	7	8
1	*			A				
2		A		B				
3	B		A					
4					B		A	
5								B
6						B		A
7						A	B	
8		B			A			

Figure 5.12a. The Paull's matrix before rearrangements

	1	2	3	4	5	6	7	8
1	A			B				
2		B		A				
3	B		A					
4					A		B	
5								A
6						A		B
7						B	A	
8		A			B			

Figure 5.12b. The Paull's matrix after rearrangements

that using the *Paull* algorithm not more than $2r - 3$ rearrangements would be needed. When modified algorithms are used, the number of rearrangements needed is reduced to $r - 1$ [9].

	1	2	3	4
1	3,2			1
2			1,2	
3		3,4		
4		1		3,4

Figure 5.13. The state matrix of $C_{SD}(4,4,4,1)$ after rearrangements for connection $(1,4)$

In the switching matrix of Fig. 5.11b, connection (I_1, O_4) is blocked since in row 1 we have already numbers 1 and 2, while numbers 3 and 4 are in column 4. Numbers 1 and 2 are not in column 4, so we choose $A = 1$. In row 1 we have two numbers, 3 and 4, which are available; we choose any of them, let say $B = 3$. We now set up the new call through switch M_1 ($\mathbf{M}[1;4] := 1$ - step 3) and look for column j' containing 1 in row 1. We have $\mathbf{M}[1;1] = 1$, so $j' = 1$ and we replace $\mathbf{M}[1;1] = 1$ with $\mathbf{M}[1;1] = 3$ (step 4). In the next step we check if there is another row with 3 in column 1. Since there is no such row, the algorithm is ended and the new connection is established with one rearrangement. The matrix after this rearrangement is shown in Fig. 5.13, while the state of the switching fabric is shown in Fig. 5.14. The new connection $\langle 3, 12\rangle$ (i.e., (I_1, O_4)) is shown in dashed line and the rearranged connection $\langle 0, 0\rangle$ is shown in dotted lines.

The *Paull* algorithm can be also used in switching fabrics composed of more than three stages, obtained by recursively replacing center stage switches with other three-stage fabrics. The example of a five-stage 27×27 switch fabric, obtained from $C_{SD}(3,9,3,1)$ by replacing the center stage 9×9 switches with $C_{SD}(3,3,3,1)$, is shown in Fig. 5.15. Connections are shown in bold lines. Paull's matrices for this switching fabric are given in Fig. 5.16a. Entries in matrix $\mathbf{M}$ represent the three-stage switching fabric (1, 2 or 3) used for respective connections. Matrices $\mathbf{M}_1$, $\mathbf{M}_2$ and $\mathbf{M}_3$ correspond to states of three-stage switching fabrics 1, 2, and 3, respectively. In this switching fabric connection $\langle 25, 3\rangle$ (i.e., (I_9, O_1)) is blocked, since we cannot put any of numbers 1, 2 or 3 in position (I_9, O_1) of matrix $\mathbf{M}$. We use the *Paull* algorithm to rearrange the state of the switching fabric. Connection (I_9, O_1) is set

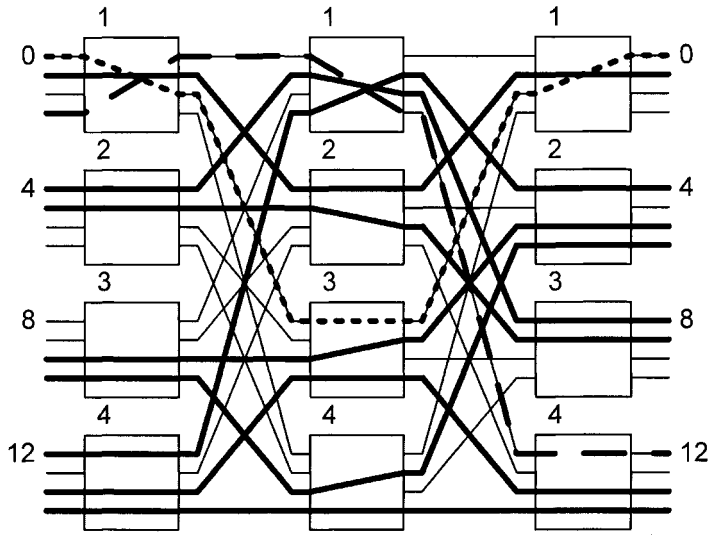


Figure 5.14. A state of $C_{SD}(4,4,4,1)$ after rearrangements

up through the center three-stage switching fabric M_3 and connection (I_9, O_9) is moved to the switching fabric M_2 (see matrix $\mathbf{M}$ after this rearrangement in Fig. 5.16b). Now we have to set up these two connections $((I_9, O_1)$ and $(I_9, O_9))$ in the center switching fabrics. Connection (I_9, O_1) is blocked in the center switching fabric M_3, while connection (I_9, O_9) is blocked in the center switching fabric M_2 (see $\mathbf{M}_2$ and $\mathbf{M}_3$ in Fig. 5.16a). We can now use the *Paull* algorithm for $\mathbf{M}_2$ and $\mathbf{M}_3$. All the matrices after rearrangements are shown in Fig. 5.16b and connections are shown in Fig. 5.17 (the new connection is shown in dashed line, and rearranged calls are shown in doted lines).

5.1.6.2 Simultaneous Connections

In simultaneous connections we have a set of compatible connections and we have to choose the center stage switch for each call. Several approaches and algorithms were proposed to do this simultaneously.

Matrix Decompositions. In algorithms based on matrix decomposition a maximal assignment is represented in the matrix $\mathbf{H}_r^{(n)}$ with r rows and r columns. Rows and columns correspond to the first and the third stage switches, respectively, and the matrix is defined as follows:

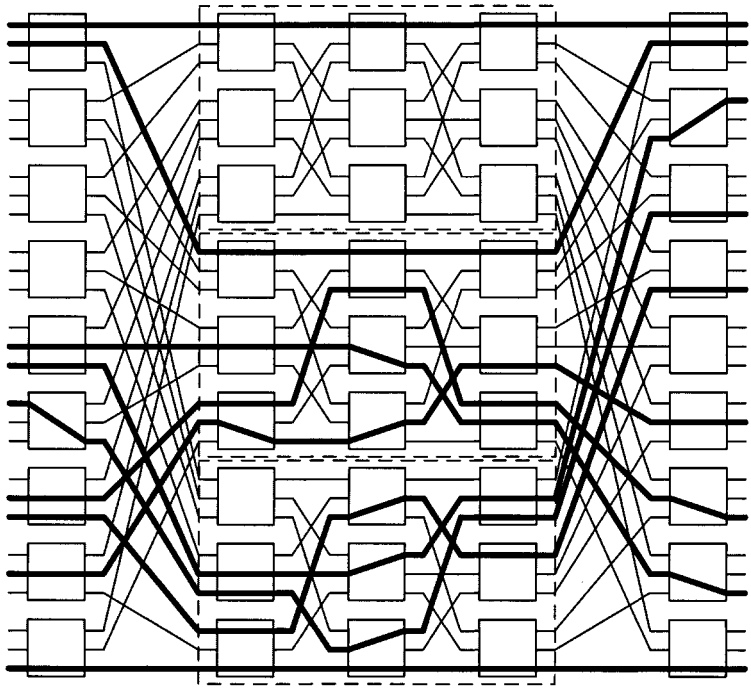


Figure 5.15. The state of the five-stage 27 × 27 switch fabric before rearrangements

DEFINITION 5.22

$$\mathbf{H}_r^{(n)} = \left[h_{i,j} = \begin{cases} k, & \text{if there are } k \text{ connections } (I_i, O_j); \\ 0, & \text{if there is no such connection.} \end{cases} \right] \tag{5.35}$$

An entry $h_{i,j} = k$ means that the number of connections between switches I_i and O_j is equal to k, i.e., that a maximal assignment contains k connections (I_i, O_j). When $k = 0$, there is no such connection. Since each first stage switch has n inputs, we have n connections from this switch in a maximal assignment. Therefore, when we add all elements in any row, we obtain n. The same is true for any column, since each third stage switch has n outputs. These properties of matrix $\mathbf{H}_r^{(n)}$ can be denoted by equations:

$$\sum_{i=1}^{r} \mathbf{H}[i;j] = n \quad \text{and} \quad \sum_{j=1}^{r} \mathbf{H}[i;j] = n. \tag{5.36}$$

Consider now a matrix for which the sum of any row or column is unity. It is often called an elementary permutation matrix, and is de-

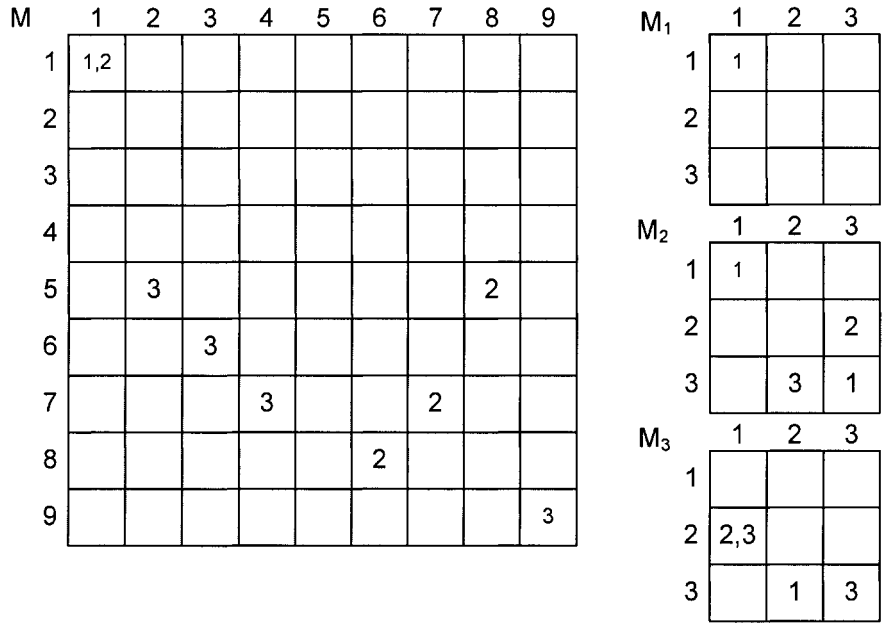


Figure 5.16a. Paull's matrices before rearrangements

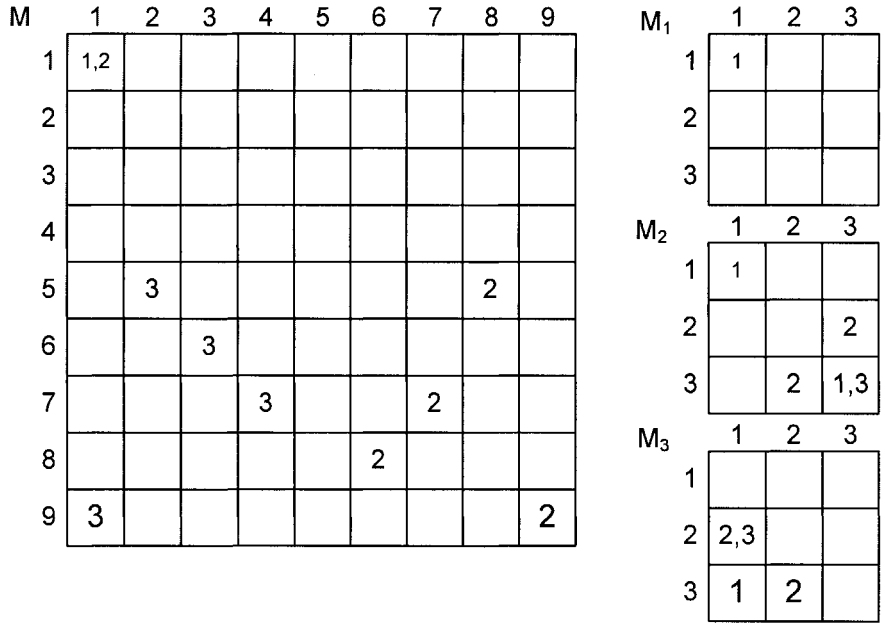


Figure 5.16b. Paull's matrices after rearrangements

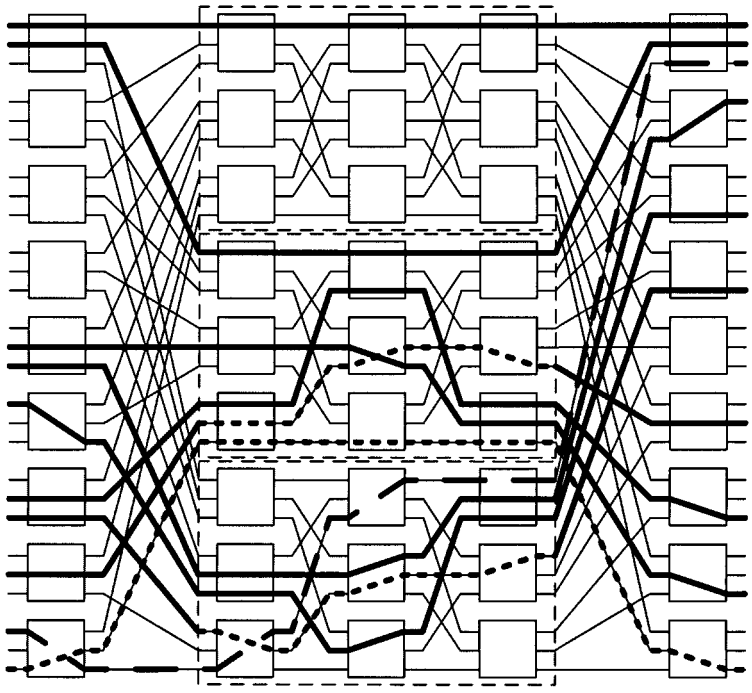


Figure 5.17. The state of the five-stage 27 × 27 switch fabric after rearrangements

noted by $\mathbf{E}_{r\times r}$. Neiman [118] has shown that the control of the rearrangeable switch fabric can be interpreted as a procedure of finding a set of $\mathbf{E}_{r\times r}$ matrices which can be subtracted, one at a time, from some given $\mathbf{H}_r^{(n)}$, and connections corresponding to each of these $\mathbf{E}_{r\times r}$ matrices can be set up through one center stage switch. We say also that matrix $\mathbf{H}_r^{(n)}$ is decomposed into n matrices $\mathbf{E}_{r\times r}$. Let us consider $C_{SD}(4,4,4,1)$ and the permutation

$$\Pi = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 1 & 15 & 8 & 12 & 13 & 3 & 9 & 2 & 11 & 6 & 14 & 5 & 10 & 4 & 0 & 7 \end{pmatrix} \tag{5.37}$$

We have one connection (I_1, O_1) (i.e., connection $\langle 0,1\rangle$), one connection (I_1, O_3) ($\langle 2,8\rangle$), and two connections (I_1, O_4) ($\langle 1,15\rangle$ and $\langle 3,12\rangle$). Therefore, $h_{1,1} = h_{1,3} = 1$ and $h_{1,4} = 2$. There is no connection between switches I_1 and O_2, so $h_{1,2} = 0$. Matrix $\mathbf{H}_r^{(n)}$ is shown in Fig. 5.18a. This matrix can be decomposed to four elementary permutations matrices $\mathbf{E}_1$, $\mathbf{E}_2$, $\mathbf{E}_3$, and $\mathbf{E}_4$. These matrices are shown in Fig. 5.18b, while connecting paths in $C_{SD}(4,4,4)$ are shown in Fig. 5.19. Several algo-

H	1	2	3	4
1	1	0	1	2
2	2	0	1	1
3	0	2	1	1
4	1	2	1	0

Figure 5.18a. Simultaneous connections in $C_{SD}(4,4,4,1)$ - matrix $\mathbf{H}$

$\mathbf{E}_1$	1	2	3	4
1	1	0	0	0
2	0	0	0	1
3	0	1	0	0
4	0	0	1	0

$\mathbf{E}_2$	1	2	3	4
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
4	1	0	0	0

$\mathbf{E}_3$	1	2	3	4
1	0	0	1	0
2	1	0	0	0
3	0	0	0	1
4	0	1	0	0

$\mathbf{E}_4$	1	2	3	4
1	0	0	0	1
2	1	0	0	0
3	0	0	1	0
4	0	1	0	0

Figure 5.18b. Simultaneous connections in $C_{SD}(4,4,4,1)$, - matrices $\mathbf{E}$

rithms have been proposed for decomposing matrix $\mathbf{H}_r^{(n)}$ into matrices $\mathbf{E}_{r\times r}$. The main drawbacks are time complexity and number of iterations. Neiman proposed an algorithm with the time complexity $O(r^2m^2)$ [118]. Some modifications in this algorithm, which resulted in fewer iterations, were proposed in [162], [60], [12]. Another algorithm with time complexity $O(nr^2)$ was proposed in [96]. Parallel algorithms as well as algorithms which realize only some of all possible permutations were also considered [54]. More efficient algorithms use graph coloring.

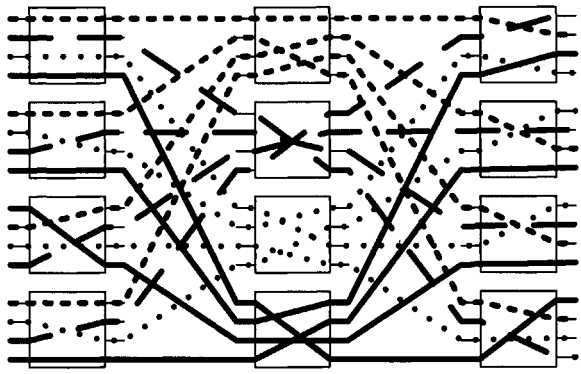


Figure 5.19. Simultaneous connections in $C_{SD}(4,4,4,1)$ - a state of the switch fabric

Matching and Graph Coloring Algorithms. Algorithms for finding perfect matchings in bipartite multigraph and graph coloring can be used to route connections in the switching fabric. In this approach a maximal assignment is represented by a bipartite multigraph $\mathcal{G}(\mathbb{V},\mathbb{E})$. Such a graph representing a maximal assignment in $C_{SD}(n,r,m,1)$ has $2r$ nodes (r nodes in each set), $N = nr$ edges, degree n, and is n-colorable, or we can find n perfect matchings. Connections corresponding to one perfect matching, or corresponding to edges colored with the same color, can be realized using one center stage switch. A bipartite multigraph $\mathcal{G}$ for connections considered in Fig. 5.19 is shown in Fig. 5.20. In switch I_1 we have one connection to switch O_1, one connection to switch O_3, and two connections to switch O_4. Therefore, node 1 on the left hand side is joined by one edge to node 1 and to node 3, and by two edges to node 4 on the right hand side. The graph $\mathcal{G}$ has $2r = 8$ nodes and $N = 16$ edges. Its degree is equal to 4 and it is 4-regular since each node in one set is joined to nodes in another set by exactly four edges. Perfect matchings in $\mathcal{G}$ and $\mathcal{G}$ colored with three colors are shown in Figs. 5.21 and 5.22, respectively. It is clear that calls corresponding to edges in each perfect matching or colored with the same color can be set up through one center stage switch.

An efficient algorithm for finding a perfect matching in bipartite graphs was proposed in [4]. The algorithm use the property saying that any $2k$-regular multigraph $\mathcal{H}$ can be split into two k-regular spanning subgraphs $\mathcal{H}_1$ and $\mathcal{H}_2$. A perfect matching is a 1-regular spanning subgraph. To obtain a perfect matching of graph $\mathcal{H}$ by splitting it t times, it should be 2^t-regular. Therefore, the bipartite multigraph representing maximal assignment must be first converted to a 2^t-regular bipartite multigraph.

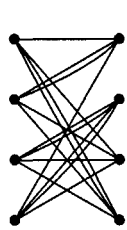


Figure 5.20. The bipartite multigraph representations of connections for $C_{SD}(4,4,4,1)$

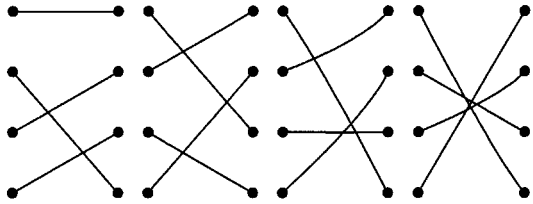


Figure 5.21. Perfect matchings in the bipartite multigraph of Fig. 5.20

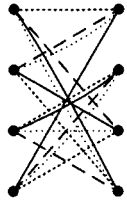


Figure 5.22. Coloring of the bipartite multigraph of Fig. 5.20

This conversion is done by replacing each edge in $\mathcal{G}$ with $\alpha = \lfloor 2^t/n \rfloor$ multiple edges, and by adding $\beta = 2^t - n\alpha$ copies of edges in an arbitrary perfect matching $\mathcal{P}$, where t is the nearest integer satisfying $2^t \geqslant N(= nr)$, i.e., $t = \lceil log_2(rn) \rceil$. The edges of $\mathcal{P}$ and their copies are called bad edges. The algorithm is as follows.

ALGORITHM 5.9 *Perfect Matching*

Input: Maximal assignment Π; $\mathcal{G}$; n; r;
Output: Perfect matching $\mathcal{H}^{t+1}$;

Step 1 Calculate parameters: $t = \lceil log_2(2rn/2) \rceil$, $\alpha = \lfloor 2^t/n \rfloor$, and $\beta = 2^t - n\alpha$.

Step 2 Create an arbitrary perfect matching $\mathcal{P}$ in $\mathcal{G}$. This $\mathcal{P}$ does not necessarily consist of edges of $\mathcal{G}$.

Step 3 Create the graph $\mathcal{H}_1$ by replacing each edge in $\mathcal{G}$ with α multiple edges and by adding β copies of each edge in $\mathcal{P}$ (called bad edges). Graph $\mathcal{H}_1$ is 2^t-regular.

Step 4 Starting from $s = 1$ to t do: split graph $\mathcal{H}^s$ into $\mathcal{H}_1^{s+1}$ and $\mathcal{H}_2^{s+1}$ edge disjoint 2^{t-s}-regular spanning subgraphs in the following way:

- put $\lfloor \alpha/2^s \rfloor$ copies of each edge to $\mathcal{H}_1^{s+1}$, and $\lfloor \alpha/2^s \rfloor$ copies to $\mathcal{H}_2^{s+1}$; remove these edges from $\mathcal{H}^s$;
- put $\lfloor \beta/2^s \rfloor$ copies of each bad edge to $\mathcal{H}_1^{s+1}$, and $\lfloor \beta/2^s \rfloor$ copies to $\mathcal{H}_2^{s+1}$; remove these edges from $\mathcal{H}^s$;
- find an Euler cycle in each connected component of the remaining subgraph of $\mathcal{H}^s$, move odd numbered edges to $\mathcal{H}_1^{s+1}$ and even numbered edges to $\mathcal{H}_2^{s+1}$. Choose a subgraph with the lower number of bad edges as the graph $\mathcal{H}^{s+1}$.

Step 5 Graph $\mathcal{H}^{t+1}$ is a perfect matching.

The total running time of this algorithm is $O(Nt) = O(N \log N)$. The use of the algorithm is presented in the following example. Let us consider the graph presented in Fig. 5.23a. In step 1 the parameters of the algorithm are calculated. Parameter t is equal to 4 (ceiling of $\log_2 16$), and it means that the algorithm will perform step 4 four times. In steps 2 and 3 two additional graphs are created. Graph $\mathcal{P}$ is presented in Fig. 5.23b, while $\mathcal{H}^1$, created from $\mathcal{G}$ and $\mathcal{P}$, is shown in Fig. 5.23c. Since $\beta = 0$ graph $\mathcal{H}^1$ do not contain any copies of edges of $\mathcal{P}$. Each edge in $\mathcal{H}^1$ is multiplied $\alpha = 4$ times. So the graph is still a regular graph but of degree $2^t = 16$. The first execution of step 4 results in two subgraphs presented in Fig. 5.23d. They were obtained by placing $\lfloor \alpha/2 \rfloor = 2$ copies of each edge in $\mathcal{H}_1^1$ and $\mathcal{H}_2^1$. Since all edges are placed in new graph, there is no need to look for an Euler cycle in this step. Both graphs are identical, so any of them can be taken as $\mathcal{H}^2$. Graph $\mathcal{H}^2$ is again divided into $\mathcal{H}_1^2$ and $\mathcal{H}_2^2$ shown in Fig. 5.23e, and since both are identical, any of them can be taken as $\mathcal{H}^3$. At this step $\lfloor \alpha/2^s \rfloor = 0$ so we do not move any edges graphs $\mathcal{H}_1^3$ and $\mathcal{H}_2^3$ and in the remaining graph which is graph $\mathcal{H}^3$ in this case, we have to find an Euler cycle (see for instance [146]). This cycle can be for instance: $1 \to 5 \to 3 \to 7 \to 3 \to 6 \to 2 \to 5 \to 1 \to 6 \to 4 \to 7 \to 4 \to 8 \to 2 \to 8 \to 1$, where nodes on the left side of the graph are numbered 1, 2, 3, 4 from top to bottom, and nodes on the right side are numbered 5, 6, 7, 8 from bottom to top. Graphs $\mathcal{H}_1^3$ and $\mathcal{H}_2^3$ are shown in Fig. 5.23f. Graph $\mathcal{H}_1^3$ was taken as

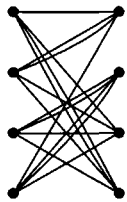


Figure 5.23a. Input graph $\mathcal{G}$

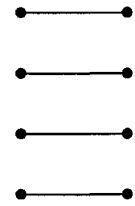


Figure 5.23b. Arbitrary perfect matching $\mathcal{P}$

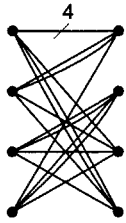


Figure 5.23c. Graph $\mathcal{H}_1$

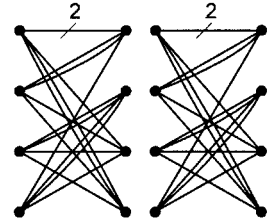


Figure 5.23d. Graphs $\mathcal{H}_1^1$ and $\mathcal{H}_2^1$

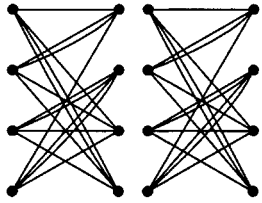


Figure 5.23e. Graphs $\mathcal{H}_1^2$ and $\mathcal{H}_2^2$

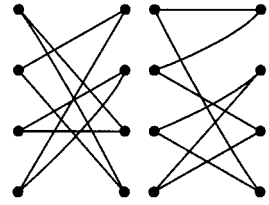


Figure 5.23f. Graphs $\mathcal{H}_1^3$ and $\mathcal{H}_2^3$

graph $\mathcal{H}^4$. This graph was divided into $\mathcal{H}_1^4$ and $\mathcal{H}_2^4$ (see Fig. 5.23g, and an Euler cycle was $1 \rightarrow 5 \rightarrow 2 \rightarrow 8 \rightarrow 4 \rightarrow 7 \rightarrow 3 \rightarrow 6 \rightarrow 1$. Both graphs are perfect matchings and any of them can be taken as final graph, so finally $\mathcal{H}^{t+1} = \mathcal{H}^5$ is returned by the algorithm as its result.

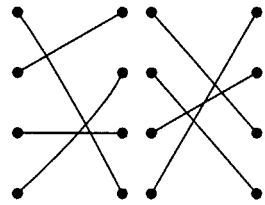


Figure 5.23g. Graphs $\mathcal{H}_1^4$ and $\mathcal{H}_2^4$

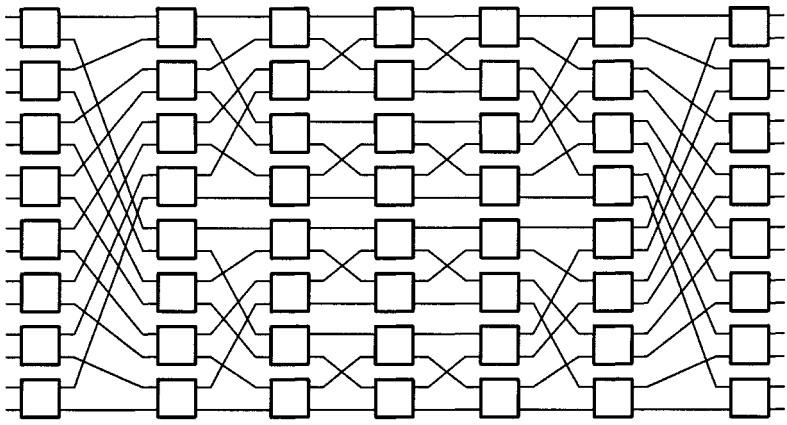


Figure 5.24. The 16 × 16 Beneš switching fabric

Connection routing in three-stage switching fabrics may be also modeled as a network-flow problem. This approach for the space-division switching fabric with multicast connections was considered by Varma and Chalasani [167].

The Looping Algorithm. The looping algorithm was originally proposed to route simultaneous connections in switching fabrics composed of 2×2 switches [132]. It was later extended to switching fabrics with $2^t \times 2^t$ switches [5]. When the switching fabric is composed of 2×2 switches, two inputs (outputs) of the same first (last) stage switch are called dual. The dual of input x (output y) is denoted by $\sim x$ ($\sim y$). The $N \times N$ Beneš switching fabric contains $2n - 1$ stages of 2×2 switches, where $n = \log_2 N$. The 16×16 switching fabric is shown in Fig. 5.24. It can be also constructed recursively by using two $N/2 \times N/2$ switching fabrics (called the upper and the lower switching fabric, respectively), and adding outer stages, each of $N/2$ switches, as it is shown in Fig. 5.25.

Let $x_{n-1}x_{n-2}\ldots x_0$ be a binary representation of an input terminal x, The $\sim x$ is represented in binary form by $x_{n-1}x_{n-2}\ldots\overline{x_0}$, where $\overline{x_0}$ is complementation of x_0. For instance 0 and 1 or 2 and 3 are duals. We use two tables: $\mathbf{MI}[N;2]$ and $\mathbf{MO}[N;2]$. An entry $\mathbf{MI}[x;1] = y$ denotes that input terminal x is to be connected to output terminal y. Analogically, $\mathbf{MO}[y;1] = x$. The second column of these matrices contains 0s or 1s, where $\mathbf{MI}[x;2] = \mathbf{MO}[y;2] = 0$ means that connection $\langle x;y\rangle$ is to be set up through the upper switching fabric, and $\mathbf{MI}[x;2] = \mathbf{MO}[y;2] = 1$ means that the lower switching fabric is used. Another entry, for instance

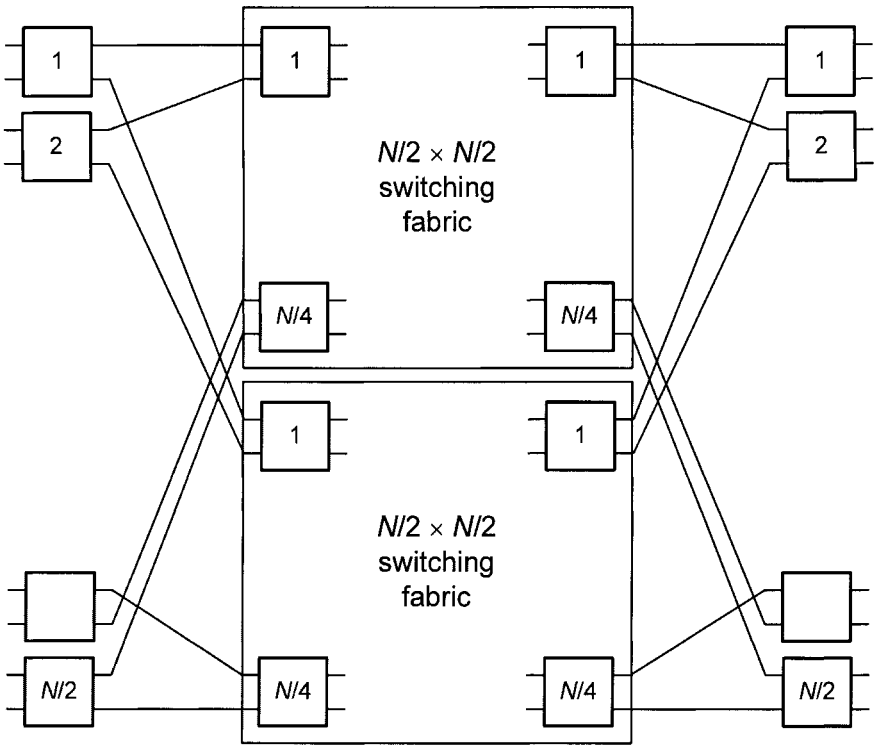


Figure 5.25. Recursive construction of the Beneš switching fabric

$\mathbf{MI}[x;2] = \mathbf{MO}[y;2] = 2$, denotes that this connection is not set up yet. The algorithm is as follows:

ALGORITHM 5.10 *Looping*

Input: Maximal assignment Π;
Output: Matrices $\mathbf{MI}[N;2]$ and $\mathbf{MO}[N;2]$;

Step 1 Set matrices **MI** and **MO**: for $x = 0$ to $N-1$ (i.e., for all input and output terminals) $\mathbf{MI}[x;1] := \pi(x)$ (input terminal x is to be connected to $\pi(x)$); $\mathbf{MI}[x;2] := 2$ (input terminal x is not connected yet); $\mathbf{MO}[x;1] := \pi^{-1}(x)$ (output terminal x is to be connected to $\pi^{-1}(x)$); $\mathbf{MO}[x;2] := 2$ (output terminal x is not connected yet).

Step 2 Find a not connected input terminal x and set $x_0 := x$. If all input terminals are connected the algorithm is ended.

Step 3 Connect x_0 to $\pi(x_0)$ through the upper switch fabric: $\mathbf{MI}[x_0;2] := 0$; $\mathbf{MO}[\pi(x_0);2] := 0$.

Step 4 Find $\sim \pi(x_0)$.

Step 5 Find an input terminal x_1, $\sim \pi(x_0)$ is to be connected to: $x_1 := \pi^{-1}(\sim \pi(x_0))$.

Step 6 Connect x_1 to $\sim \pi(x_0)$ through the lower switch fabric: $\mathbf{MO}[\sim \pi(x_0); 2] := 1$; $\mathbf{MI}[x_1; 2] := 1$;

Step 7 Find $\sim x_1$; if connected go to step 2; otherwise, set $x_0 = \sim x_1$ and go to step 3.

Let's assume that permutation (5.37) is to be realized in 16×16 Beneš switching fabric. Matrices **MI** and **MO** obtained in step 1 of the looping algorithm are shown in Fig. 5.26a. In step 2 input terminal 0 is chosen as the first not connected input terminal. It is connected with output terminal 1 through the upper subnetwork ($\mathbf{MI}[0; 2] = 0$). Since $\pi(0) = 1$ than $\mathbf{MO}[1; 2]$ is also set to 0. The dual of 1 is 0 and output terminal 0 is to be connected with input terminal 14, since $\pi^{-1}(0) = 14$. Connection $\langle 14, 0 \rangle$ is set up through lower subnetwork - $\mathbf{MI}[14; 2] = \mathbf{MO}[0; 2] = 1$. The rest run of this loop is shown in Fig. 5.26b. It ends on input terminal 1. The dual of this input terminal is input terminal 0 which is already connected. Therefore, we start the next loop from the first not connected input terminal, i.e., from input terminal 2. The run of this loop is shown in Fig. 5.26c. This loop ends at input terminal 3 (the dual for input terminal 2) and after this loop all input terminals are connected. So the looping algorithm ends. After this run of the algorithm all outer stage switches are set up and it is determined which connections are to be set up through upper and lower subnetworks. This set up is shown in Fig. 5.26d. Now the looping algorithm is again used to set up switches in the upper and the lower switch fabrics in the similar way.

5.1.7 Repackable Switching Fabrics

Call repacking enables to increase the loading of the most used center stage switches, leaving the less loaded switches free to carry new connections that would otherwise have been blocked. The idea of call repacking as well as first call repacking algorithms have been proposed by Ackroyd [1]. In repackable switching fabrics, in contrast to rearrangeable ones, rearrangement of existing connections is realized each time a connection is terminated. Ackroyd has shown that using connection repacking the traffic performance of a switching fabric is improved. He proposed the following algorithms:

MI	1	2
0	1	2
1	15	2
2	8	2
3	12	2
4	13	2
5	3	2
6	9	2
7	2	2
8	11	2
9	6	2
10	14	2
11	5	2
12	10	2
13	4	2
14	0	2
15	7	2

2	1	MO
2	14	0
2	0	1
2	7	2
2	5	3
2	13	4
2	11	5
2	9	6
2	15	7
2	2	8
2	6	9
2	12	10
2	8	11
2	3	12
2	4	13
2	10	14
2	1	15

Figure 5.26a. Matrices **MI** and **MO** for permutation (5.37)

MI	1	2
0	1	0
1	15	1
2	8	2
3	12	2
4	13	2
5	3	2
6	9	2
7	2	2
8	11	0
9	6	1
10	14	0
11	5	1
12	10	1
13	4	0
14	0	1
15	7	0

2	1	MO
1	14	0
0	0	1
2	7	2
2	5	3
0	13	4
1	11	5
1	9	6
0	15	7
2	2	8
2	6	9
1	12	10
0	8	11
2	3	12
2	4	13
0	10	14
1	1	15

Figure 5.26b. Matrices **MI** and **MO** after the first loop

MI	1	2
0	1	0
1	15	1
2	8	0
3	12	1
4	13	0
5	3	1
6	9	1
7	2	0
8	11	0
9	6	1
10	14	0
11	5	1
12	10	1
13	4	0
14	0	1
15	7	0

2	1	MO
1	14	0
0	0	1
0	7	2
1	5	3
0	13	4
1	11	5
1	9	6
0	15	7
0	2	8
1	6	9
1	12	10
0	8	11
1	3	12
0	4	13
0	10	14
1	1	15

Figure 5.26c. Matrices **MI** and **MO** after the second loop

ALGORITHM 5.11 *Acroyd-single*

When the new connection is to be set up use Minimum Index algorithm; When connection (I_i, O_j) realized through switch M_k, $1 \leqslant k < m$ is terminated than:

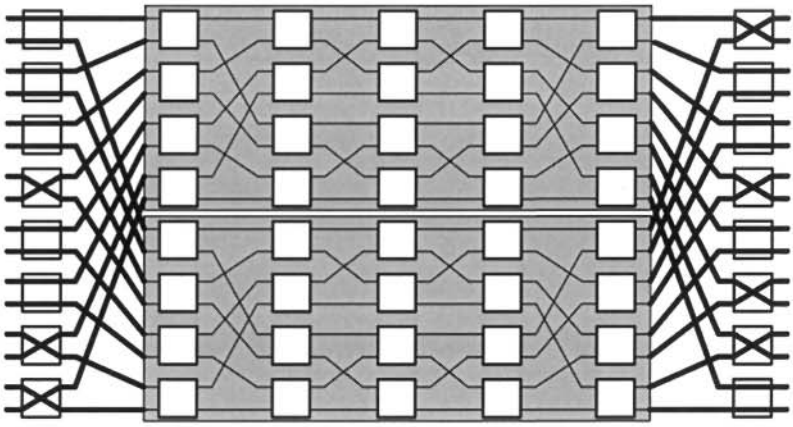


Figure 5.26d. Connections in 16×16 Beneš switching fabric after first run of looping algorithm

Step 1 Search in turn, starting from switch M_m to switch M_{k+1}, for a connection that can be moved to switch M_k.

Step 2 If such connection exists, say in switch M_{k_1}, move it to switch M_k.

ALGORITHM 5.12 *Acroyd-multiple*

If, after using Acroyd-single algorithm, call repacking was realized from switch M_{k_1}, $k_1 < m$, set $k = k_1$ and repeat algorithm Acroyd-single.

Jajszczyk and Jekel gave the required number of middle stage switches to obtain nonblocking operation of the $C_{SD}(n, r, m, 1)$ [67]. They proved the following theorem:

THEOREM 5.23 *When $C_{SD}(n, r, m, 1)$ is controlled in such a way that each overweight state is nonpermanent, then it is nonblocking if and only if*

$$m \geqslant 2n - \left\lceil \frac{n}{r-1} \right\rceil, \tag{5.38}$$

where a nonpermanent state is a state which is immediately replaced by another state, and an overweight state is a state in which there exists a connecting path which can be moved to another more heavily loaded center stage switch.

Proof. Let the new connection be (I_i, O_j); let $\mathbb{M}_1$ be the set of center stage switches through which connections (I_x, O_j) are realized, $1 \leqslant x \leqslant r$, $x \neq i$; let $\mathbb{M}_2$ be the set of center stage switches through which connections (I_i, O_y),$1 \leqslant y \leqslant r$; and let $\mathbb{M}_1 \cap \mathbb{M}_2 = \emptyset$. Without lost of generality it may be assumed that the new connection (I_i, O_j) is set up through switch in $\mathbb{M}_2$, and $\mathbb{M}_2$ contains switches with higher indexes while $\mathbb{M}_1$ contains switches with lower indexes. Since the switching fabric is repackable, the obtained state is not an overweight one, i.e., it is not possible to re-switch any connection from any switch in $\mathbb{M}_2$ to any switch in $\mathbb{M}_1$. We have $|\mathbb{M}_1| \leqslant n$ and $|\mathbb{M}_2| \leqslant n$, since switches I_i and O_j can have at most n connections each. We have also $|\mathbb{M}_2| \leqslant a \max\limits_{1 \leqslant e \leqslant a} |(I_i, O_y)|$, where $|(I_i, O_y)|$ denotes the number of connections (I_i, O_y) and e is the number of different O_y. So $|\mathbb{M}_2| \leqslant \min\left\{n; a \max\limits_{1 \leqslant y \leqslant a} |(I_i, O_y)|\right\}$. Since it is not possible to move any of the connections from $\mathbb{M}_2$ to $\mathbb{M}_1$, there are some connections (I_z, O_y), $z \neq i$ realized through switches in $\mathbb{M}_1$. It means that in switch O_y we have $\max\limits_{1 \leqslant e \leqslant a} |(I_i, O_y)|$ connections from switches in $\mathbb{M}_2$, so not more than $n - \max\limits_{1 \leqslant e \leqslant a} |(I_i, O_y)|$ connections may come from switches in $\mathbb{M}_1$, i.e., $|\mathbb{M}_1| \leqslant n - \max\limits_{1 \leqslant e \leqslant a} |(I_i, O_y)|$. The total number of switches required is

$$m \leqslant n - \max_{1 \leqslant e \leqslant a} |(I_i, O_y)| + \min\left\{n; a \max_{1 \leqslant e \leqslant a} |(I_i, O_Y)|\right\}. \tag{5.39}$$

We have to maximize this number through all $\max\limits_{1 \leqslant e \leqslant i} |(I_i, O_y)|$, and a, and this maximum is obtained for (see [67] for details)

$$m = 2n - \left\lceil \frac{n}{r-1} \right\rceil.$$

The sequence of events which results in the occupancy of this number of center stage switches is also given in the cited paper. □

For $r > n$ we have $m \geqslant 2n-1$, i.e., the required number of center stage switches is the same as for strict sense nonblocking switching fabrics.

5.2 Time-division Switching

5.2.1 Switching Fabrics Composed of Digital Switches

Three-stage, time-division switching networks, composed of digital matrices, were proposed by Charransol et al. [19]. A structure of asym-

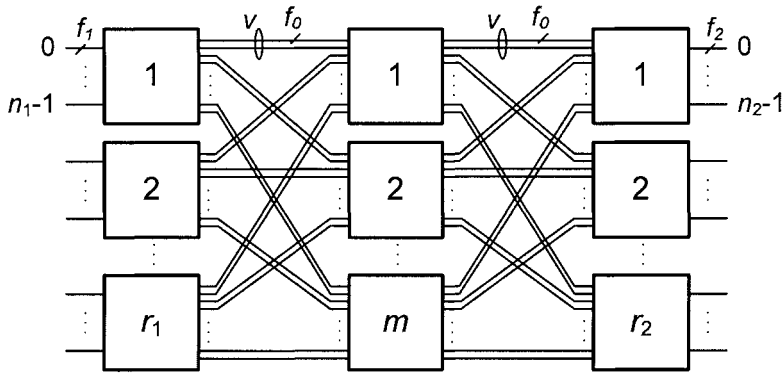


Figure 5.27. A time-division two-sided v-dilated three-stage switching fabric

metrical time-division two-sided v-dilated three-stage switching fabric is shown in Fig. 5.27. It contains r_1 switches in the first stage, m switches in the second stage, and r_2 switches in the third stage. Switches in adjacent stages are connected between themselves by means of v links. Each input stage switch has n_1 input links and vm output links, while each output stage switch has mv input links and n_2 output links. Center stage switches has a capacity of $r_1 v \times r_2 v$ links. Input links in first stage switches and output links in third stage switches carry f_1 time slots each, while each interstage links carries f_0 time slots. This switching fabric will be denoted by $C_{TD}(n_1, r_1, f_1, m, n_2, r_2, f_2, v, f_0)$. When $f_1 = f_2 = f$, $n_1 = n_2 = n$ and $r_1 = r_2 = r$ the switching fabric is symmetrical and will be denoted by $C_{TD}(n, r, f, m, v, f_0)$. Combinatorial properties of such switching fabrics were first considered by Jajszczyk [62] in the case of unicast connections. Several other papers deal with the performance evaluation of switching networks with multi-slot connections [68, 121, 152, 153].

5.2.2 Path Searching Algorithms

For 1-slot connections algorithms RAN (5.1), SEQ (5.2), MINIX (5.3), Q-RAN (5.4), CS (5.5), STU (5.6), and PACK (5.7) can be easily modified. Other algorithms which may be used to route connections are:

ALGORITHM 5.13 *Link packing (PACK-L)*

If the connection (I_i, O_j) is to be set up search for outgoing link from switch I_i according to the load of these links, starting from the most loaded link. If the accessible link lead to switch M_k and M_k has also accessible link to O_j, route the connection through M_k.

ALGORITHM 5.14 *Link unpacking (UNPACK-L)*

If the connection (I_i, O_j) is to be set up, search for outgoing link from switch I_i according to the load of these links, starting from the least loaded link. If the accessible link leads to switch M_k and M_k has also accessible link to O_j, route the connection through M_k.

In algorithm PACK-L, the most loaded link from switch I_i may not necessary lead to the most loaded center stage switch. Algorithm UNPACK-L results in that connections from switch I_i are more equally spread between all links from this switch. In the case of multi-slot connections, we should count the number of time slots occupied by connections instead of the number of connections to determine the load carried by switches or links when PACK, PACK-L, or UNPACK-L algorithm are to be used. In case of path searching for multi-slot connections in two-sided three-stage switching fabrics routing strategy based on functional division of center stage switches can be used, similarly as it was described in the case of one-sided two-stage switching fabrics composed of triangular switches.

5.2.3 Strict-sense Nonblocking Conditions

Nonblocking conditions for 1-slot connections will be considered first.

THEOREM 5.24 *$C_{TD}(n_1, r_1, f_1, m, n_2, r_2, f_2, v, f_0)$ is nonblocking in the strict sense for 1-slot connections if and only if:*

$$m \geqslant \min\left\{\left\lfloor\frac{n_1 f_1 - 1}{v f_0}\right\rfloor + \left\lfloor\frac{n_2 f_2 - 1}{v f_0}\right\rfloor + 1;\ \left\lfloor\frac{r_1 n_1 f_1 - 1}{v f_0}\right\rfloor + 1;\ \left\lfloor\frac{r_2 n_2 f_2 - 1}{v f_0}\right\rfloor + 1\right\}. \tag{5.40}$$

Proof. Sufficiency can be proved by showing the worst state in the switching fabric. In this state $n_1 f_1 - 1$ input channels of the first stage switch I_i may be connected to output channels of $r_2 - 1$ third stage switches (others than switch O_j), but no more than $(r_2 - 1)n_2 f_2$ such connections can be set up. These connections will occupy all channels of interstage links to $\lfloor \min\{n_1 f_1 - 1; (r_2 - 1)\, n_2 f_2\}/v f_0 \rfloor$ middle stage switches. Similarly, to the third stage switch O_j there may be $n_2 f_2 - 1$ connections from the first stages switches (others than switch I_i), but no more than $(r_1 - 1)n_1 f_1$ such connections can be set up. These connections may occupy another set of $\lfloor \min\{n_2 f_2 - 1; (r_1 - 1)\, n_1 f_1\}/v f_0 \rfloor$ middle stage switches. In the worst case these sets of switches are disjoint and

one more switch is needed to set up the connection (I_i, O_j). We have:

$$m \geqslant \left\lfloor \frac{\min\{n_1 f_1 - 1; (r_2 - 1) n_2 f_2\}}{v f_0} \right\rfloor + \\ + \left\lfloor \frac{\min\{n_2 f_2 - 1; (r_1 - 1) n_1 f_1\}}{v f_0} \right\rfloor + 1. \tag{5.41}$$

After similar considerations as in the proof of Theorem 5.1 we get inequality 5.40.

Necessity can be proved easily by presenting the switching fabric with lower number of middle stage switches, in which the new call is blocked. □

In the case of symmetrical switching fabric we obtain conditions given earlier in [62]:

COROLLARY 5.25 *$C_{TD}(n, r, f, m, v, f_0)$ is nonblocking in the strict sense for 1-slot connections if and only if:*

$$m \geqslant 2 \left\lfloor \frac{nf - 1}{v f_0} \right\rfloor + 1 \tag{5.42}$$

Proof. Set $f_1 = f_2 = f$, $n_1 = n_2 = n$, and $r_1 = r_2 = r$ in equation (5.41). Taking into account that $nf - 1 < rnf - nf$ for $r \geqslant 2$ we obtain (5.42). □

COROLLARY 5.26 *$C_{SD}(n, r, m, v)$ is nonblocking in the strict sense if and only if:*

$$m \geqslant 2 \left\lfloor \frac{n - 1}{v} \right\rfloor + 1 \tag{5.43}$$

Proof. Set $f = f_0 = 1$ in inequality 5.42. □

COROLLARY 5.27 *$C_{SD}(n_1, r_1, m, n_2, r_2, v)$ is strict-sense nonblocking if and only if:*

$$m \geqslant \min\left\{ \left\lfloor \frac{n_1 - 1}{v} \right\rfloor + \left\lfloor \frac{n_2 - 1}{v} \right\rfloor + 1; \right. \\ \left. \left\lfloor \frac{r_1 n_1 - 1}{v} \right\rfloor + 1; \left\lfloor \frac{r_2 n_2 - 1}{v} \right\rfloor + 1 \right\} \tag{5.44}$$

Proof. Set $f_1 = f_2 = f_0 = 1$ in inequality (5.40). □

Let us now consider the switching fabric presented in Fig. 5.27 when s-slot connections can be set up.

THEOREM 5.28 *$C_{TD}(n_1, r_1, f_1, m, n_2, r_2, f_2, v, f_0)$ is nonblocking in the strict sense for s-slot connections, $1 \leqslant s \leqslant s_{\max} \leqslant f_i$, if and only if:*

$$m \geqslant \max_{1 \leqslant s \leqslant s_{\max}} \left\{ \min \left\{ \left\lfloor \frac{n_1 f_1 - s}{v \left\lfloor \frac{f_0}{s} \right\rfloor} \right\rfloor + \left\lfloor \frac{n_2 f_2 - s}{v \left\lfloor \frac{f_0}{s} \right\rfloor} \right\rfloor + 1; \right. \right.$$
$$\left. \left. \left\lfloor \frac{r_1 n_1 f_1 - s}{v \left\lfloor \frac{f_0}{s} \right\rfloor} \right\rfloor + 1; \left\lfloor \frac{r_2 n_2 f_2 - s}{v \left\lfloor \frac{f_0}{s} \right\rfloor} \right\rfloor + 1 \right\} \right\} \tag{5.45}$$

Proof: Let us consider the "worst" state of the switching fabric presented in Fig. 5.27. Let us also assume, that the s-slot connection is to occupy s adjacent slots, $1 \leqslant s \leqslant s_{\max}$, i.e., the fixed or floating time-slot assignment is used. In the "worst" state $n_1 f_1 - s$ input channels of the first stage switch I_i and $n_2 f_2 - s$ output channels of the third stage switch O_j are occupied, and there is not any connection between these switches. However, no more than $(r_2 - 1) n_2 f_2$ time slots are available in the third stage switches others than O_j, and similarly, there are no more than $(r_1 - 1) n_1 f_1$ time slots available in the first stage switches others than I_i. The occupied time slots of these outer stage switches are assigned only to the 1-slot connections. The time slots are occupied in such way that there are exactly $s - 1$ free time slots between two successively occupied time slots, as shown in Fig. 2.17a. The link, whose time slots are occupied in the way described above, is inaccessible to the new s-slot connection if a fixed or floating time-slot assignment is used. In this link $\lfloor \min \{n_1 f_1 - s; (r_2 - 1) n_2 f_2\} / v \lfloor f_0 / s \rfloor \rfloor$ middle stage switches inaccessible by the new s-slot connection. Similarly, a set of $\lfloor \min \{n_2 f_2 - s; (r_1 - 1) n_1 f_1\} / v \lfloor f_0 / s \rfloor \rfloor$ middle stage switches may be inaccessible by the new s-slot connection. In the worst case these sets middle stage switches are disjoint. If the s-slot connection is to be realized between switches I_i and O_j an additional switch in the middle stage

is needed, so

$$m \geqslant \left\lceil \frac{\min\{n_1 f_1 - s;\,(r_2-1)\,n_2 f_2\}}{v\left\lfloor \frac{f_0}{s} \right\rfloor} \right\rceil + \left\lceil \frac{\min\{n_2 f_2 - s;\,(r_1-1)\,n_1 f_1\}}{v\left\lfloor \frac{f_0}{s} \right\rfloor} \right\rceil + 1. \tag{5.46}$$

When any of the values $(n_1 f_1 - s)/v\lfloor f_0/s \rfloor$, $(n_2 f_2 - s)/v\lfloor f_0/s \rfloor$, $[(r_2-1)\,n_2 f_2]/v\lfloor f_0/s \rfloor$, $[(r_1-1)\,n_1 f_1]/v\lfloor f_0/s \rfloor$ is not an integer the additional middle stage switch may be partially occupied, but it still will be accessible for the new s-slot connection. To realize any s-slot connection, the number of middle stage switches must be equal to the maximum value of m for $1 \leqslant s \leqslant s_{\max}$. □

Since both space-division and time-division switching are the special cases of the multi-slot switching, the theorems for the multi-slot switching should include these known results. For $s = 1$ (i.e., time-division switching), we obtain conditions given in Theorem 5.24. For $f_1 = f_2 = f_0 = v = s = 1$ (i.e., space-division switching), we obtain conditions given in Theorem 5.1.

Strict-sense nonblocking time-division switching fabrics with multicast connections where not considered in the literature. Some results can be obtained from conditions derived for switching fabrics with multirate discrete bandwidth connections, however, in this case it is assumed that flexible time-slots assignment is used for s-slot connections.

5.2.4 Wide-sense Nonblocking Conditions

In case of 1-slot connections, the only result for wide-sense nonblocking conditions is derived by Fishburn et al. [43] for switching fabrics with $r = 2$:

THEOREM 5.29 *$C_{TD}(n, 2, f, m, 1, f)$ is nonblocking in the wide sense for 1-slot connections under STU if*

$$m \geqslant \left\lceil \frac{3n}{2} \right\rceil. \tag{5.47}$$

They also proved the necessary conditions for $r = 3$. In other cases there is no algorithm currently known under which the switching fabric

will require less switches in the middle stage. In case of s-slot connections, the number of middle stage switches be reduced by using other time slot assignment algorithms. First the switching fabric with flexible assignment will be considered.

THEOREM 5.30 *$C_{TD}(n_1, r_1, f_1, m, n_2, r_2, f_2, v, f_0)$ is nonblocking in the wide sense for s-slot connections, $1 \leqslant s \leqslant s_{\max} \leqslant \min\{f_1; f_2\}$, under flexible assignment if and only if:*

$$m \geqslant \min\left\{\left\lfloor\frac{n_1 f_1 - s_{\max}}{v\,(f_0 - s_{\max} + 1)}\right\rfloor + \left\lfloor\frac{n_2 f_2 - s_{\max}}{v\,(f_0 - s_{\max} + 1)}\right\rfloor + 1;\right.$$
$$\left.\left\lfloor\frac{r_1 n_1 f_1 - s_{\max}}{v\,(f_0 - s_{\max} + 1)}\right\rfloor + 1;\left\lfloor\frac{r_2 n_2 f_2 - s_{\max}}{v\,(f_0 - s_{\max} + 1)}\right\rfloor + 1\right\}. \quad (5.48)$$

Proof. Proof is very similar to the proof of Theorem 5.28, but now $\lfloor[\min\{n_1 f_1 - s; (r_2 - 1)\, n_2 f_2\}]\,/v\,(f_0 - s_{\max} + 1)\rfloor$ middle stage switches are inaccessible by the new s-slot connection from the first stage switch, and $\lfloor[\min\{n_2 f_2 - s; (r_1 - 1)\, n_1 f_1\}]\,/v\,(f_0 - s_{\max} + 1)\rfloor$ switches are inaccessible from the third stage switch. This number must be maximized through all s, and it reaches maximum for $s = s_{\max}$. □

It should be noted that for $f_1 = f_2 = f_0 = v = s = 1$ or when only $s = 1$ we obtain the strict-sense nonblocking conditions for space-division switching (Theorem 5.1) or for time-division switching (Theorem 5.24), respectively. This is because we consider the special time slot assignment algorithm but we do not put any restriction on the way in which the second-stage switches are chosen. Since in both space-division switching and time-division switching, the time slot assignment is not considered, the result given in Theorem 5.30 should include these known results.

Let us now consider the switching fabric with $f = f_0 = f_1 = f_2 = s_{\max}$ in which the routing strategy with functional division is used. In this strategy, middle stage switches are divided into two groups. One group, containing m_1 switches, will be used for setting up s-slot connections, for which $s \leqslant \lfloor f/2 \rfloor$. Connections using $s > \lfloor f/2 \rfloor$ slots will be set up through one of switches in the other group, which contains m_2 switches.

THEOREM 5.31 *$C_{TD}(n_1, r_1, f, m, n_2, r_2, f, v, f)$ is nonblocking in the wide sense for s-slot connections, $1 \leqslant s \leqslant f$, under flexible assignment*

and functional division strategy if and only if: $m \geqslant m_1 + m_2$, *where*

$$m_1 = \min\left\{\left\lfloor \frac{n_1 f - \left\lfloor \frac{f}{2} \right\rfloor}{v\left(\left\lceil \frac{f}{2} \right\rceil + 1\right)} \right\rfloor + \left\lfloor \frac{n_2 f - \left\lfloor \frac{f}{2} \right\rfloor}{v\left(\left\lceil \frac{f}{2} \right\rceil + 1\right)} \right\rfloor + 1; \right.$$
$$\left. \left\lfloor \frac{r_1 n_1 f - \left\lfloor \frac{f}{2} \right\rfloor}{v\left(\left\lceil \frac{f}{2} \right\rceil + 1\right)} \right\rfloor + 1; \left\lfloor \frac{r_2 n_2 f - \left\lfloor \frac{f}{2} \right\rfloor}{v\left(\left\lceil \frac{f}{2} \right\rceil + 1\right)} \right\rfloor + 1 \right\} \tag{5.49}$$

denotes the number of middle stage switches which serves s-slot connections for which $1 \leqslant s \leqslant \lfloor f/2 \rfloor$, *and*

$$m_2 = \min\left\{\left\lfloor \frac{n_1 - 1}{v} \right\rfloor + \left\lfloor \frac{n_2 - 1}{v} \right\rfloor + 1; \right.$$
$$\left. \left\lfloor \frac{r_1 n_1 - 1}{v} \right\rfloor + 1; \left\lfloor \frac{r_2 n_2 - 1}{v} \right\rfloor + 1 \right\} \tag{5.50}$$

denotes the number of middle stage switches used for serving s-slot connections for which $\lfloor f/2 \rfloor < s \leqslant f$.

Proof. For m_1 set $s_{\max} = \lfloor f/2 \rfloor$ in equation 5.48. When $s > \lfloor f \rfloor$ only one connection can be set up in one link, so it corresponds to the case when $f = s = 1$. When we put these values in equation (5.48), we obtain (5.50). □

5.2.5 Rearrangeable Switching Fabrics

Rearrangeable time-division switching fabrics for unicast 1-slot connections were considered by Jajszczyk [63, 64]. The conditions in the symmetrical case are given by the following theorem.

THEOREM 5.32 $C_{TD}(n, r, f, m, v, f_0)$ *is rearrangeable if and only if*

$$m \geqslant \left\lfloor \frac{nf - 1}{vf_0} \right\rfloor + 1 \tag{5.51}$$

Proof. The proof is very similar to the proof of theorem 5.18. The difference is that in one center stage switch vf_0 subassignments can be realized. □

THEOREM 5.33 $C_{TD}(n_1, r_1, f_1, m, n_2, r_2, f_2, v, f_0)$ *is rearrangeable if and only if*

$$m \geqslant \left\lfloor \frac{\max\{n_1 f_1, n_2 f_2\} - 1}{v f_0} \right\rfloor + 1. \tag{5.52}$$

Proof. The proof is very similar to the proof of theorem 5.19. However, the bipartite graph representing the maximum assignment is now of maximum degree $d \leqslant \max\{n_1 f_1, n_2 f_2\}$, and in one center stage switch connections colored with the same $v f_0$ colors can always be realized. □

Strict-sense, wide-sense, and rearrangeable nonblocking two-sided time-division switching fabrics require different numbers of second stage switches. The comparison for $f_1 = f_0 = 32$ and different n, $s_{\max}$, and v is given in Table 5.3. It is obvious, that the lowest number of second stage switches is required in rearrangeable nonblocking switching fabrics. In case of 1-slot connections strict-sense conditions and wide-sense nonblocking conditions when flexible assignment is used are equal. For $1 < s \leqslant s_{\max}$ and $s_{\max} = f_1 = f_0$ the lowest number of switches is obtained for wide-sense nonblocking switching fabrics with functional division of second stage switches. The number of required switches m as a function of $s_{\max}$ for $s_{\max} = f_1 = f_0 = 32$ and different n is plotted in Fig. 5.28. For $s_{\max} = f_1 = f_0$ wide-sense nonblocking switching fabrics with flexible assignment and strict-sense nonblocking switching fabrics require the same number of the second stage switches.

		$v=1$				$v=2$			
	$n=$	8	16	24	32	8	16	24	32
SNB									
$s_{\max}=$	1	15	31	47	63	7	15	23	31
	2	31	63	95	127	15	31	47	63
	6	101	203	305	407	51	101	153	203
	16	241	497	753	1009	121	249	377	505
	32	449	961	1473	1985	225	481	737	993
WNB flexible									
$s_{\max}=$	1	15	31	47	63	7	15	23	31
	2	17	33	49	65	9	17	23	31
	6	19	37	57	75	9	19	29	37
	16	29	59	89	119	15	29	45	59
	32	449	961	1473	1985	225	481	737	993
WNB functional									
$s_{\max}=$	32	44	60	76	92	22	30	38	46
RNB									
$s_{\max}=$	1	8	16	24	32	4	8	12	16

Table 5.3. Number of second stage switches in nonblocking $C_{TD}(n, r, 32, m, v, 32)$ switching fabrics; SNB - strict-sense nonblocking, WNB - wide-sense nonblocking, RNB - rearrangeable nonblocking

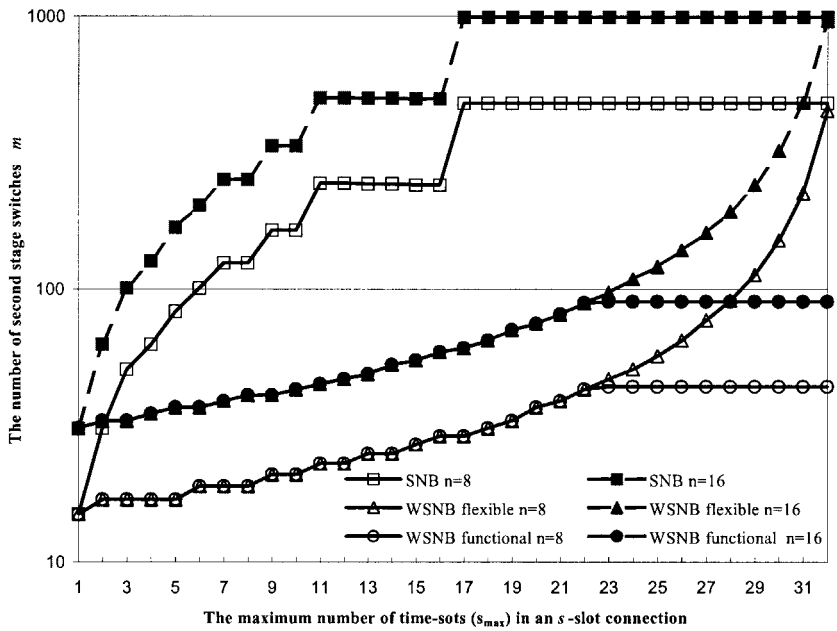


Figure 5.28. Number of second stage switches versus $s_{\max}$; $f_1 = f_0 = 32$

5.2.6 Repackable switching fabrics

Repackable time-division switching fabrics were considered by Ohta [122]. The following notation will be used to describe the repacking algorithm. Let $L_{M_k}(I_i, O_j)$ denote the number of connections (I_i, O_j) set up through switch M_k; let $M_k^{\max}(I_i, O_j)$ denote the center stage switch for which $L_{M_k}^{\max}(I_i, O_j) = \max_k L_{M_k}(I_i, O_j)$; and let $M_k^{\min}(I_i, O_j)$ denote the center stage switch for which $L_{M_k}^{\min}(I_i, O_j) = \min_k L_{M_k}(I_i, O_j)$. The control is as follows.

ALGORITHM 5.15 *RepackingTD*

Step 1 If (I_i, O_j) is a new connection, set this connection through M_k for which $L_{M_k}(I_i, O_j) = L_{M_k}^{\min}(I_i, O_j)$.

Step 2 If (I_i, O_j) is terminated and it is routed through M_k then if $L_{M_k}(I_i, O_j) = L_{M_k}^{\max}(I_i, O_j)$ before this disconnection do nothing, else rearrange another connection (I_i, O_j) from switch $M_k^{\max}(I_i, O_j)$ to M_k.

THEOREM 5.34 *$C_{TD}(n, r, f, m, v, f_0)$ is repackable under algorithm RepackingTD if and only if:*

$$vf_0 \geqslant \left\lfloor \frac{nf - r}{m} \right\rfloor + r. \tag{5.53}$$

From the above theorem it can be seen that nonblocking conditions depend on r. For $r > vf_0$ the conditions given in Theorem 5.34 cannot be fulfilled. For lower r repacking will result in lower number of center stage switches. The comparison of strict-sense nonblocking and repackable switching fabrics for $f = f_0 = 32$, $v = 1$, $n = 8$, and $n = 16$ is given in Fig. 5.29.

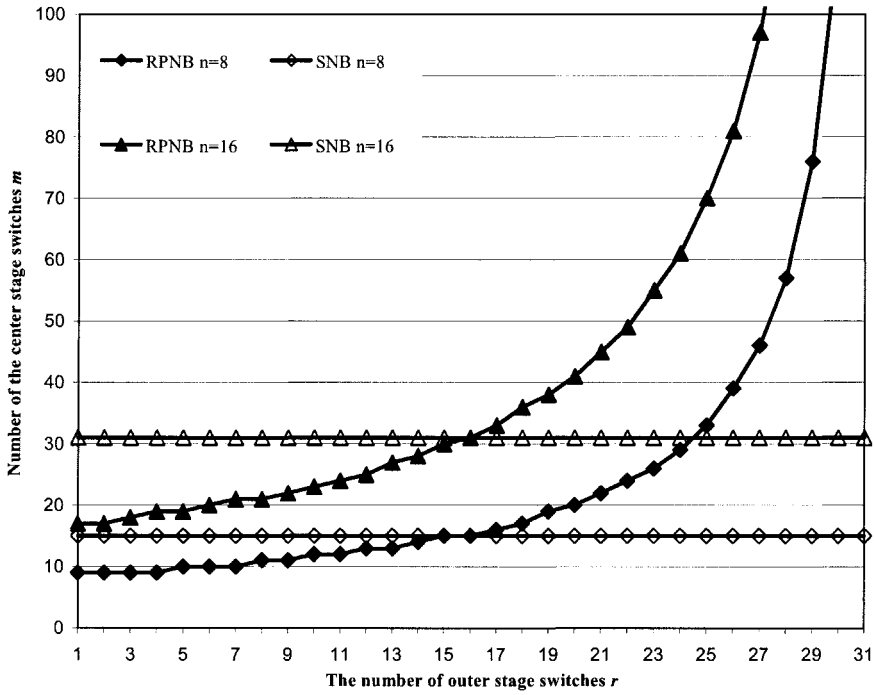


Figure 5.29. Number of center stage switches versus r for SNB and RPNB switching fabrics with $f = f_0 = 32$ and $v = 1$

5.3 Multirate Switching

5.3.1 Multirate Switching Fabrics

An architecture of the two-sided three-stage switching fabric in the multirate environment is shown in Fig. 5.30. Similarly as in the space-division and time-division switching fabrics, it consists of r_1 switches in the first stage, r_2 switches in the third stage, and m switches in the second stage. Each pair of switches in adjacent stages is connected between themselves by means of v bidirectional links. Each of the first stage switches has n_1 input of capacity β_1 ($\beta_1 \leqslant 1$), and mv outputs of capacity 1. Switches in the third stage each have mv inputs of capacity 1 and n_2 outputs of capacity β_2 ($\beta_2 \leqslant 1$). Middle stage switches are of size $r_1 v \times r_2 v$, each link of capacity 1. This switching fabric will be denoted by $C_{MR}(n_1, r_1, \beta_1, m, n_2, r_2, \beta_2, v)$. When $n_1 = n_2 = n$, $r_1 = r_2 = r$, and $\beta_1 = \beta_2 = \beta$ the switching fabric is called symmetrical and will be denoted by $C_{MR}(n, r, \beta, m, v)$.

If a new connection $\langle x, y, \omega \rangle$ is to be set up through the switching fabric, a control algorithm has to find a middle stage switch which is accessible for this connection. This switch must have available bandwidth of at least ω in one of its v inputs connected to the first stage switch containing input terminal x, and available bandwidth of the same weight in one of its v outputs to the third stage switch with output terminal y. In order to preserve the cell order, we will assume that a connection is routed through a single link to one second stage switch.

Algorithms RAN, SEQ, MINIX, Q-RAN, CS, STU, and PACK proposed for space-division switching may be used after simple modifications. In the case of PACK algorithm it should be noted that the

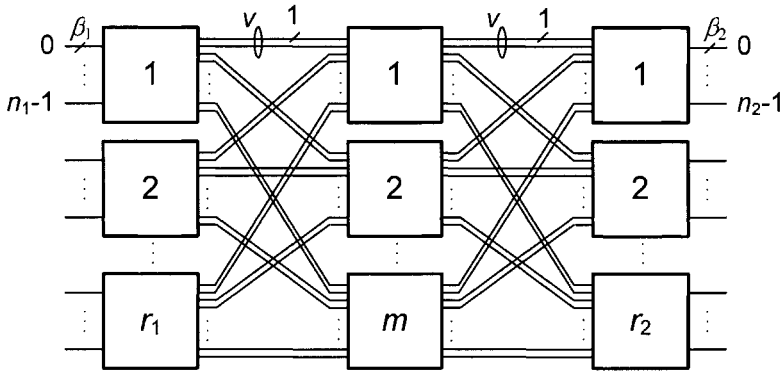


Figure 5.30. A multirate two-sided v-dilated three-stage switching fabric

switch load should be now calculated as the total bandwidth used by connections set up through the considered center stage switch. Algorithm PACK-L and UNPACK-L proposed for time-division switching can also be used. Routing strategy with functional division of center stage switches can be used as well in these switching fabrics [101]. In this routing strategy, similarly as in the time-division switching, m_1 center stage switches are reserved for connections with weights less than or equal to 0.5 (i.e., $b \leqslant \omega \leqslant 0.5$) and m_2 center stage switches are reserved for connections of weights greater than 0.5 (i.e., $0.5 < \omega \leqslant B$). This approach may be applied for both discrete bandwidth and continuous bandwidth cases. The routing strategy with functional division of middle stage switches can be further modified in the way proposed by Gao and Hwang [44]. In this modified algorithm (called also *quota scheme*) connections are divided into three groups: light, medium, and heavy. A connection of weight ω is a heavy connection if $\omega > 1/(p+1)$, a medium connection if $1/(p+1) \geqslant \omega > 1/(p+2)$, and a light connection if $1/(p+1) \geqslant \omega$, where $p = \lfloor 1/B \rfloor$. The set of middle stage switches is divided into three subsets of m_1, m_2, and m_3 switches. Each of m_1 switches is reserved for carrying up to l_1 of light and medium connections and as many heavy connections as possible. Each of m_2 switches is reserved for carrying up to l_2 light connections and as many medium connections as possible.

5.3.2 Strict-sense Nonblocking

The first upper bound of nonblocking conditions in the case of continuous bandwidth was proposed by Melen and Turner [114]. This upper bound was later improved by Chung and Ross [22]. Asymmetrical switch configurations were considered in [24]. More generalized three-stage Clos switching fabrics were considered by Liotopoulos and Chalasani [104]. The results derived in those papers were limited to $b = 0$ or $B \in (1-b, \beta]$. Both sufficient and necessary non-blocking conditions for any B and $b > 0$ were proved in [74] and [73] in the case of symmetrical and asymmetrical three-stage Clos switching networks, respectively. In some papers blocking probability at the connection level was also considered [101, 154, 166].

First strict sense nonblocking conditions in the discrete bandwidth case will be considered.

THEOREM 5.35 *$C_{MR}(n_1, r_1, \beta_1, m, n_2, r_2, \beta_2, v)$ is strict-sense nonblocking for discrete bandwidth case if and only if:*

$$m \geqslant \left\lfloor \frac{\min\{A_1(n_1, \beta_1, B); B_1(r_2, n_2, \beta_2)\}}{vC_1(B)} \right\rfloor + \left\lfloor \frac{\min\{A_1(n_2, \beta_2, B); B_1(r_1, n_1, \beta_1)\}}{vC_1(B)} \right\rfloor + 1, \quad (5.54)$$

where

$$A_1(n, \beta, B) = (n-1)\left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - B}{b} \right\rfloor, \quad (5.55)$$

$$B_1(r, n, \beta) = (r-1)\, n \left\lfloor \frac{\beta}{b} \right\rfloor, \quad (5.56)$$

and

$$C_1(B) = \left\lfloor \frac{1 - B + b}{b} \right\rfloor. \quad (5.57)$$

Proof. Necessary conditions can be proved by showing the blocking state in the switching fabric with less value of m than that given by (5.54). The following path searching algorithm will be used. If a new connection of weight ω arrives at the same first stage switch as the last connection set up, we start to search a path from the interstage link through which the last connection was set up. When a new connection appears in another first stage switch, we start to search a path from the middle stage switch next to the last engaged. Let connection $(I_i, O_j, B$ is to be set up. In inequality (5.54):

- $A_1(n_1, \beta_1, B)$ represents the maximum number of connections with weight b, that can constitute a state of first stage switch I_i, compatible with a connection of weight B (see also Fig. 5.31);
- $A_1(n_2, \beta_2, B)$ represents the maximum number of connections with weight b, that can constitute a state of third stage switch O_j, compatible with a connection of weight B (see also Fig. 5.31);
- $B_1(r_2, n_2, \beta_2)$ represents the maximum number of connections with weight b, that can fit in all third stage switches but switch O_j;
- $B_1(r_1, n_1, \beta_1)$ represents the maximum number of connections with weight b, that can fit in all third stage switches but switch I_i;
- $C_1(B)$ represents the maximum number of connections with weight b, that can make an interstage link inaccessible by a connection of weight B.

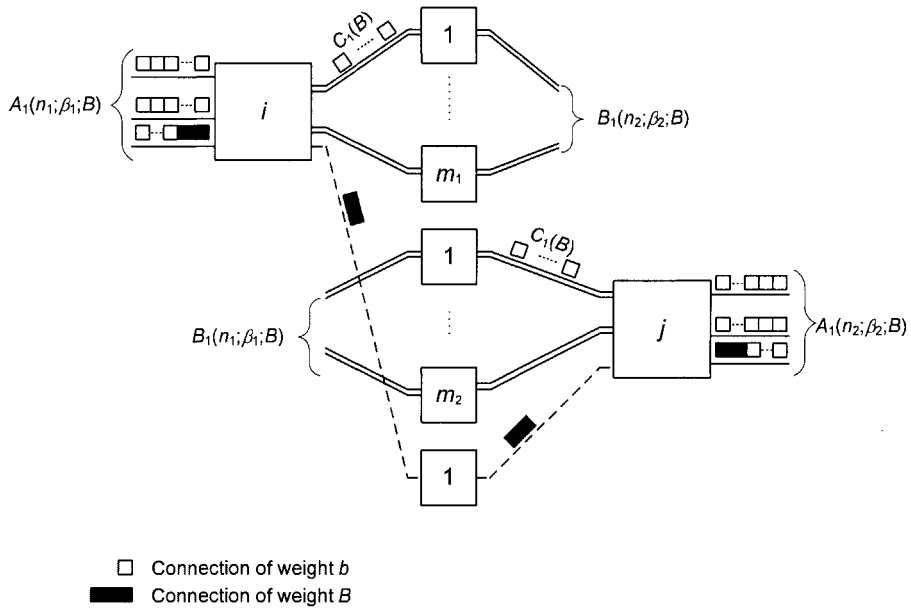


Figure 5.31. Worst case scenario for connection placement in switches I_i and O_j in the strictly non-blocking mode.

The following set of events leads to the occupancy of all middle stage switches given by inequality (5.54).

Step 1 Set up $\lfloor (1 - B + b)/b \rfloor$ connections (I_i, O_k, b), where $k \neq j$.

Step 2 Set up connection (I_i, O_k, γ), where $\gamma = 1 - b\lfloor (1-B+b)/b \rfloor < B$.

Step 3 Set up connection (I_i, O_k, b), disconnect (I_i, O_k, γ), and then set up $\lfloor (1 - B + b)/b \rfloor$-1 connections (I_i, O_k, b).

Step 4 Repeat connections and disconnections from Steps 2 and 3 up until the number of connections (I_i, O_k, b) is equal to $vC_1(B) \lfloor \min\{A_1(n_1, \beta_1, B); B_1(r_2, n_2, \beta_2)\} / vC_1(B) \rfloor$.
The number of middle stage switches occupied by these connections is equal to $m_1 = \lfloor \min\{A_1(n_1, \beta_1, B); B_1(r_2, n_2, \beta_2)\} / vC_1(B) \rfloor$, and these switches will be inaccessible to the new connection of weight B from switch I_i.

Step 5 Repeat Steps 1 to 4 for connections (I_k, O_j, b), $k \neq i$. The next $m_2 = \lfloor \min\{A_1(n_2, \beta_2, B); B_1(r_1, n_1, \beta_1)\} / vC_1(B) \rfloor$ middle stage switches will be occupied by these connections, and these switches

will be inaccessible to the new connection of weight B to the third stage switch O_j.

In the above switching fabric, m_1+m_2 middle stage switches are occupied and the new connection (I_i, O_j, B) will occupy the second stage switch numbered $m_1 + m_2+1$. It should be noted that the weight B is available in one of the input links of switch I_i, as well as in one of the output links of switch O_j.

Sufficiency will be proved by showing the worst state in the switching network. In order to maximize the utilization of all links and minimize capacity fragmentation of the switch, we consider that only connections of weight b are already set up in the switching fabric. Suppose we want to add the new connection (I_i, O_j, ω), $0 < b \leqslant \omega \leqslant B$. In switch I_i we may have at most $\lfloor \beta_1/b \rfloor$ connections of weight b in each of $n_1 - 1$ input links, and $\lfloor (\beta_1 - \omega)/b \rfloor$ such connections in the remaining link. So we may have at most $(n_1 - 1) \lfloor \beta_1/b \rfloor + \lfloor (\beta_1 - \omega)/b \rfloor = A_1 (n_1, \beta_1, \omega)$ connections of weight b in switch I_i. However, in all the third stage switches, except switch O_j, it is possible to set up no more than $(r_2 - 1) n_2 \lfloor \beta_2/b \rfloor = B_1 (r_2, n_2, \beta_2)$ connections of weight b. Thus the total number of connections is $\min \{A_1 (n_1, \beta_1, \omega); B_1 (r_2, n_2, \beta_2)\}$.

The interstage link is inaccessible by a new connection if $C_1 (\omega) = \lfloor (1 - \omega)/b \rfloor + 1 = \lfloor (1 - \omega + b)/b \rfloor$ connections of weight b are already set up through this link. So the connections from switch I_i can occupy $\lfloor \min \{A_1 (n_1, \beta_1, \omega); B_1 (r_2, n_2, \beta_2)\} / vC_1 (\omega) \rfloor$ middle stage switches.

Similar considerations for the third stage switch O_j show that no more than $\lfloor \min \{A_1 (n_2, \beta_2, \omega); B_1 (r_1, n_1, \beta_1)\} / vC_1 (\omega) \rfloor$ middle stage switches will be inaccessible by the connection (I_i, O_j, ω), where $A_1 (n_2, \beta_2, \omega) = (n_2 - 1) \lfloor \beta_2/b \rfloor + \lfloor (\beta_2 - \omega)/b \rfloor$, and $B_1 (r_1, n_1, \beta_1) = (r_1 - 1) n_1 \lfloor \beta_1/b \rfloor$. In the worst case these sets of middle stage switches are disjoint and one more switch is needed for the connection (I_i, O_j, ω), so

$$m \geqslant \left\lfloor \frac{\min \{A_1 (n_1, \beta_1, \omega); B_1 (r_2, n_2, \beta_2)\}}{vC_1 (\omega)} \right\rfloor + \left\lfloor \frac{\min \{A_1 (n_2, \beta_2, \omega); B_1 (r_1, n_1, \beta_1)\}}{vC_1 (\omega)} \right\rfloor + 1, \quad (5.58)$$

$$A_1 (n_1, \beta_1, \omega) = (n_1 - 1) \left\lfloor \frac{\beta_1}{b} \right\rfloor + \left\lfloor \frac{\beta_1 - \omega}{b} \right\rfloor, \quad (5.59)$$

$$A_1 (n_2, \beta_2, \omega) = (n_2 - 1) \left\lfloor \frac{\beta_2}{b} \right\rfloor + \left\lfloor \frac{\beta_2 - \omega}{b} \right\rfloor, \quad (5.60)$$

$$B_1 (r_1, n_1, \beta_1) = (r_1 - 1) n_1 \left\lfloor \frac{\beta_1}{b} \right\rfloor, \quad (5.61)$$

$$B_1(r_2, n_2, \beta_2) = (r_2 - 1)\, n_2 \left\lfloor \frac{\beta_2}{b} \right\rfloor, \tag{5.62}$$

and

$$C_1(\omega) = \left\lfloor \frac{1 - \omega + b}{b} \right\rfloor. \tag{5.63}$$

The function (5.58) must be maximized through all ω, and it reaches maximum for $\omega = B$. Applying $\omega = B$ in inequality (5.58), we obtain formula (5.54). □

The nonblocking conditions in the discrete bandwidth case when $1/b$ is an integer and $\beta = v = 1$ were given in [22]. The case with $\beta < 1$ was considered in [104]. In the case of symmetrical multirate switching fabrics we obtain:

COROLLARY 5.36 *$C_{MR}(n, r, \beta, m, v)$ is strict-sense nonblocking for discrete bandwidth case if and only if:*

$$m \geqslant 2 \left\lfloor \frac{(n-1)\left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - B}{b} \right\rfloor}{v \left\lfloor \frac{1 - B + b}{b} \right\rfloor} \right\rfloor + 1. \tag{5.64}$$

Proof. Put $n_1 = n_2 = n$, $r_1 = r_2 = r$, and $\beta_1 = \beta_2 = \beta$ in equation (5.54). Since for $r \geqslant 2$ we have $A_1(n, \beta, B) < B_1(n, \beta, B)$, we obtain inequality (5.64). □

When we replace β_1, β_2, b, and B with f_1, f_2, 1, and $s_{\max}$, respectively, and we denote the interstage link capacity by f_0 in inequality (5.64), we obtain wide-sense nonblocking conditions for switching fabrics with s-slot connections under flexible assignment algorithm given in inequality (5.48).

Let us now consider $C_{MR}(n_1, r_1, \beta_1, m, n_2, r_2, \beta_2, v)$ in the continuous bandwidth case. Since input and output terminals have different capacities, weight ω is limited by $0 < b \leqslant \omega \leqslant B \leqslant \min\{\beta_1; \beta_2\} \leqslant 1$. We will use functions (4.27), (4.28), (4.29), (4.30), and (4.31) defined in section 4.2.3.2.

THEOREM 5.37 *$C_{MR}(n_1, r_1, \beta_1, m, n_2, r_2, \beta_2, v)$ is strict-sense nonblocking for continuous bandwidth case if and only if:*

$$m \geqslant \left\lfloor \frac{\min\{A_2(n_1, \beta_1, B); B_2(r_2, n_2, \beta_2, B)\}}{v} \right\rfloor + \tag{5.65}$$

$$+ \left\lfloor \frac{\min\{A_2(n_2, \beta_2, B); B_2(r_1, n_1, \beta_1, B)\}}{v} \right\rfloor + 1,$$

where:

$$A_2(n,\beta,B) = \tag{5.66}$$

$$\begin{cases} (n-1)\left\lfloor \dfrac{\beta}{b} \right\rfloor & \text{for } B \in (1-b,\ \beta] \\ \dfrac{(n-1)\left\lfloor \dfrac{\beta}{b} \right\rfloor + \left\lfloor \dfrac{\beta - B}{b} \right\rfloor}{2} & \text{for } B \in \left(1-2b,\ \frac{1}{2}\right] \text{ and } \frac{1}{4} < b < \frac{1}{2} \\ (n-1)\,P(\beta; 1-B) & \text{for all other } B \text{ and } R_1(\beta; 1-B) < b \\ (n-1)\,P(\beta; 1-B) + \lfloor R_3(n-1; a(\beta,B)) \rfloor + P(\alpha(\beta,B); 1-B) & \text{for all other } B \text{ and } b \leqslant R_1(\beta; 1-B) < 2b \\ (n-1)\,P(\beta; 1-B) + P[(n-1)\,R_1(\beta; 1-B) + R_5(\beta - B); 1-B] & \text{for all other } B \text{ and } R_1(\beta; 1-B) \geqslant 2b \end{cases}$$

$$B_2(r,n,\beta,B) = \tag{5.67}$$

$$\begin{cases} (r-1)\,n\left\lfloor \dfrac{\beta}{b} \right\rfloor & \text{for } B \in (1-b,\ \beta] \\ \dfrac{(r-1)\,n\left\lfloor \dfrac{\beta}{b} \right\rfloor}{2} & \text{for } B \in \left(1-2b,\ \frac{1}{2}\right] \text{ and } \frac{1}{4} < b < \frac{1}{2} \\ (r-1)\,nP(\beta; 1-B) & \text{for all other } B \text{ and } R_1(\beta; 1-B) < b \\ (r-1)\,nP(\beta; 1-B) + P((r-1)\,nR_1(\beta; 1-B); 1-B) & \text{for all other } B \text{ and } b \leqslant R_1(\beta; 1-B) \end{cases}$$

$$a(\beta,B) = R_2(\beta; 1-B) + 1 \tag{5.68}$$

$$\alpha(\beta, B) = [(n-1) - a(\beta, B)\lfloor R_3(n-1; a(\beta, B))\rfloor] R_1(\beta; 1-B) + \\ + R_5(\beta - B) \tag{5.69}$$

Proof. Sufficient conditions will be proved by showing the worst case in the switching fabric. Suppose we want to add the new connection (I_i, O_j, ω), $0 < b \leqslant \omega \leqslant B$. Any interstage link from switch I_i will be inaccessible to the new connection of weight ω, if the sum of connection weights already set up through this link is greater than $1-\omega$. In the worst case this sum of weights should be as small as possible, say $1-\omega+\varepsilon$, where ε is close to but greater than 0. We have three cases:

1 $1-\omega < b$,

2 $b \leqslant 1-\omega < 2b$ and $1-\omega \geqslant B$,

3 other values of $1-\omega$.

Case 1: $1-\omega < b$. The interstage link is inaccessible to the new connection, if it carries one connection of weight b. In the worst case, each connection with this weight from switch I_i may be set up through a separate interstage link. At one input link there may be at most $\lfloor \beta_1/b \rfloor$ connections of weight b. There are $n_1 - 1$ such input links of switch I_i. In the last input link of this switch there is a free bandwidth of weight $\beta_1 - \omega$. Since $1-\omega < b$ and $\beta_1 \leqslant 1$, then $\beta_1 - \omega < b$. At this link we cannot set up a connection of weight b. In the first stage switch I_i there may be $A_2(n_1, \beta_1, \omega) = (n_1 - 1)\lfloor \beta_1/b \rfloor$ connections of weight b set up, and these connections may occupy $\lfloor A_2(n_1, \beta_1, \omega)/v \rfloor$ middle stage switches. However, in all third stage switches, except switch O_j, it is possible to set up no more than $B_2(r_2, n_2, \beta_2, \omega) = (r_2 - 1)n_2 \lfloor \beta/b \rfloor$ connections of weight b, so no more than $\lfloor B_2(r_2, n_2, \beta_2, \omega)/v \rfloor$ middle stage switches will be occupied.

Case 2: $b \leqslant 1-\omega < 2b$ and $1-\omega \geqslant B$. The interstage link is inaccessible to the new connection of weight ω, if it carries connections of a total weight greater than $1-\omega$. Since $1-\omega \geqslant B$, there is no possible way to set up a connection of such weight. But because $1-\omega < 2b$, the interstage link will be inaccessible by the new connection of weight ω, if it carries two connections of weight b. Similarly as in case 1, we may have $(n_1 - 1)\lfloor \beta_1/b \rfloor$ connections of weight b at $n_1 - 1$ input links of switch I_i, and $\lfloor (\beta_1 - \omega)/b \rfloor$ connections of such weight at the last input link of this switch. At the input links of switch I_i there may be $(n_1 - 1)\lfloor \beta_1/b \rfloor + \lfloor (\beta_1 - \omega)/b \rfloor$ connections of weight b, and these connections will make $A_2(n_1, \beta_1, \omega) = [(n_1 - 1)\lfloor \beta_1/b \rfloor + \lfloor (\beta_1 - \omega)/b \rfloor]/2$ links outgoing from switch I_i inaccessible by the new connection. Therefore,

$\lfloor A_2(n_1, \beta_1, \omega)/v \rfloor$ middle stage switches will be inaccessible, provided that all these connections can be accepted in some third stage switches, other than switch O_j. Since in these third stage switches it is possible to set up at most $(r_2 - 1)n_2 \lfloor \beta_2/b \rfloor$ connections of weight b, no more than $\lfloor B_2(r_2, n_2, \beta_2, \omega)/v \rfloor$ middle stage switches will be occupied, where $B_2(r_2, n_2, \beta_2, \omega) = [(r_2 - 1)n_2 \lfloor \beta_2/b \rfloor]/2$.

Case 3: $1 - \omega \geqslant 2b$ or $b \leqslant 1 - \omega < 2b$ and $1 - \omega \leqslant B$. If $1 - \omega < B$ only one connection of such a weight may be set up. In the other case, at least two connections of total weight greater than $1 - \omega$ must be set up in an interstage link. In one input link we can have no more than $\lim_{\varepsilon \to 0} \lfloor \beta_1/(1 - \omega + \varepsilon) \rfloor$ connections of total weight greater than $1 - \omega$, so this number of connections is given by function $P(\beta; 1 - \omega)$. So, at most $(n_1 - 1)P(\beta_1; 1 - \omega)$ connections of such weight may be set up in the first stage switch I_i. There is still free bandwidth of weight $\beta_1 - \omega$ in the last input link, but it cannot be used by a connection of weight greater than $1 - \omega$ ($\beta_1 - \omega \leqslant 1 - \omega$). In each of the $n_1 - 1$ input links there is free bandwidth of weight $\beta_1 - P(\beta_1; 1 - \omega)(1 - \omega) = R_1(\beta_1; 1 - \omega)$, but when $R_1(\beta_1; 1 - \omega) < b$ this bandwidth cannot be used by the next connection. This means that, in switch I_i we have $A_2(n_1, \beta_1, \omega) = (n_1 - 1)P(\beta_1; 1 - \omega)$ connections of weight greater than $1 - \omega$, and these connections occupy $\lfloor A_2(n_1, \beta_1, \omega)/v \rfloor$ middle stage switches.

When $b \leqslant R_1(\beta_1; 1 - \omega) < 2b$, the remaining bandwidth in each link can be used by the next connection. Several such connections of weight $R_1(\beta_1; 1 - \omega)$ in one interstage link may lead to a state, in which this link will be inaccessible to the new connection (the minimum number of these connections is denoted by $a(\beta_1, \omega)$. The next interstage link will be inaccessible by the new connection, if it carries $a(\beta_1, \omega) = R_2(\beta_1; 1 - \omega) + 1$ connections of weight $R_1(\beta_1; 1 - \omega)$. This means, that the next $\lfloor R_3(n_1 - 1; a(\beta_1, \omega) \rfloor$ interstage links from switch I_i will be inaccessible to the new connection. In switch I_i, we now have $n_1 - 1 - a(\beta_1, \omega) \lfloor R_3(n_1 - 1; a(\beta_1, \omega) \rfloor$ input links with available bandwidth $R_1(\beta_1; 1 - \omega)$, and one link with available bandwidth of $R_5(\beta_1 - \omega)$ (function $R_5(\gamma)$ indicates whether the bandwidth quantity γ is less than b or not, so it determines whether this bandwidth can be used by a connection or not). Connections of such weights may occupy a bandwidth of weight $\alpha(\beta_1, \omega) = [(n_1 - 1) - a(\beta_1, \omega) \lfloor R_3(n_1 - 1; a(\beta_1, \omega)) \rfloor] R_1(\beta_1; 1 - \omega) + R_5(\beta_1 - \omega)$ in an interstage link. If $\alpha(\beta_1, \omega)$ is greater than $1 - \omega$, then this interstage link will also be inaccessible to the new connection from switch I_i. Whether this interstage link is accessible or not, can be calculated by function $P(\alpha(\beta_1, \omega); 1 - \omega)$. The state of switch I_i for this case is shown in Fig. 5.32. In this state, we may have $A_2(n_1, \beta_1, \omega) = (n_1 - 1)P(\beta_1; 1 - \omega) + \lfloor R_3(n_1 - 1; a(\beta_1, \omega) \rfloor + P(\alpha(\beta_1, \omega); 1 - \omega)$ inter-

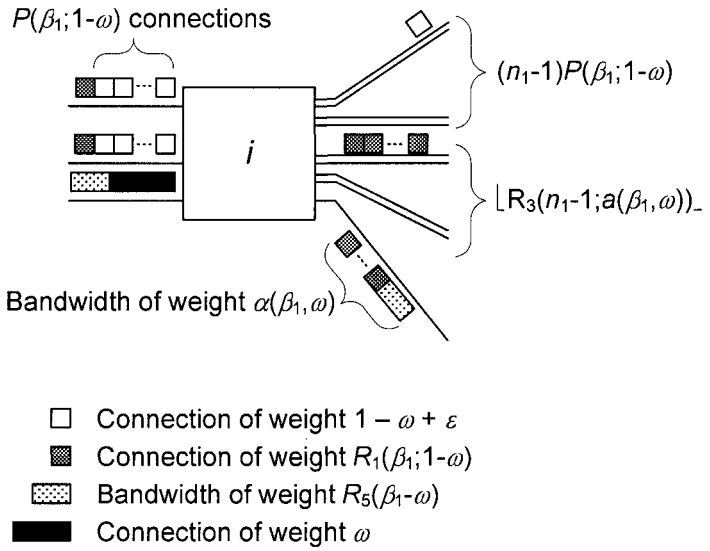


Figure 5.32. The state of switch I_i in case $b \leqslant R_1(\beta_1; 1-\omega) < 2b$

stage links from switch I_i, which are inaccessible by the new connection of weight ω. These links will fully occupy $\lfloor A_2(n_1, \beta_1, \omega)/v \rfloor$ middle stage switches.

For $R_1(\beta_1; 1-\omega) > 2b$ this remaining bandwidth may be divided among more than one connection. However, these connections may occupy no more than $P([(n_1-1)R_1(\beta_1; 1-\omega) + R_5(1-\omega)]; 1-\omega)$ interstage links. This means that $A_2(n_1, \beta_1, \omega) = (n_1-1)P(\beta_1; 1-\omega) + P([(n_1-1)R_1(\beta_1; 1-\omega) + R_5(1-\omega)]; 1-\omega)$ interstage links may be inaccessible by the new connection of weight ω and they will occupy $\lfloor A_2(n_1, \beta_1, \omega)/v \rfloor$ middle stage switches.

Similarly, as in cases 1 and 2, all connections in switch I_i, have to be accepted by all third stage switches except switch O_j. The number of connections of weight greater than $1-\omega$, which can be accepted is given by: $B_2(r_2, n_2, \beta_2, \omega) = (r_2-1)n_2P(\beta_2; 1-\omega) + P((r_2-1)n_2R_1(\beta_2; 1-\omega); 1-\omega)$. These connections will occupy no more than $\lfloor B_2(r_2, n_2, \beta_2, \omega)/v \rfloor$ middle stage switches.

Combining these cases together, we can write that in the middle stage $\lfloor \min\{A_2(n_1, r_1, \omega); B_2(r_2, n_2, \beta_2, \omega)\}/v \rfloor$ switches will be inaccessible to the new connection of weight ω, where $A_2(n_1, r_1, \omega)$, and $B_2(r_2, n_2, \beta_2, \omega)$ are given by equations (5.66) and (5.67), respectively.

On the other hand, any interstage link to switch O_j will be inaccessible by a new connection of weight ω, if the sum of connection weights already

set up through this link is greater than $1-\omega$. Similar considerations, as for switch I_i, show that $\lfloor A_2(n_2, r_2, \omega)/v \rfloor$ middle stage switches will be inaccessible to the new connection of weight ω. However, in switch O_j, we cannot set up more connections than can be accepted in all first stage switches except switch I_i. Hence, no more than $\lfloor B_2(r_1, n_1, \beta_1, \omega)/v \rfloor$ middle stage switches can be occupied. This implies that another set of $\lfloor \min\{A_2(n_2, r_2, \omega); B_2(r_1, n_1, \beta_1, \omega)\}/v \rfloor$ middle stage switches is inaccessible to the new connection, where $A_2(n_2, r_2, \omega)$, and $B_2(r_1, n_1, \beta_1, \omega)$ are given by equations (5.66) and (5.67), respectively. To set up the connection (I_i, O_j, ω), one more switch is needed in the middle stage. Therefore, we have:

$$m \geqslant \left\lfloor \frac{\min\{A_2(n_1, \beta_1, \omega); B_2(r_2, n_2, \beta_2, \omega)\}}{v} \right\rfloor + \left\lfloor \frac{\min\{A_2(n_2, \beta_2, \omega); B_2(r_1, n_1, \beta_1, \omega)\}}{v} \right\rfloor + 1. \tag{5.70}$$

The function (5.70) must be maximized through all ω, and it reaches maximum for $\omega = B$. Applying $\omega = B$ in (5.70), we obtain formula (5.65). □

It should be noted that in cases 1 and 2 and also in case 3 when $R_1(\beta; 1-B) < b$ Theorem 5.37 provides necessary conditions as well. Such conditions can be proved by showing a set of events leading to the occupancy of the number of middle stage switches given by (5.65). This set of events can be constructed in a similar way, as in the proof of Theorem 5.35.

Multirate switching networks are more generalized case of space-division and multi-channel switching. Theorems 5.35 and 5.37 should then include already known results. Let us consider the $C_{MR}(n_1, r_1, 1, m, n_2, r_2, 1, v)$ with $b = B = 1$. For the continuous bandwidth case we have $1 \in (0, 1]$, which is case 1. We obtain $A_2(n_1,\ 1,\ 1) = n_1 - 1$, $B_2(r_2, n_2,\ 1,\ 1) = (r_2 - 1)n_2$, $A_2(n_2,\ 1,\ 1) = n_2 - 1$, and $B_2(r_1, n_1,\ 1,\ 1) = (r_1 - 1)n_1$. Putting this values in equation (5.65) we will finally get Corollary 5.27. More general architecture of three-stage multirate switching fabrics with different number of links and of different capacities in each of input and output stage switches as well as the different number of interstage links between switches was considered in [77].

5.3.3 Wide-sense Nonblocking Conditions

5.3.3.1 Unicast Connections

For unicast connections the number of center stage switches required can be reduced by using the routing strategy with functional division.

THEOREM 5.38 *$C_{MR}(n_1, \beta_1, r_1, m, n_2, \beta_2, r_2, v)$ is wide-sense nonblocking for continuous bandwidth case, when the algorithm with the functional division is used, if: $m \geqslant m_1 + m_2$, where*

$$m_1 \geqslant \left\lfloor \frac{\min\{A_2(n_1, \beta_1, 0.5); B_2(r_2, n_2, \beta_2, 0.5)\}}{v} \right\rfloor + \left\lfloor \frac{\min\{A_2(n_2, \beta_2, 0.5); B_2(r_1, n_1, \beta_1, 0.5)\}}{v} \right\rfloor + 1, \tag{5.71}$$

is the group of switches used for connections with $b \leqslant \omega \leqslant 0.5$, and A_2, B_2, are defined in Theorem 5.37, and

$$m_2 \geqslant \min\left\{ \left\lfloor \frac{n_1 - 1}{v} \right\rfloor + \left\lfloor \frac{n_2 - 1}{v} \right\rfloor + 1; \left\lfloor \frac{r_1 n_1 - 1}{v} \right\rfloor + 1; \left\lfloor \frac{r_2 n_2 - 1}{v} \right\rfloor + 1 \right\} \tag{5.72}$$

is the group of switches used for connections with $\omega > 0.5$.

Proof: Formula (5.71) is obtained by letting $B = 0.5$ in (5.65), and (5.72) is derived from (5.65) by assuming $b > 0.5$. □

In the similar way, the theorem for the discrete bandwidth case can be proved.

It should be noted that the routing strategy with a functional decomposition of middle stage switches results in a reduction of the required switching elements for some values of B and β. For instance, in the discrete bandwidth case, WSN symmetrical networks require less middle stage switches than SNB, for $B = \beta \geqslant 0.75$. In the continuous bandwidth case, this reduction also depends on b and n.

Hwang and Gao [44] proved the wide-sense nonblocking conditions when quota scheme routing strategy is used with $l_1 = l_2 = 0$:

THEOREM 5.39 *$C_{MR}(n, r, \beta, m, 1)$ is wide-sense nonblocking for continuous bandwidth case, when the modified algorithm with the functional division is used, if:*

$$m \geqslant \min\{5.75n, \frac{2(p+1)(Bp + B + p - 1)n}{p^2}\}, \tag{5.73}$$

where numbers of m_1 *and* m_2 *are given by:*

$$(m_1, m_2) = \begin{cases} \left(\dfrac{2(p+1)(Bp+B-1)n}{p^2}, 0 \right), & \text{for } B \leqslant 23/32; \\ (2n-1, 0.75n), & \text{for } B \geqslant 23/32. \end{cases} \tag{5.74}$$

The condition given in Theorem 5.38 provides a little improvement in comparison to those given by Theorem 5.39. For example, for $n = 8$, β=*1*, $B = 0.9$ and $b = 0.1$ (continuous bandwidth case), according to Theorem 5.39 $5.75n = 46$ switches are sufficient, while from Theorem 5.38 it can be calculated that only 44 switches are sufficient.

Some wide-sense nonblocking conditions for k-rate connections model were considered in the literature. The 2-rate model was considered by Tsai, Wang, and Hwang [161, 160], who showed:

THEOREM 5.40 $C_{MR}(n, r, \beta, m, 1)$ *is nonblocking in the wide sense for 2-rate connection model with two rates* B *and* b *satisfying condition* $B + b > 1$ *if:*

$$m \leqslant \min \left\{ \left\lfloor \frac{1}{b} \right\rfloor ; n \right\} + 2n - 3. \tag{5.75}$$

Gao and Hwang [44] considered also wide-sense nonblocking conditions for 2-rate and 3-rate connection models under modified algorithm with functional division of middle stage switches. Some other results for switching fabrics with k-rate connections can be also found in [43, 102].

5.3.3.2 Multicast Connections

Respective theorems for $C_{MR}(n_1, r_1, \beta_1, m, n_2, r_2, \beta_2, v)$ may be derived from $C_{SD}(n_1, r_1, m, n_2, r_2, v)$ and using approach presented for strict-sense nonblocking $C_{MR}(n_1, r_1, \beta_1, m, n_2, r_2, \beta_2, v)$. Switching fabrics with multirate multicast connections under p-limited no-split routing strategy were considered by Zegura [180] and Chan, Chan and Yeung [14, 15]. They derived sufficient conditions using modified algorithm proposed by Yang and Masson [173] for space-division switching fabrics and applying Chung and Ross [22] approach for multirate connections. This upper bound was later improved by Danilewicz and Kabaciński [28]. The new upper bound for p-limited no-split algorithm is given by the following theorem:

THEOREM 5.41 $C_{MR}(n_1, r_1, \beta_1, m, n_2, r_2, \beta_2, v)$ *is wide-sense nonblocking for continuous bandwidth case and* q*-cast connections,* $1 \leqslant q \leqslant q_2 \leqslant r_2$ *under* p*-limited no-split restriction strategy if:*

$$m \geqslant \max_{1 \leqslant j \leqslant q_2} \left\{ \min_{1 \leqslant p \leqslant \min\{j; k(T; n_2; B)\}} \{\min\{A_3; B_3\} + \min\{A_4; B_4\} + 1\} \right\}, \tag{5.76}$$

where

$$\begin{aligned} A_3 &= p \times A_2(n_1;\beta;B), \\ B_3 &= B_2((r_2-p+1);n_2;\beta;B), \\ A_4 &= \max_z \left\{ \begin{pmatrix} z \\ p \end{pmatrix} \leqslant j \times \begin{pmatrix} A_2(n_2;\beta;B) \\ p \end{pmatrix} \right\}, \\ B_4 &= p \times B_2((r_1-1);n_1;\beta;B), \end{aligned}$$

$A_2(n;\beta;B)$ and $B_2(r;n;\beta;B)$ are given by equations (5.66) and (5.67), respectively.

Proof. The proof is based on the proofs of Theorems 5.17 and 5.37. In multirate environment each column of state matrix **C** (see def. 5.25) represents a connection (or a set of connections) which blocks the interstage link. The number of m_2 center stage switches may be derived similarly as in Theorem 5.17, but the number of columns in matrix **C** (i.e., the number of connections which may block interstage links) is equal to

$$A_4 = \max_z \left\{ \begin{pmatrix} z \\ p \end{pmatrix} \leqslant j \times \begin{pmatrix} A_2(n_2;\beta;B) \\ p \end{pmatrix} \right\}, \tag{5.77}$$

where function A_2 is defined by equation (5.66). This number of connections cannot be greater than the number of connections which can be accepted by first stage switches, which is given by $B_4 = p \times B_2((r_1 - 1);n_1;\beta;B)$. Similarly, at the first stage switch there may be at most $A_2(n_1;\beta;B)$ connections, each may blocks links to p center stage switches. Thus $A_3 = p \times A_2(n_1;\beta;B)$ switches will be inaccessible by the new connection $\langle x, \mathbb{Y}, B\rangle$, provided that all these connections can be accepted in $r_2 - p$ third stage switches (we may have up to $B_2(r_2 - p + 1;n_2;\beta;B)$ such connections). □

Similarly as in the space-division switching case, the upper bound given by Theorem 5.41 requires less center stage switches than those given in [14, 15]. The comparison of m for B=0.75 and different r and n is given in Table 5.4. In this table results for different b obtained according to Theorem 5.41 are denoted by p-limited 1, while results obtained in the cited paper are in columns denoted by p-limited 2. It can be seen that for small values of b results in both cases are very similar. For greater b less switches are needed for p-limited 1 estimation. This relation can be also seen from curves presented in Fig. 5.33, where m versus B is plotted in $C_{MR}(8,8,1,m,1)$ for b=0.3, 0.1, and 0.01. In the case of p-limited 2, m is not dependent on b and reaches infinity when B is equal to 1. In Theorem 5.41 m depends on b. When $B = 1$ m is 102, 338, 3380 for b equal to 0.3, 0.1, and 0.01, respectively.

$n=$	10				20			
	p-limited 1			p-lim. 2	p-limited 1			p-lim. 2
	b=0.01	b=0.1	b=0.3		b=0.01	b=0.1	b=0.3	
r=4	144	132	108	145	304	280	228	305
8	174	160	131	175	367	338	276	368
16	199	183	150	201	420	387	315	422
32	223	204	167	225	470	433	352	472
$r=$	10				20			
	b=0.01	b=0.1	b=0.3		b=0.01	b=0.1	b=0.3	
n=4	62	57	47	63	69	63	52	71
8	145	135	109	146	161	149	121	163
16	310	284	232	311	343	315	258	345

Table 5.4. Number of center stage switches in WSNB multirate switching fabrics

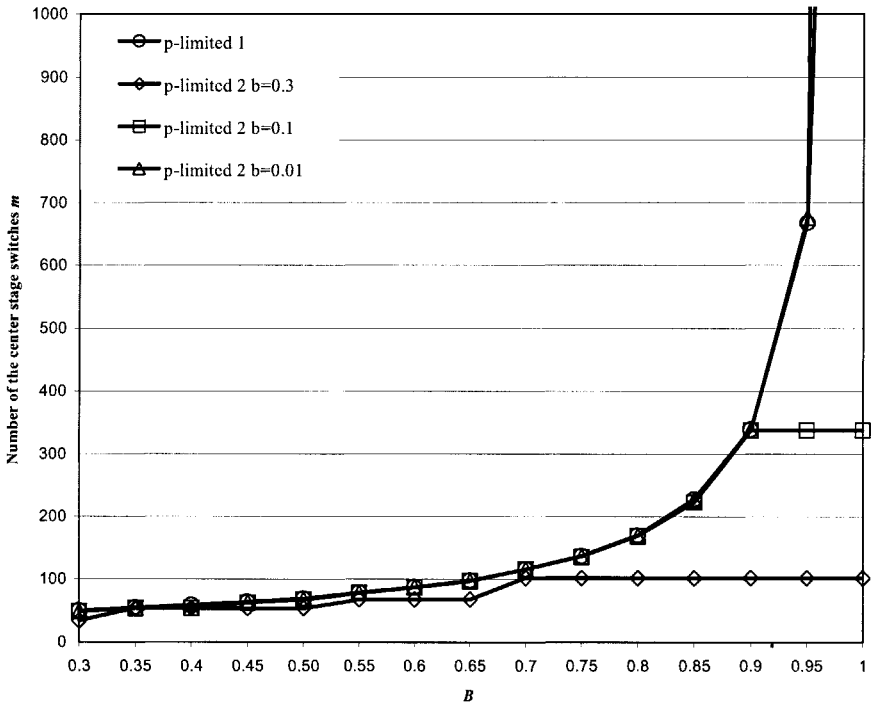


Figure 5.33. The number of center stage switches versus B in $C_{MR}(8,8,1,m,1)$

Kim and Du [87] considered multirate multicast switching fabrics under p-limited no-split restriction and quota scheme routing strategy. They proved the following theorem.

THEOREM 5.42 *$C_{MR}(n, r, \beta, m, 1)$ is wide-sense nonblocking for continuous bandwidth case and q-cast connections, $1 \leqslant q \leqslant q_2 \leqslant r_2$ under p-limited no-split restriction strategy and quota scheme if:*

$$m > \begin{cases} \frac{\beta n(k+1)(Bk+B+k-1)}{k^2} \min\limits_{1 \leqslant p \leqslant q_2} (p + r^{1/p}), & \text{for } B < 23/32 \ ; \\ \left(\frac{15\beta n}{8} + n - 1\right) \min\limits_{1 \leqslant p \leqslant q_2} (p + r^{1/p}), & \text{for } B \geqslant 23/32; \end{cases} \tag{5.78}$$

where $k = \lfloor 1/B \rfloor$.

This condition is better then that proposed in [14, 15], however, condition given in Theorem 5.41 gives better bound for greater b. For instance for $n = r = 8$ and $\beta = B = 1$, we obtain $m = 749$ for any b, while from Theorem 5.41 we have $m = 102$ and 338 for $b = 0.3$ and 0.1, respectively.

5.3.4 Rearrangeable Switching Fabrics

Rearrangeable nonblocking multirate switching fabrics were first considered by Melen and Turner [114]. They also proposed the control algorithm based on graph coloring.

THEOREM 5.43 *$C_{MR}(n, r, \beta, m, 1)$ is rearrangeable nonblocking for continuous bandwidth case if:*

$$m \geqslant \left\lceil \frac{\beta n - B}{1 - B} \right\rceil . \tag{5.79}$$

Proof. In the proof the control algorithm will be given to realize a set of requests in the switching fabric. The control algorithm use bipartite graph representation of connection requests, but now the edges of the bipartite graph are assigned weights representing the bandwidth used by each of the connections. The graph is to be colored with different colors, but at this case edges incident to the same vertex are allowed to have the same color, provided that the total weight of the edges incident to the same vertex and colored with the same color does not exceed the capacity of interstage links. Melen and Turner proposed to convert this weighted graph coloring problem to an ordinary graph coloring problem by splitting each vertex into subsets of vertices. Each subset contains $k = \lceil |\Omega_i|/m \rceil$ sub-vertices, where Ω_i denotes a set of connections in switch I_i. The weights of edges adjacent to the vertex representing I_i are ordered in descending order and m heaviest edges are assigned to one sub-vertex, the next m heaviest edges are assigned to the second sub-vertex, and so forth. The resulting graph is of order m and is m-colorable. The sum of weight of edges colored with the same color is less than $B + (n - B)/m$. Connections represented by this edges can be set

up through the same center stage switch if $1 \geqslant B + (\beta n - B)/m$, i.e., $m \geqslant \lceil (\beta n - B)/(1 - B) \rceil$. □

The example of how connections are set up in $C_{MR}(3, 3, 0.5, 2, 1)$ is shown in Fig 5.34a, b, and c. The set of connections is given in Figure 5.34a. At switch I_1 we have 7 connections which are divided into $\lceil 7/2 \rceil = 4$ subsets, each subset is represented by one node in the bipartite graph. The same is done for connections in switches I_2 and I_3 as well as for switches O_1, O_2, and O_3. The bipartite graph is of order 2 and is 2-colorable. This graph is shown in Fig. 5.34b. One color is marked by dashed line and the second one is denoted by solid line. Connections in the switching fabric are shown in Fig. 5.34c, where connections denoted by dashed line in the bipartite graph are set up through switch 1 of the second stage (this switch is also marked by dashed line). Weight of connections carried by each interstage link are also given. As can be seen from the figure, none of the interstage links has occupancy greater than 1 (i.e., the capacity of the link). The most loaded link is the link between switch M_1 and O_1, whose occupancy is equal to 1.

Du, et al. [36] proposed to use the routing strategy with functional division of center stage switches also in rearrangeable switching fabrics. They proved:

THEOREM 5.44 *$C_{MR}(n, r, \beta, m, 1)$ is rearrangeable nonblocking for continuous bandwidth case if:*

$$m \geqslant 3n - 1, \tag{5.80}$$

where n switches are reserved for routing connections with $\omega > 1/2$, and the remaining $2n - 1$ switches are reserved for connections with $\omega \leqslant 1/2$.

In the same paper author considered also the switching fabric with quota scheme routing and showed that the required number of center stage switches can be further reduced. Liotopoulos [105] consider multirate rearrangeable switching fabrics with split-connection routing, i.e., when one multirate connection can be routed through more than one center stage switch. He also proposed the algorithm based on solving the network flow problem.

Rearrangeable switching fabrics with discrete bandwidth model are also considered in the literature. Chung and Ross [22] conjectured, that three-stage switching fabric with $2n - 1$ center stage switches is rearrangeable. Some cases for k-rate switching fabrics where considered in [102, 103]. Kim and Du [87] gave also conditions for rearrangeability of three-stage switching fabrics with p-limited no-split routing strategy for

$\langle 0, 0, 0.25 \rangle$
$\langle 0, 6, 0.25 \rangle$
$\langle 1, 3, 0.3 \rangle$
$\langle 1, 7, 0.1 \rangle$
$\langle 1, 2, 0.1 \rangle$
$\langle 2, 1, 0.4 \rangle$
$\langle 2, 5, 0.1 \rangle$
$\langle 3, 4, 0.5 \rangle$
$\langle 4, 0, 0.25 \rangle$
$\langle 4, 6, 0.25 \rangle$
$\langle 5, 3, 0.2 \rangle$
$\langle 5, 7, 0.3 \rangle$
$\langle 6, 1, 0.2 \rangle$
$\langle 6, 5, 0.4 \rangle$
$\langle 7, 2, 0.4 \rangle$
$\langle 7, 7, 0.1 \rangle$
$\langle 8, 8, 0.5 \rangle$

Figure 5.34a. Set of connections in $C_{MR}(3, 0.5, 3, 2, 1)$ with $B = 0.5$

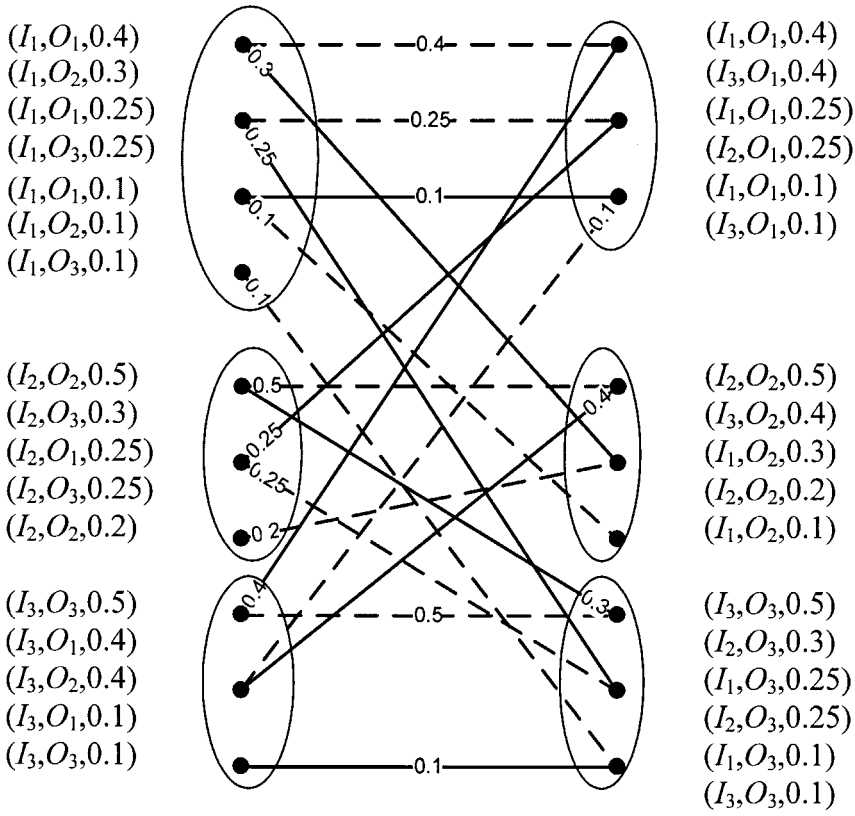


Figure 5.34b. Connection graph

discrete bandwidth case, when the weight of a multicast connection belongs to the set $\{p_1, p_2, \ldots, p_h\}$, where $1 \geqslant p_1 > 1/2 \geqslant p_2 > \cdots > p_h > 0$ and p_i is the integer multiplicity of p_{i-1}, $3 \leqslant i \leqslant h$.

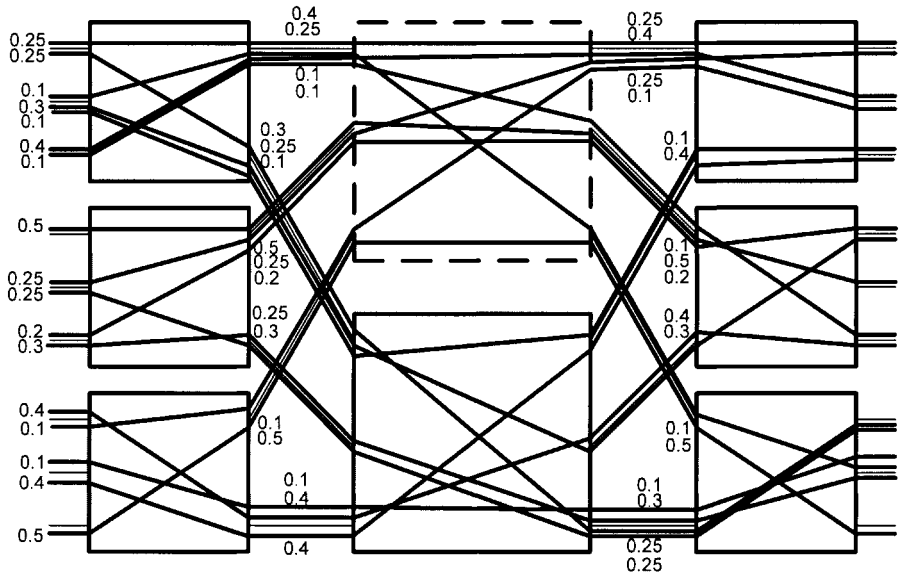


Figure 5.34c. Connections set-up in $C_{MR}(3, 0.5, 3, 2, 1)$ switching fabric

5.3.5 Repackable Switching Fabrics

Repackable time-division switching fabrics proposed by Ohta were extended to multirate case by Liotopoulos and Chalasani [106]. By analogy to the time-division switching, let $L_{M_k}(I_i, O_j)$ denote the total bandwidth occupied by all connections (I_i, O_j) set up through switch M_k; let $M_k^{\max}(I_i, O_j)$ denote the center stage switch for which $L_{M_k}^{\max}(I_i, O_j) = \max_p L_{M_p}(I_i, O_j)$; and let $M_k^{\min}(I_i, O_j)$ denote the center stage switch for which $L_{M_k}^{\min}(I_i, O_j) = \min_p L_{M_p}(I_i, O_j)$. Denote also by Δ a weight which satisfies following conditions:

$$2B \leqslant \Delta < \min\left\{\frac{1-B}{r_2-1}; \frac{1-B}{r_1-1}\right\}. \tag{5.81}$$

The control algorithm can be written as follows.

ALGORITHM 5.16 *RepackingMR*

Step 1 If (I_i, O_j, ω) is a new connection, set this connection through M_k for which $L_{M_k}(I_i, O_j) = L_{M_k}^{\min}(I_i, O_j)$.

Step 2 If (I_i, O_j, ω) is terminated and it is routed through M_k then if there is such switch M_p that $L_{M_k}(I_i, O_j) - \omega < L_{M_p}^{\max}(I_i, O_j) - \Delta$ then

rearrange another connection (I_i, O_j, γ) (it may be one connection or a set of connections of total weight γ), $\gamma \in [\omega, \Delta]$ from switch $M_p^{\max}(I_i, O_j)$ to M_k.

THEOREM 5.45 *$C_{MR}(n_1, r_1, \beta, m, n_2, r_2, \beta, 1)$ is repackable under algorithm RepackingMR if and only if:*

$$m \geqslant \max\left\{\left\lceil \frac{\beta n_1 - (r_2 - 1)\Delta - b}{1 - (r_2 - 1)\Delta - B} \right\rceil ; \left\lceil \frac{\beta n_2 - (r_1 - 1)\Delta - b}{1 - (r_1 - 1)\Delta - B} \right\rceil \right\}. \quad (5.82)$$

In this case the algorithm ensures that connections between switches I_i and O_j are divided approximately equally among center stage switches. It is because before disconnecting (I_i, O_j, ω) from switch M_k the condition $L_{M_k}(I_i, O_j) - \omega < L_{M_p}^{\max}(I_i, O_j) - \Delta$ is fulfilled, and after disconnection the difference in occupancy of M_k and M_p will be greater than Δ, connections from more loaded switch will be moved to less loaded switch.

5.4 Practical Implementations of Clos Networks

When Clos proposed in his paper [23] the three-stage switching fabric architecture, which is now called the Clos switching fabric many researchers considered it as purely theoretical work. At that time implementation of nonblocking switching fabrics of great capacity was costly and impractical. Practical systems were design using blocking switching fabrics composed of smaller number of switches. However, when electronic technology became matured and integrated circuits where available on the market, construction of nonblocking switching fabrics became more economical and in some applications nonblocking switching fabrics are essential. Now three-stage switching fabrics are widely used in switching systems of various kinds. Time-division switching fabrics of this structure are used in telephone exchanges and digital cross-connect systems. When telecommunication network started to migrate towards ATM technology, many considered such architecture as old and out of date. However, when small ATM switches and later on also IP switches were implemented in the network, it become clear that high capacity switches for core network cannot be implemented in one integrated circuit due to the technological constrains and number of pins. Therefore, Clos architecture were taken again into consideration. The same is also true in case of optical switches and optical cross-connect systems which are being introduced in optical transport networks. The attractive feature of the Clos switching fabric is also that it can be easily expanded by adding switches in outer stages, up till full capacity of the switching fabric is reached.

Implementation of the Clos switching fabric was reported in numerous papers. We will show here only some examples of such implementations; this architecture is useful for many applications and in different switching technologies.

One example of using multi-stage Clos architecture is the ATLANTA switch architecture described in [18, 20]. In this architecture switches in the output stages contain buffers to resolve cells output contention, while the center stage switches are bufferless crossbars. This enables preservation of cells sequence integrity even if cells are routed individually through the switching fabric. The switching fabric supports also multicast connections.

One problem in implementing large-capacity switching fabric is interconnection. When a switch used for constructing the switching fabric is implemented in a printed circuit board, a large number of interconnections is required to connect different such boards. An interesting approach to overcome this interconnection problem was proposed in [129]. In this approach optical WDM links with dynamic bandwidth sharing are used for the interconnection between switches.

Three-stage switching fabrics are also considered for optical packet switching. Recently the packet switch architecture called PetaStar was proposed [17]. It uses three-stage photonic bufferless switching fabric. Packet buffering is implemented electronically at the input and output port controllers.

Chapter 6

REPLICATED BASELINE SWITCHING FABRICS

6.1 Switching Fabric Topologies

6.1.1 Vertically Replicated Baseline Switching Fabrics

In section 3.6 banyan-type switching fabrics were considered. The major drawback of these switching fabrics is their blocking characteristics. To obtain nonblocking switching fabrics two methods have been proposed: horizontal cascading (HC) and vertical stacking (VS) [84, 92, 95]. These methods are shown in Fig. 6.1. In the HC method output terminals of one switching fabric are connected one-to-one to respective input terminals of the next switching fabric. Input terminals of the first switching fabric are also input terminals of the whole switching fabric, and the output terminals of the last switching fabric constitute also output terminals of the whole switching fabric. An example of horizontally cascaded 16×16 baseline and 16×16 baseline^{-1} are shown in Fig. 6.2. The HC method results in greater number of stages between each input-output terminal pair. More stages in a switching fabric causes greater signal attenuation in the case of photonic switching, or greater delay in buffered packet switches.

In the VS method p copies of $\log_2 N$ switching fabrics are connected in parallel. Each copy is called a plane. Each input terminal of the switching fabric is connected by $1 \times p$ splitters to respective inputs of each plane. Outputs of each plane are connected by means of $p \times 1$ combiners to output terminals. Vertically stacked switching fabrics are called multi-$\log_2 N$ or $\log_2(N, 0, p)$ switching fabrics. The $\log_2(8, 0, 3)$ switching fabric is shown in Fig. 6.3 as an example. Input and output terminals of the $\log_2(N, 0, p)$ switching fabric are numbered $0, 1, \ldots, N-1$, from

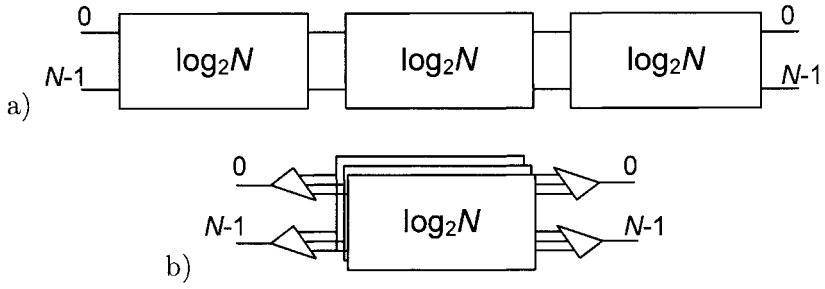


Figure 6.1. Horizontal cascading (a) and vertical stacking (b) methods

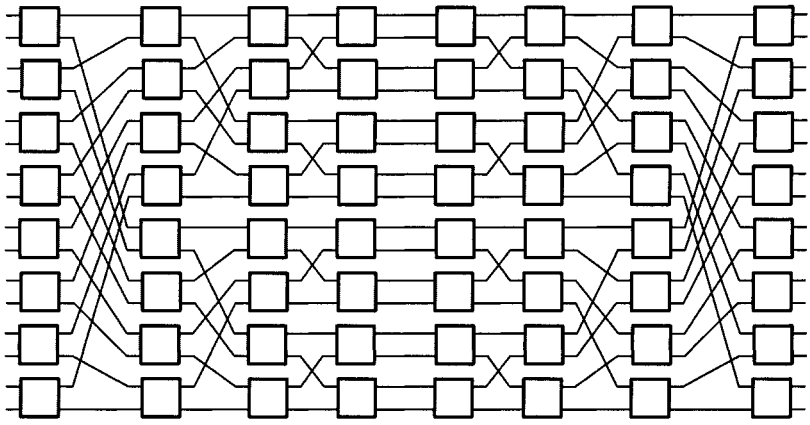


Figure 6.2. Horizontally cascaded 16×16 baseline and baseline^{-1} switching fabrics

top to bottom, respectively, and stages in each plane are respectively numbered $1, 2, \ldots n$, from left to right. Planes are numbered from 1 to p. The bipartite graph considered in section 3.6 will be used to represent each plane of the $\log_2(N, 0, p)$ switching fabric.

6.1.2 Extended Baseline and Vertically Replicated Extended Baseline Switching Fabrics

An extended banyan-type switching fabric is obtained by adding m extra stages, $1 \leqslant m \leqslant n-1$, to a banyan-type switching fabric. Different interconnection patterns may be used to interconnect these additional stages. They may used the same interconnection patterns as first m stages of the main switching fabric, last m stages, or mirror images of these stages, as it is shown for example in Figures 6.4a, 6.4b, 6.4c, and

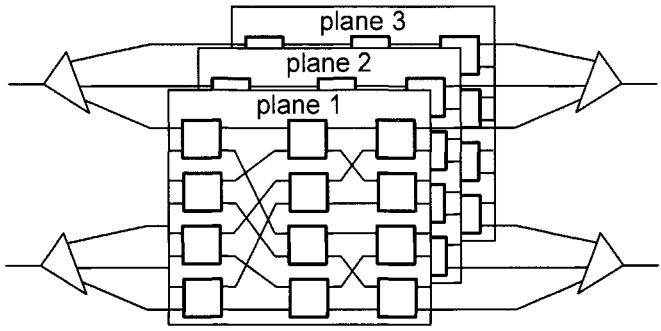


Figure 6.3. The $\log_2(8,0,3)$ switching fabric

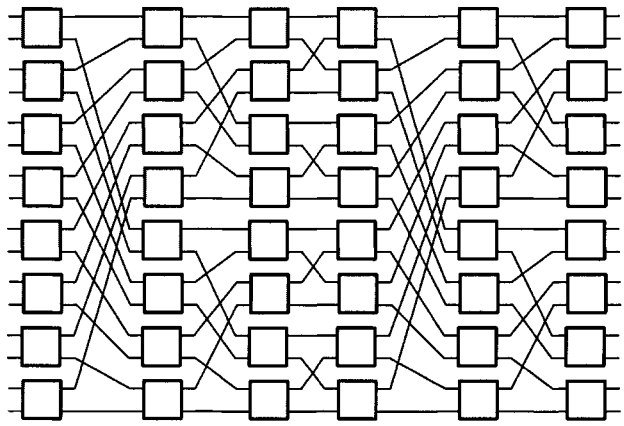


Figure 6.4a. Extended baseline switching fabric with two additional stages which uses the same interconnection pattern as first two stages

6.4d , respectively, for the 16×16 baseline switching fabric. The topology with m extra stages being mirror image of the first m stages is referred to as the $\log_2(N,m,1)$ switching fabric, and this topology will be later considered in this chapter. Similarly as in $\log_2(N,0,1)$ switching fabric, input terminals are numbered $0,1,\ldots,N-1$, from top to bottom, and stages are numbered $1,2,\ldots,n+m$, from left to right.

The bipartite graph will be also used as representation of the $\log_2(N,m,1)$ switching fabric. This graph contains $n+m+1$ stages of nodes numbered $0,1,\ldots,n+m$, from left to right. Nodes in stages 1 and $n+m-1$ constitutes the first shell; nodes in stages 2 and $n+m-2$ forms the second shell. In general, stages i and $n+m-i$ are collec-

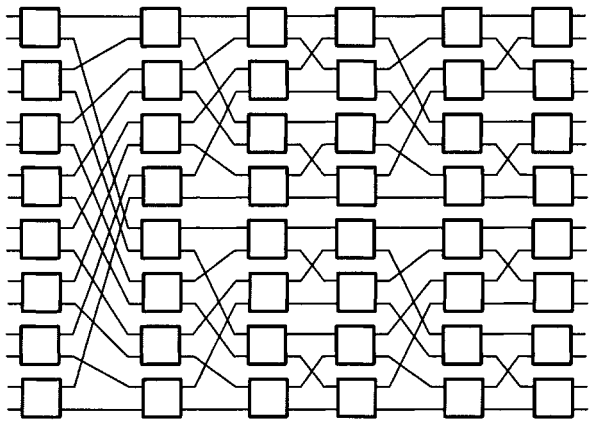


Figure 6.4b. Extended baseline switching fabric with two additional stages which uses the same interconnection pattern as last two stages

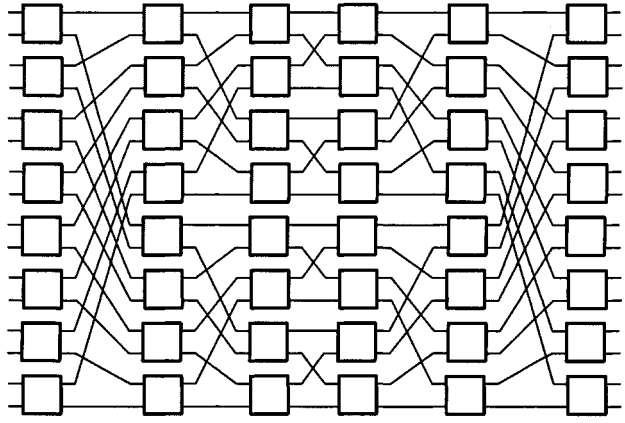


Figure 6.4c. Extended baseline switching fabric with two additional stages which uses the mirror image of interconnection pattern in first two stages

tively called shell i, stage i is called the left part of the shell and stage $n+m-i$ is called the right part of the shell. When $n+m$ is odd, there are $(n+m-1)/2 = \lfloor (n+m)/2 \rfloor$ shells, each containing two stages. For $n+m$ even, we have $[(n+m)/2]-1$ shells with two stages, and one center shell which contains nodes of only one stage numbered $(n+m)/2$. The bipartite graph representation of the $\log_2(16,2,1)$ switching fabric and its shells are shown in Fig. 6.5. All possible connecting paths between input terminal 0 and output terminal 0 are shown in bold lines.

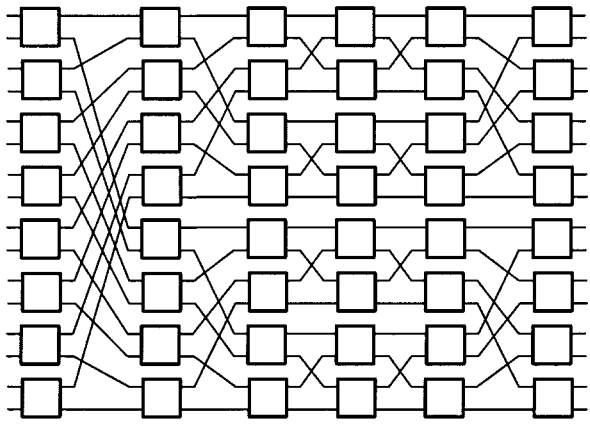


Figure 6.4d. Extended baseline switching fabric with two additional stages which uses the mirror image of interconnection pattern in last two stages

In general, there are 2^m possible connecting paths for any input-output terminal pairs. The channel graph of the $\log_2(N, m, 1)$ switching fabric is shown in Fig. 6.6. The number of paths passing through nodes in different stages in not identical, as can be seen in the channel graph. The connecting path between input terminal x and output terminal y can use one of two nodes in stages 1 and $n+m-1$, four nodes in stages 2 and $n+m-2$, and so forth. However, in stage i, $m \leqslant i \leqslant n$ only one of 2^m nodes can be used by the connection. Let us use the following notation:

DEFINITION 6.1 *Let $\mathbb{NP}_i$ denotes a set of connecting paths which can go through the nodes of shell i. The cardinality of $\mathbb{NP}_i$ is given by*

$$|\mathbb{NP}_i| = \begin{cases} 2^i & \text{for} \quad 1 \leqslant i < m \\ 2^m & \text{for} \quad m \leqslant i \leqslant \left\lceil \dfrac{n+m}{2} \right\rceil \end{cases} . \tag{6.1}$$

The $|\mathbb{NP}_i|$ denotes the number of paths passing through each stage of shell i. Sets $\mathbb{SI}_i$ and $\mathbb{SO}_j$ (Def. 3.2 and 3.3) defined for $\log_2(N, 0, 1)$ switching fabrics have to be rewritten in case of $\log_2(N, m, 1)$ switching fabrics in the following way:

DEFINITION 6.2 *Let $\mathbb{SI}_j^m$ be the set of these input terminals (excluding input terminal x and all $\mathbb{SI}_i^m$, where $1 \leqslant i < j$, $1 \leqslant j \leqslant n$) whose paths can intersect with path $\langle x, y \rangle$ in a node of stage j. For $n < j \leqslant n+m-1$ there are no additional paths which can intersect with path $\langle x, y \rangle$ but from a node of stage j all input terminals are available. Input terminals*

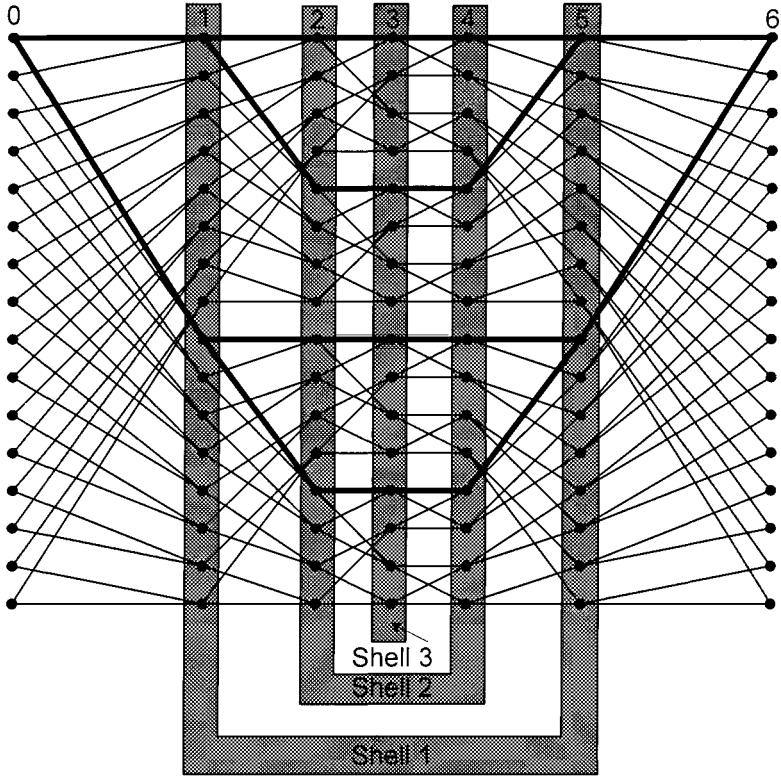


Figure 6.5. The bipartite graph representation of the $\log_2(16, 2, 1)$ switching fabric

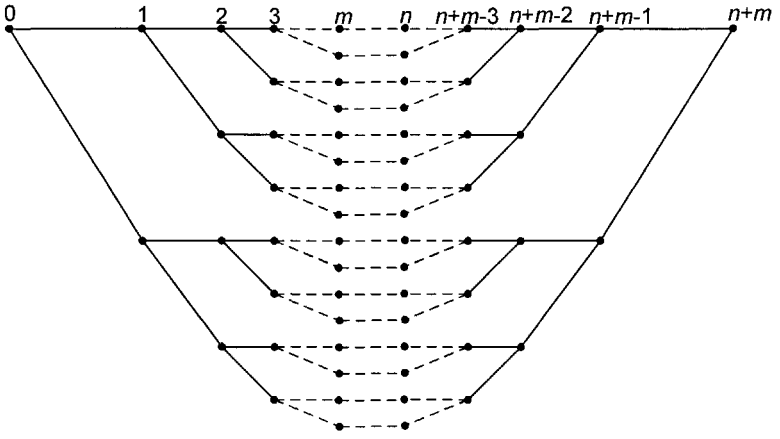


Figure 6.6. The channel graph of the $\log_2(N, m, 1)$ switching fabric

belonging to set $\mathbb{SI}_j^m$ *will be called accessible from stage* j. *The cardinality of* $\mathbb{SI}_j^m$ *is given by*

$$\left|\mathbb{SI}_j^m\right| = \begin{cases} 2^{j-1} & \text{for } 1 \leqslant j \leqslant n \\ 2^n & \text{for } n < j \leqslant m+n-1 \end{cases} . \tag{6.2}$$

DEFINITION 6.3 *Let* $\mathbb{SO}_j^m$ *be the set of those output terminals (excluding output terminal* y *and all* $\mathbb{SO}_i^m$, *where* $j < i \leqslant m+n-1$, $m \leqslant j \leqslant m+n-1$*) whose paths can intersect with path* $\langle x, y\rangle$ *in a node of stage* j. *For* $1 \leqslant j < m$ *there are no additional paths which can intersect with path* $\langle x, y\rangle$ *but from a node of stage* j *all output terminals are available. Output terminals belonging to set* $\mathbb{SO}_j^m$ *will be called accessible from stage* j. *The cardinality of* $\mathbb{SO}_i^m$ *is given by*

$$\left|\mathbb{SO}_j^m\right| = \begin{cases} 2^{m+n-j-1} & \text{for } m \leqslant j \leqslant m+n-1 \\ 2^n & \text{for } 1 < j \leqslant m \end{cases} . \tag{6.3}$$

Vertically replicated extended banyan-type switching fabrics are obtained by vertically stacking p copies of $\log_2(N, m, 1)$ switching fabrics, and are denoted by $\log_2(N, m, p)$. When the switching fabric is composed, in general, of $d \times d$ switches respective topologies are denoted by $\log_d(N, 0, 1)$, $\log_d(N, 0, p)$, $\log_d(N, m, 1)$, and $\log_d(N, m, p)$.

6.2 Control Algorithms

When a new connection is to be set up, a plane is to be chosen to realize the appropriate connecting path. When in one plane there is more than one connecting path possible, one of these paths has also to be selected. Algorithms which can be used here are similar to those considered for three-stage switching fabrics. The difference is that instead of looking for a middle stage switch we look for a plane. Respective algorithms can be defined as follows:

ALGORITHM 6.1 *Random (RAN)*
Check planes randomly and set up the connection through the first available plane.

ALGORITHM 6.2 *Sequential (SEQ)*
Check planes sequentially starting form plane i, $1 \leqslant i \leqslant p$ and choose the first available plane.

ALGORITHM 6.3 *Minimum index (MINIX)*
This algorithm is the same as sequential, but $i = 1$.

ALGORITHM 6.4 *Quasi-random (Q-RAN)*

This algorithm is the same as sequential, but $k = l + 1$, $i = P + 1$, where P denotes the plane used to route the last request ($i = 1$ for $P = p$).

ALGORITHM 6.5 *Cyclic static (CS)*
This algorithm is the same as Q-RAN, but $i = P$, i.e., we start to check planes from the the plane used to route the last request.

ALGORITHM 6.6 *Save the unused (STU)*
Do not route a new connection through any empty plane unless there is no choice.

ALGORITHM 6.7 *Packing (PACK)*
Route a new connection through the busiest but available plane.

When all planes have been checked and no plane was found, the connection is blocked. These algorithms may be used for both unicast and multicast connections. For multicast connections, paths belonging to the given multicast connection can be set up using different routing strategies. In one strategy all paths of a multicast connection have to be set up through the same plane. In another strategy each path of a multicast connection can be set up independently of other paths of this connection. A multicast connection can be also set up using strategies based on the concept called a blocking window [163, 164]. The blocking window is defined as follows.

DEFINITION 6.4 *Let a set of output terminals* $\mathbb{O} = \{0, 1, \ldots, N-1\}$ *be divided into* N/K *subsets* $\mathbb{BW}_i = \{K \times i, K \times i + 1, \ldots, K \times i + (K-1)\}$, *where* $K = 2^t$, $i = 0, 1, \ldots, (N/K) - 1$ *and* $1 \leqslant t \leqslant n$. *Each subset* $\mathbb{BW}_i$ *is called the blocking window.*

Examples of blocking windows in the $\log_2(32, 0, 1)$ switching fabric are shown in Fig. 6.7. For instance, when $t = 3$, output terminals are divided into four blocking windows, each of which contains eight output terminals: $\mathbb{BW}_0 = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$, $\mathbb{BW}_1 = \{9, 10, 11, 12, 13, 14, 15\}$, $\mathbb{BW}_2 = \{16, 17, 18, 19, 20, 21, ,22, 23\}$, and $\mathbb{BW}_3 = \{24, 25, 26, 27, 28, 29, 30, 32\}$. For $t = n$ there is only one blocking window containing all output terminals of the switching network. An example of multicast connection $\langle 0, \{1, 4, 12, 18, 30\}\rangle$ is also shown in Fig. 6.7. For $t = 3$ output terminals 1 and 4 belong to the same blocking window and other output terminals of this connection are in different blocking windows.

Blocking windows will be used in the control algorithm for setting up multicast connections. Let $\langle x, \mathbb{Y}\rangle$ be a new connection. The set of outputs $\mathbb{Y}$ can be divided into subsets $\mathbb{Y}_i$, $\mathbb{Y}_i = \{y : y \in \mathbb{BW}_i, i =$

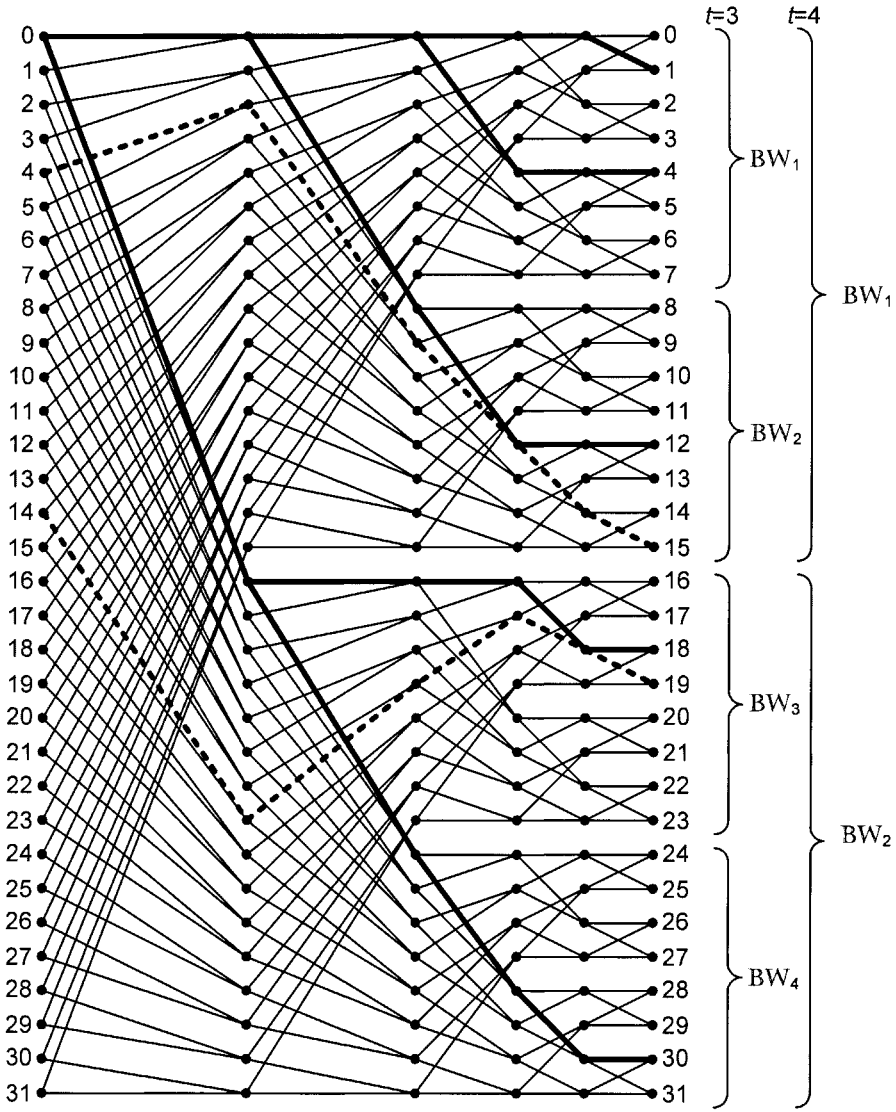


Figure 6.7. Blocking windows in $\log_2(32, 0, 1)$

$0, 1, \ldots, (N/K) - 1\}$. Connections $\langle x, \mathbb{Y}_i \rangle$ are called subconnections of connection $\langle x, \mathbb{Y} \rangle$. For instance, connection $\langle 0, \{1, 4, 12, 18, 30\} \rangle$ considered earlier can be divided into subconnections $\langle 0, \{1, 4\} \rangle$, $\langle 0, \{12\} \rangle$, $\langle 0, \{18\} \rangle$, and $\langle 0, \{30\} \rangle$ if $t = 3$ or subconnections $\langle 0, \{1, 4, 12\} \rangle$ and $\langle 0, \{18, 30\} \rangle$ if $t = 4$. Each of subonnections $\langle x, \mathbb{Y}_i \rangle$ is to be set up through one plane of the $\log_2(N, 0, p)$ switching fabric. It should be

noted that if a new connection is a point-to-point connection, then it has one subconnection. The *blocking window control algorithm* can be described as follows [32].

ALGORITHM 6.8 *Blocking window control algorithm*

Step 1 Divide a new connection $\langle x, \mathbb{Y} \rangle$ into subconnections $\langle x, \mathbb{Y}_i \rangle$. Choose one of the subconnections.

Step 2 Choose an arbitrary plane of already occupied planes (i.e., one plane through which at least one other connection is already set up). If this plane is available for the subconnection, set up this subconnection through this plane. If not, check other already occupied planes.

Step 3 If all already occupied planes are not available for the subconnection, set it up through a free plane.

Step 4 Choose the next subconnection and go to Step 2. Repeat steps 2 - 4 up until all subconnections of connection $\langle x, \mathbb{Y} \rangle$ are set up.

In this algorithm the save the unused algorithm is used to setting up subconnections. However, any of earlier given algorithm can be used for subconnections.

Let us consider the switching fabric of Fig. 6.7, and let $t = 3$. In this network connections $\langle 4, \{15\} \rangle$ and $\langle 14, \{19\} \rangle$ (marked by dashed lines) are already set up through the first plane. The new connection $\langle 0, \{1, 4, 12, 18, 30\} \rangle$ will be divided into subconnections $\langle 0, \{1, 4\} \rangle$, $\langle 0, \{12\} \rangle$, $\langle 0, \{18\} \rangle$, and $\langle 0, \{30\} \rangle$ (Step 1). We choose the first subconnection $\langle 0, \{1, 4\} \rangle$ and check if it can be set up through the first plane. Since connections $\langle 4, \{15\} \rangle$ and $\langle 14, \{19\} \rangle$ do not block the subconnection considered, it can be set up through the first plane. The second subconnection $\langle 0, \{12\} \rangle$ cannot be set up through the first plane since it is blocked by connection $\langle 4, \{15\} \rangle$. We have to choose the next plane for this subconnection. Similarly, subconnection $\langle 0, \{18\} \rangle$ cannot be set up through the first plane (connection $\langle 14, \{19\} \rangle$), but it can be added to connection $\langle 0, \{12\} \rangle$ in the second plane, since both belong to the same multicast connection. Finally, subconnection $\langle 0, \{30\} \rangle$ can be set up either through the first or the second plane.

In switching fabrics with extra stages more routes can be used in one plane. Thus connecting paths to output terminals in the same blocking window may follow different routes. Therefore, in $\log_2(N, m, p)$ switching fabrics we may consider two possible routing strategies in the blocking window control algorithm:

- *Duplication routing*: A multicast connection may use two or more nodes in each of stages up to $n + m - t$.

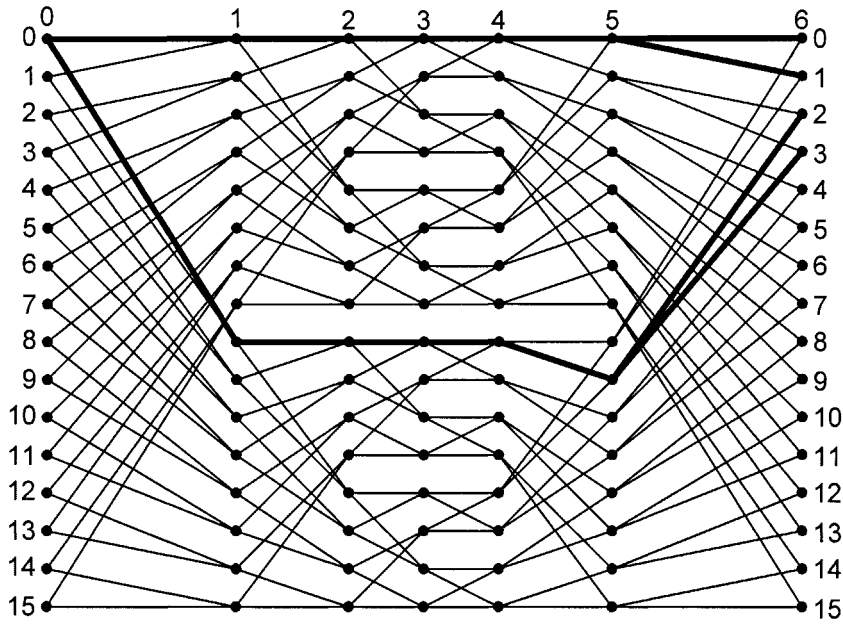


Figure 6.8a. Duplication routing in $\log_2(16, 2, 1)$ switching fabric

- *Non-duplication routing*: A multicast connection may use only one node in each of stages up to $n + m - t$.

Examples of duplication and non-duplication routing in $\log_2(16, 2, 1)$ switching fabric are shown in Figures 6.8a and 6.8b, respectively. It is assumed, that $t = 2$ and connection $\langle 0, \{1, 2, 3, 4\}\rangle$ is to be set up. In duplication routing the connection uses two nodes in stages from 1 to $n + m - t = 4$. In non-duplication routing only one node is used by this connection in each of stages from 1 to 4. Further on it is assumed, that non-duplication routing is used in switching fabrics.

6.3 Space-division Switching

6.3.1 Strict-sense Nonblocking Conditions

6.3.1.1 Unicast Connections

In $\log_2(N, 0, p)$ and $\log_2(N, m, p)$ switching fabrics the question is how many planes are to be connected in parallel to ensure nonblocking operation. In the case of space-division switching fabrics and point-to-point connections, the number of copies needed was given in [84, 92, 150] for $\log_2(N, 0, p)$ and $\log_2(N, m, p)$ switching fabrics, respectively.

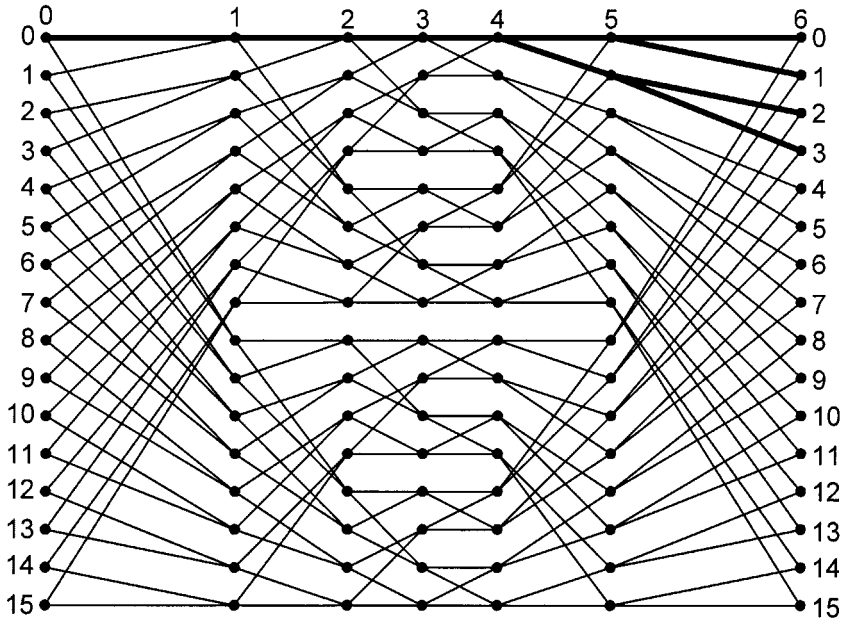


Figure 6.8b. Non-duplication routing in $\log_2(16, 2, 1)$ switching fabric

THEOREM 6.5 *The* $\log_2(N, 0, p)$ *switching fabric is nonblocking in the strict sense if and only if:*

$$p \geqslant \begin{cases} \frac{3}{2}2^{\frac{n}{2}} - 1 & \text{for } n \text{ even}, \\ 2^{\frac{n+1}{2}} - 1 & \text{for } n \text{ odd}. \end{cases} \tag{6.4}$$

Proof. Sufficiency. Let consider the new connection $\langle x, y \rangle$. At the node of stage i, $1 \leqslant i \leqslant \lfloor n/2 \rfloor$, the connecting path of this connection may be blocked by connecting paths from nodes in $\mathbb{SI}_i$. On the other hand, the connecting path may be also blocked in the node of stage $n - i$ by connecting paths to nodes in $\mathbb{SO}_{n-i}$. For n odd a set of p_1 planes may be inaccessible to the new connection because of connecting path from $\mathbb{SI}_i$, where

$$p_1 = \sum_{i=1}^{\lfloor n/2 \rfloor} |\mathbb{SI}_i| = \sum_{i=1}^{\lfloor n/2 \rfloor} 2^{i-1} = 2^{\lfloor \frac{n}{2} \rfloor} - 1, \tag{6.5}$$

and another set of p_2 planes may be blocked because of connections to $\mathbb{SO}_{n-i}$, where

$$p_2 = \sum_{i=1}^{\lfloor n/2 \rfloor} |\mathbb{SO}_{n-i}| = \sum_{i=1}^{\lfloor n/2 \rfloor} 2^{i-1} = 2^{\lfloor \frac{n}{2} \rfloor} - 1. \tag{6.6}$$

In the worst case these sets of planes are disjoint and one more plane is needed for the new connection, thus

$$p \geqslant p_1 + p_2 + 1 = 2\left(2^{\lfloor \frac{n}{2} \rfloor} - 1\right) + 1 = 2^{\frac{n+1}{2}} - 1. \tag{6.7}$$

When n is even connections from nodes in $\mathbb{SI}_{n/2}$ is to be connected with nodes in $\mathbb{SO}_{n/2}$ to block connection $\langle x, y \rangle$. Therefore, stage $n/2$ must be counted only once, so we have:

$$p_1 = \sum_{i=1}^{\lfloor n/2 \rfloor} |\mathbb{SI}_i| = \sum_{i=1}^{\lfloor n/2 \rfloor} 2^{i-1} = 2^{\lfloor \frac{n}{2} \rfloor} - 1, \tag{6.8}$$

$$p_2 = \sum_{i=1}^{\lfloor n/2 \rfloor - 1} |\mathbb{SO}_{n-i}| = \sum_{i=1}^{\lfloor n/2 \rfloor - 1} 2^{i-1} = 2^{\lfloor \frac{n}{2} \rfloor - 1} - 1, \tag{6.9}$$

and

$$\begin{aligned} p \geqslant p_1 + p_2 + 1 &= \left(2^{\lfloor \frac{n}{2} \rfloor} - 1\right) + \left(2^{\lfloor \frac{n}{2} \rfloor - 1} - 1\right) + 1 = \\ &= \frac{3}{2} 2^{\frac{n}{2}} - 1. \end{aligned} \tag{6.10}$$

Necessity can be proved by showing for any path searching algorithm a set of events leading to the occupancy of all planes. It is very easy to find an example of such set for quasi-random algorithm. □

THEOREM 6.6 *The* $\log_2(N, m, p)$ *switching fabric is nonblocking in the strict sense if and only if:*

$$p \geqslant \begin{cases} m + 3 \cdot 2^{\frac{n-m}{2} - 1} - 1 & \text{for } n + m \text{ even}, \\ m + 2^{\frac{n-m+1}{2}} - 1 & \text{for } n + m \text{ odd}. \end{cases} \tag{6.11}$$

Proof. The proof is similar to Theorem 6.5, but at stage i connections must block $\mathbb{NP}_i$ paths in each of planes. Thus, for $n + m$ odd we have

$$\begin{aligned} p_1 = \sum_{i=1}^{\lfloor (n+m)/2 \rfloor} \frac{|\mathbb{SI}_i|}{|\mathbb{NP}_i|} &= \sum_{i=1}^{m} \frac{2^{i-1}}{2^i} + \sum_{i=m+1}^{\lfloor (n+m)/2 \rfloor} \frac{2^{i-1}}{2^m} \\ &= \frac{m}{2} + 2^{\frac{n-m-1}{2}} - 1, \end{aligned} \tag{6.12}$$

$$p_2 = \sum_{i=1}^{\lfloor (n+m)/2 \rfloor} \frac{|\mathbb{SO}_{n-i}|}{|\mathbb{NP}_{n-i}|} = \sum_{i=1}^{m} \frac{2^{i-1}}{2^i} + \sum_{i=m+1}^{\lfloor (n+m)/2 \rfloor} \frac{2^{i-1}}{2^m}$$

$$= \frac{m}{2} + 2^{\frac{n-m-1}{2}} - 1, \tag{6.13}$$

and

$$p \geqslant p_1 + p_2 + 1 = 2\left(\frac{m}{2} + 2^{\frac{n-m-1}{2}} - 1\right) + 1 =$$

$$= m + 2^{\frac{n-m+1}{2}} - 1. \tag{6.14}$$

For $n + m$ even p_1 is the same as for $n + m$ odd,

$$p_2 = \sum_{i=1}^{\lfloor (n+m)/2 \rfloor - 1} \frac{|\mathbb{SO}_{n-i}|}{|\mathbb{NP}_{n-i}|} = \sum_{i=1}^{m} \frac{2^{i-1}}{2^i} + \sum_{i=m+1}^{\lfloor (n+m)/2 \rfloor - 1} \frac{2^{i-1}}{2^m}$$

$$= \frac{m}{2} + 2^{\frac{n-m}{2}-1} - 1, \tag{6.15}$$

and

$$p \geqslant p_1 + p_2 + 1 = \left(\frac{m}{2} + 2^{\frac{n-m-1}{2}} - 1\right) + \left(\frac{m}{2} + 2^{\frac{n-m}{2}-1} - 1\right) + 1$$

$$= m + 3 \cdot 2^{\frac{n-m}{2}-1} - 1. \tag{6.16}$$

These conditions are also necessary. Again the set of events for quasi-random routing is very easy to find. □

Hwang generalized these results to $\log_d(N, m, p)$ switching fabrics [55]

THEOREM 6.7 *The $\log_d(N, m, p)$ switching fabric is nonblocking in the strict sense if and only if:*

$$p \geqslant \begin{cases} \frac{2m(d-1)}{d} + (d+1) \cdot d^{\frac{n-m}{2}-1} - 1 & \text{for } n+m \text{ even}, \\ \frac{2m(d-1)}{d} + 2d^{\frac{n-m-1}{2}} - 1 & \text{for } n+m \text{ odd}. \end{cases} \tag{6.17}$$

The number of planes needed in the strict-sense nonblocking $\log_2(N, m, p)$ switching fabric depends on m and n. Comparison of p versus m and n is given in Table 6.1. It can be seen that when m is growing for given n, the required number of planes is getting smaller till some value of m, and then it does not change. For the minimum value of m, for which the number of planes reaches minimum, the switching fabric requires also the minimum number of 2×2 switches. The number of switches, including 2×2 switches in splitters and combiners at the input and output of the switching fabric, is compared in Table 6.2.

$m=$	0	1	2	3	4	5	6	7	8	9	10	11
$n=2$	[2]	2										
3	[3]	3	3									
4	5	[4]	4	4								
5	7	6	[5]	5	5							
6	11	8	7	[6]	6	6						
7	15	12	9	8	[7]	7	7					
8	23	16	13	10	9	[8]	8	8				
9	31	24	17	14	11	10	[9]	9	9			
10	47	32	25	18	15	12	11	[10]	10	10		
11	63	48	33	26	19	16	13	12	[11]	11	11	
12	95	64	49	34	27	20	17	14	13	[12]	12	12

Table 6.1. The number of planes as a function of n and m in strict-sense nonblocking $\log_2(N, m, p)$ switching fabrics with unicast connections

$m=$	0	1	2	3	4	5	6	7
$n=2$	[16]	20						
3	[68]	80	92					
4	288	[256]	288	320				
5	944	896	[816]	896	976			
6	3392	2688	2560	[2368]	2560	2752		
7	10304	8960	7232	6912	[6464]	6912	7360	
8	34816	26112	22784	18688	17920	[16896]	17920	18944

Table 6.2. The number of 2×2 switches as a function of n and m in strict-sense nonblocking $\log_2(N, m, p)$ switching fabrics with unicast connections

6.3.1.2 Multicast Connections

Strict-sense nonblocking multicast $\log_2(N, 0, p)$ switching fabrics were first considered by Danilewicz and Kabaciński [26, 32]. They derived appropriate conditions from wide-sense nonblocking switching fabrics by finding a maximum number of planes for the blocking window algorithm with different window sizes. It turns out that the maximum number of planes is needed when a multicast connection is to be routed through one plane.

THEOREM 6.8 *The* $\log_2(N, 0, p)$ *switching fabric is nonblocking in the strict sense if and only if:*

$$p \geqslant \frac{N}{2}. \tag{6.18}$$

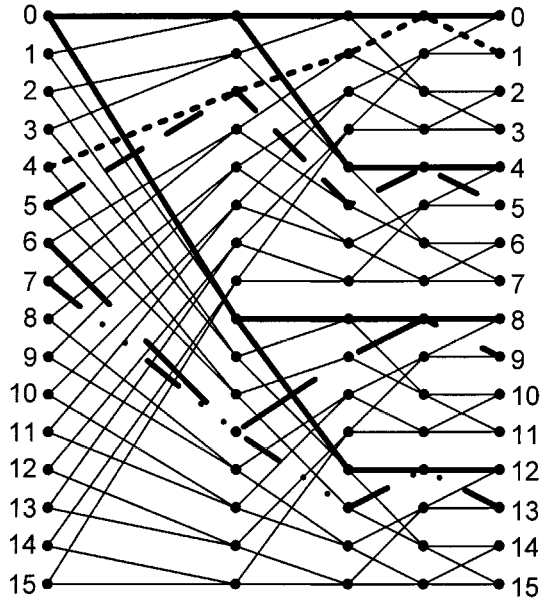


Figure 6.9a. Planes 1 to 4 are blocked to connection $\langle 0, \{0, 4, 8, 12\}\rangle$

Proof. Sufficiency. Let connection $\langle x, \mathbb{Y}\rangle$ be set up, where $|\mathbb{Y}| = |\mathbb{SI}_{n-1}|$. This connection may be blocked by point-to-point connections from input terminals in $\mathbb{SI}_i$, $1 \leqslant i \leqslant n-1$, and one more plane is needed for the new connection. We have:

$$p \geqslant \sum_{i=1}^{n-1} |\mathbb{SI}_i| + 1 = \sum_{i=1}^{n-1} 2^i + 1 = 2^{n-1}. \tag{6.19}$$

Necessity. The set of events leading to the occupancy of all planes given by equation (6.18) can be easily constructed using quasi-random algorithm. □

An example of the worst state in the $\log_2(16, 0, 8)$ switching fabric is shown in Fig. 6.9a and 6.9b. The new connection is $\langle 0, \{0, 4, 8, 12\}\rangle$. In the first plane this connection is blocked by connection $\langle 4, 1\rangle$. Connections $\langle 5, 5\rangle$, $\langle 6, 9\rangle$, and $\langle 7, 13\rangle$ make planes 2, 3 and 4 unavailable by the new connection, respectively (Fig. 6.9a). Connections $\langle 2, 2\rangle$, $\langle 3, 10\rangle$, and $\langle 1, 6\rangle$ block the new connection in planes 5, 6 and 7 (Fig. 6.9b), so plane 8 is needed to set connection $\langle 0, \{0, 4, 8, 12\}\rangle$ (marked in bold lines in both Figures).

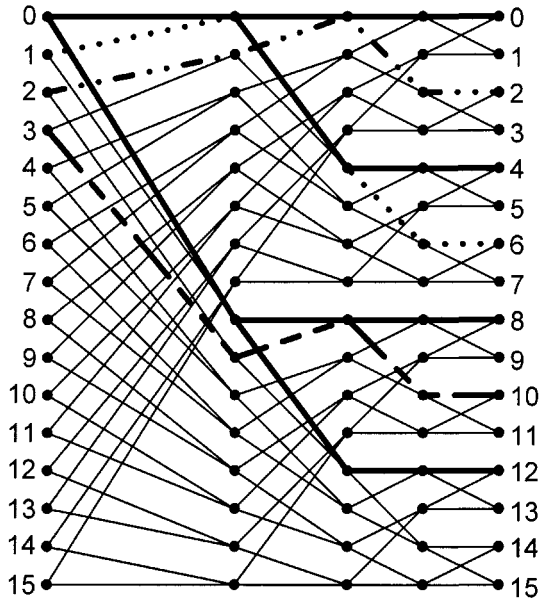


Figure 6.9b. Planes 5 to 7 are blocked to connection $\langle 0, \{0, 4, 8, 12\}\rangle$

6.3.2 Wide-sense Nonblocking Conditions

The wide-sense nonblocking conditions for $\log_2(N, 0, p)$ switching fabrics in case of unicast connections were considered in [78]. They conclude, that non of the algorithms given in section 6.2 results in lower number of planes required for nonblocking operation. More researches were done about wide-sense nonblocking operation of these switching fabrics for multicast connections. Nonblocking multicast $\log_2(N, 0, p)$ switching fabrics were first considered by Tscha and Lee [163], but result given in the cited paper constituted the lower bound of nonblocking operation. This result was improved in [164]. They used the blocking window algorithm, where the blocking window contained $2^{\lfloor n/2 \rfloor}$ outputs. A more general approach with different blocking window sizes was considered in papers [32, 29] and the thesis [26]. Before we move to the theorem terms *maximum blocking configuration* and *maximum multicast connection* have to be defined, since they will be used later on in proofs.

DEFINITION 6.9 $\mathbb{MBC}(y, n-j) = \{\langle x_k, y_k\rangle : x_k \in \mathbb{SI}_{n-\lfloor \log_2 k \rfloor - 1};\ y,\ y_k \in \mathbb{BW}_i;\ y_k \in \mathbb{SO}_{n-\lfloor \log_2 k \rfloor - 1};$ *and* $\left|\mathbb{V}_{\langle x,y\rangle} \cap \mathbb{V}_{\langle x_k, y_k\rangle}\right| = 1\}$ *where* $\mathbb{V}_{\langle x,y\rangle}$ *denotes a set of nodes in connecting path* $\langle x, y\rangle$, $1 \leqslant k \leqslant 2^{j-1}$.

In this definition $\mathbb{MBC}(y, n-j)$ denotes a maximum blocking configuration at stage $n-j$ for output y. This $\mathbb{MBC}(y, n-j)$ is a set of all possible connecting paths $\langle x_k, y_k \rangle$ which may block connecting path $\langle x, y \rangle$ in nodes of stages from $n-1$ to $n-j$. In stage $n-1$ this connection may be blocked by connections to $\mathbb{SO}_{n-1}$. Similarly, in stage $n-2$ connections to $\mathbb{SO}_{n-2}$ may block the considered connection. Finally, in stage $n-j$, $j \leqslant \lfloor n/2 \rfloor$ connections to $\mathbb{SO}_{n-j}$ will also block connection $\langle x, y \rangle$. In the worst case these connections may be set up through different planes. Thus, $\mathbb{MBC}(y, n-j)$ may occupy $\sum_{i=1}^{n-j} 2^{n-i-1} = 2^j - 1$ planes, and these planes will be inaccessible by connection $\langle x, y \rangle$. In the example of Fig. 6.10 the $\mathbb{MBC}(0,3) = \{\langle 8,1 \rangle, \langle 4,2 \rangle, \langle 6,3 \rangle\}$.

DEFINITION 6.10 *$\mathcal{MMC}(x, i, f_i, t)$ is a connection $\langle x, \mathbb{Y} \rangle$, where $x \in \mathbb{SI}_i$, $|\mathbb{Y}| = f_i \leqslant 2^{n-t-i}$, $\mathbb{Y} = \{y_j : y_j \in \mathbb{BW}_j, 0 \leqslant j \leqslant 2^{n-t-i} - 1\}$, and for $0 \leqslant j \leqslant 2^{n-t} - 1$ $|\mathbb{Y} \cap \mathbb{BW}_j| \leqslant 1$.*

The $\mathcal{MMC}(x, i, f_i, t)$ denotes a multicast connection, in which each output terminal in this connection belongs to the different blocking window accessible from the node of stage i, $1 \leqslant i \leqslant n-t-1$. This node has $|\mathbb{SI}_i| = 2^{i-1}$ accessible input terminals. Output terminals which can be reached from this node may belong to up to 2^{n-t-i} blocking windows. Connection $\langle x, \mathbb{Y} \rangle$, where $|\mathbb{Y}| = f_i \leqslant 2^{n-t-i}$ and each element of $\mathbb{Y}$ belongs to a different blocking window reachable from stage i, may be set up through different planes. Connection $\mathcal{MMC}(31, 1, 4, 2)$ is also shown in Fig. 6.10 in doted lines.

Nonblocking conditions for $\log_2(N, 0, p)$ will be now given for different sizes of blocking windows. Firstly the case with $1 \leqslant t \leqslant \lfloor n/2 \rfloor$ will be considered.

Nonblocking Conditions for $1 \leqslant t \leqslant \lfloor n/2 \rfloor$. These conditions were given in [32]

THEOREM 6.11 *The $\log_2(N, 0, p)$ switching fabric is wide-sense nonblocking for $1 \leqslant t \leqslant \lfloor n/2 \rfloor$ provided that blocking window algorithm is used if and only if:*

$$p \geqslant \begin{cases} t \times 2^{n-t-1} + 2^{n-2t-1}, & \text{for } 1 \leqslant t < \left\lfloor \frac{n}{2} \right\rfloor \text{ and} \\ & \text{for } t = \left\lfloor \frac{n}{2} \right\rfloor \text{ when } n \text{ is odd}, \\ t \times 2^{n-t-1} + 1, & \text{for } t = \left\lfloor \frac{n}{2} \right\rfloor \text{ when } n \text{ is even.} \end{cases} \tag{6.20}$$

Proof. Let the new connection $\langle x, y \rangle$ be added in the switching fabric. It may be a point-to-point connection or a subconnection of the mul-

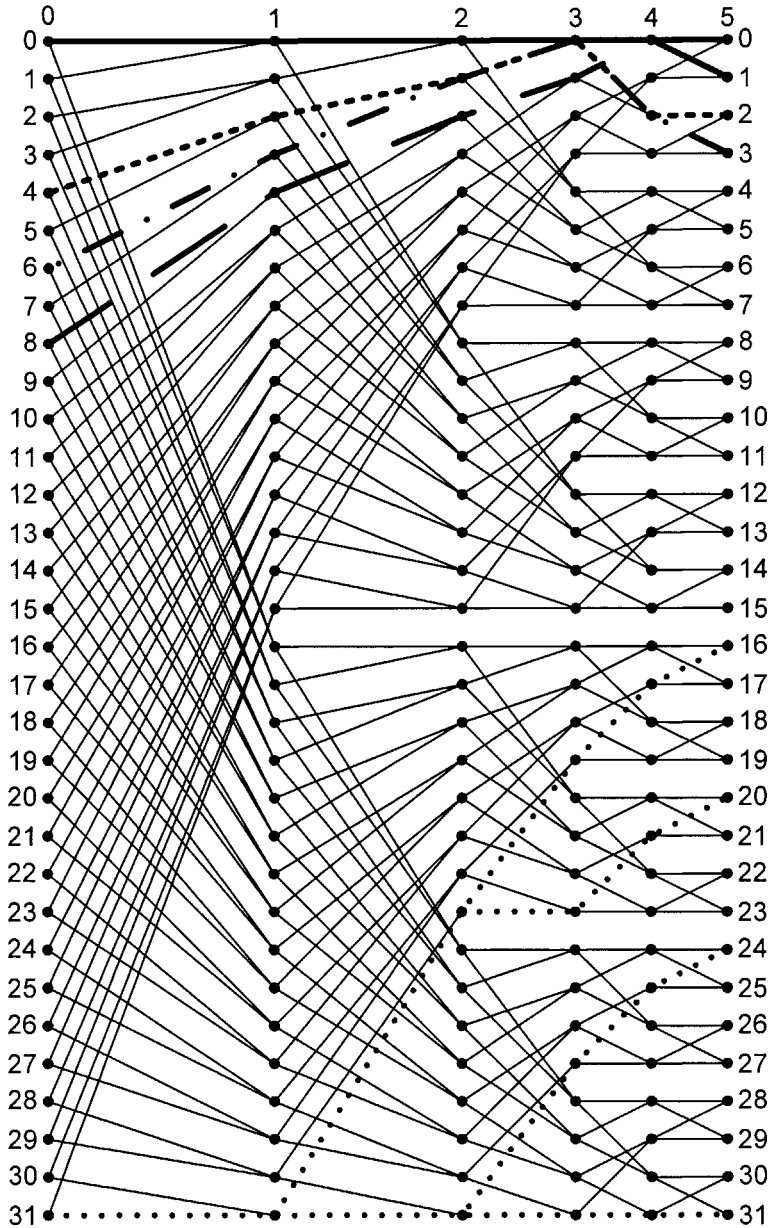


Figure 6.10. An example of $\mathbb{MBC}(0,3)$ and $\mathcal{MMC}(31,1,4,2)$ in $\log_2(32,0,p)$

ticast connection. This connection may be blocked by connections in $\mathbb{MBC}(y, n-t)$, which may occupy p_1 planes, where

$$p_1 = |\mathbb{MBC}(y, n-t)| = \sum_{i=n-t}^{n-1} |\mathbb{SO}_i| = \sum_{i=n-t}^{n-1} 2^{n-i-1} = 2^t - 1, \quad (6.21)$$

and these connections will engage all output terminals of the blocking window containing output terminal y. On the other hand, for each input terminal x_k in $\mathbb{SI}_i$, $1 \leqslant i \leqslant t$, we can set up $\mathcal{MMC}(x_k, i, 2^{n-t-i} - 1, t)$, since from 2^{n-t-i} blocking windows which can be reached from the node of stage i, one blocking window is totally seized by connections in $\mathbb{MBC}(y, n-t)$. Since each blocking window contains 2^t output terminals, we can set up these connections until all output terminals in one blocking window will be occupied. When $t < \lfloor n/2 \rfloor$ or when $t = \lfloor n/2 \rfloor$ and n is odd we may have $\sum_{i=1}^{t} |\mathbb{SI}_i| = 2^t - 1$ such $\mathcal{MMC}$ connections, which will occupy up to $2^t - 1$ output terminals in blocking windows accessible from nodes of stages from 1 to t. It means that we may also consider one $\mathcal{MMC}$ from an input terminal in $\mathbb{SI}_{t+1}$, but only if $t < \lfloor n/2 \rfloor$ (for $t = \lfloor n/2 \rfloor$ we have $t + 1 = n - t$ and the node in stage $t + 1$ cannot be considered since it was already considered in $\mathbb{MBC}(y, n-t)$). Each connection to the output terminals in different blocking windows may be set up through separate planes, so another p_2 plane may be inaccessible to the new connection, where

$$\begin{aligned} p_2 &= \sum_{i=1}^{t} (|\mathbb{SI}_i| \cdot f_i) + f_{t+1} = \\ &= \sum_{i=1}^{t} \left[2^{i-1}\left(2^{n-t-i} - 1\right)\right] + \left(2^{n-2t-1} - 1\right) = \\ &= \sum_{i=1}^{t} \left(2^{n-t-1} - 2^{i-1}\right) + \left(2^{n-2t-1} - 1\right) \\ &= t \cdot 2^{n-t-1} - 2^t + 2^{n-2t-1}. \end{aligned} \quad (6.22)$$

when $t < \lfloor n/2 \rfloor$, or

$$\begin{aligned} p_2 &= \sum_{i=1}^{t} (|\mathbb{SI}_i| \cdot f_i) = \sum_{i=1}^{t} \left[2^{i-1}\left(2^{n-t-i} - 1\right)\right] = \\ &= t \cdot 2^{n-t-1} - 2^t + 1, \end{aligned} \quad (6.23)$$

when $t = \lfloor n/2 \rfloor$.

For $t = \lfloor n/2 \rfloor$ and n even we have $n - t = t$. It means that output terminals of set $\mathbb{SO}_{n-t}$ belong to the same blocking window as output y and $\mathbb{SI}_t$ cannot be considered in calculating p_2. In this case we have:

$$\begin{aligned} p_2 &= \sum_{i=1}^{t-1} \left(|\mathbb{SI}_i| \cdot f_i\right) = \sum_{i=1}^{t-1} \left[2^{i-1}\left(2^{n-t-i} - 1\right)\right] \\ &= \sum_{i=1}^{t-1} \left(2^{n-t-1} - 2^{i-1}\right) = (t-1) \cdot 2^{n-t-1} - 2^{t-1} + 1. \quad (6.24) \end{aligned}$$

In the worst case sets of p_1 and p_2 planes are disjoint and one more plane is needed for setting up the new connection. For $t < \lfloor n/2 \rfloor$, after adding (6.21), (6.22), and 1, we obtain appropriate condition in inequality (6.20). For $t = \lfloor n/2 \rfloor$ and n odd we have:

$$\begin{aligned} p &\geqslant p_1 + p_2 + 1 = \\ &= \left(2^t - 1\right) + t \cdot 2^{n-t-1} - 2^t + 1 + 1 \\ &= t \cdot 2^{n-t-1} + 1. \quad (6.25) \end{aligned}$$

If in inequality (6.20) for n odd we put $t = \lfloor n/2 \rfloor$ we obtain inequality 6.25. This means that inequality (6.20) is also true for $t = \lfloor n/2 \rfloor$ and n odd. For $t = \lfloor n/2 \rfloor$ and n even we have:

$$\begin{aligned} p &\geqslant p_1 + p_2 + 1 = \\ &= \left(2^t - 1\right) + (t-1) \times 2^{n-t-1} - 2^{t-1} + 1 + 1 \\ &= (t-1) \times 2^{n-t-1} + 2^{t-1} + 1, \\ p &\geqslant t \times 2^{n-t-1} - 2^{n-t-1} + 2^{t-1} + 1, \quad (6.26) \end{aligned}$$

and after taking into account that for $t = \lfloor n/2 \rfloor$ and n even we have $n - t = t$, we obtain condition given in inequality (6.20) for n even. Necessity can be proved by showing the set of events leading to occupancy of p planes, where p is given by (6.20). Construction of this set of events may differ depending on t and n. Example of such set of event can be found in [32]. □

For n even and $t = n/2$ Theorem 6.11 gives the same result as in [164]:

$$p \geqslant \frac{n}{2} \times 2^{n-\frac{n}{2}-1} + 1 = \frac{n}{2} \times 2^{\frac{n}{2}-1} + 1. \quad (6.27)$$

and for n odd and $t = \lfloor n/2 \rfloor$ we have:

$$p \geqslant \left\lfloor \frac{n}{2} \right\rfloor \times 2^{n-\lfloor \frac{n}{2} \rfloor - 1} + 2^{n-2\lfloor \frac{n}{2} \rfloor - 1} = \left\lfloor \frac{n}{2} \right\rfloor \times 2^{\lfloor \frac{n}{2} \rfloor} + 1. \quad (6.28)$$

Non-blocking conditions for $\lfloor n/2 \rfloor < t \leqslant n$. Before we move to the case when $\lfloor n/2 \rfloor + 1 \leqslant t \leqslant n$, we will define a subblocking window, which is a subset of a blocking window [32].

DEFINITION 6.12 *Let a blocking window $\mathbb{BW}_i$ be divided into $K/2^{\lfloor n/2 \rfloor}$ subsets $\mathbb{SBW}_k$, $K = 2^t$, $k = 0, 1, \ldots, \dfrac{K}{2^{\lfloor n/2 \rfloor}} - 1$, $\lfloor n/2 \rfloor + 1 \leqslant t \leqslant n$, and $\mathbb{SBW}_k = \{K \cdot i + 2^{\lfloor n/2 \rfloor} \cdot k,\ K \cdot i + 2^{\lfloor n/2 \rfloor} \cdot k + 1,\ K \cdot i + 2^{\lfloor n/2 \rfloor} \cdot k + 2,\ \ldots, K \cdot i + 2^{\lfloor n/2 \rfloor} \cdot (k+1) - 1\}$. Each subset $\mathbb{SBW}_k$ is called a subblocking window.*

THEOREM 6.13 *The $\log_2(N, 0, p)$ switching fabric is wide-sense nonblocking for $\lfloor n/2 \rfloor + 1 \leqslant t \leqslant n$ provided that blocking window algorithm is used if and only if:*

$$p \geqslant 2^t + (n - t - 2) \times 2^{n-t-1} - 2^{2t-n-1} + 1. \tag{6.29}$$

Proof. Sufficiency. Let a new connection $\langle x, \mathbb{Y} \rangle$ is to be set up, where $\mathbb{Y}$ belongs to one blocking window, and let $t = \lfloor n/2 \rfloor + j$. In the blocking window there are $\dfrac{2^t}{2^{\lfloor n/2 \rfloor}} = 2^j$ subblocking windows $\mathbb{SBW}$.
Even n: Connections in $\mathbb{MBC}(y_0, t - j)$, $y_0 \in \mathbb{SBW}_0$, and $y_0 \in \mathbb{Y}$ may occupy

$$p_1^1 = \sum_{i=0}^{t-j-1} 2^i = 2^{t-j} - 1 \tag{6.30}$$

planes. All output terminals in $\mathbb{SBW}_0$ and all input terminals accessible from stage $\lfloor n/2 \rfloor$ are busy, since for even n we have $\left|\mathbb{SO}_{\lfloor n/2 \rfloor}\right| = \left|\mathbb{SI}_{\lfloor n/2 \rfloor}\right|$. Connections to output terminals in $\mathbb{SBW}_1$ my occupy next p_1^2 planes, where

$$\begin{aligned} p_1^2 &= \sum_{i=0}^{t-j-2} 2^i + \sum_{i=n-t-1}^{t-j-2} 2^i = \\ &= \left(2^{t-j-1} - 1\right) + \left(2^{t-j-1} - 1\right) - \left(2^{n-t-1} - 1\right) = \\ &= 2^{t-j} - 2^{n-t-1} - 1. \end{aligned} \tag{6.31}$$

When $y_1 \in \mathbb{SBW}_1$ and $y_1 \in \mathbb{Y}$, than these planes will be inaccessible by $\langle x, \mathbb{Y} \rangle$. In $\mathbb{SBW}_1$ we have still 2^{n-t-1} free output terminals. In stages numbered from 1 to $n-t-1$ we have $\sum_{i=1}^{n-t-1} 2^{i-1} = 2^{n-t-1} - 1$ accessible input terminals. It means that there will be possible to set up in each of these stages $\mathcal{MMC}$ for all accessible input terminals. When $j = 1$ all output terminals in the blocking window are already assigned. For $j > 1$ next $\mathbb{SBW}$ may be considered.

Connections in p_1^1 and p_1^2 occupy $2 \cdot 2^{\lfloor n/2 \rfloor - 2} = 2^{n-t+1}$ input terminals of $\mathbb{SII}_{\lfloor n/2 \rfloor + 1}$, so we have still $2 \cdot 2^{\lfloor n/2 \rfloor - 2}$ input terminals free. We may construct two $\mathbb{MBC}$: $\mathbb{MBC}(y_2, \lfloor n/2 \rfloor + 1)$ and $\mathbb{MBC}(y_3, \lfloor n/2 \rfloor + 1)$, $y_2, y_3 \in \mathbb{Y}$, $y_2, y_3 \in \mathbb{SBW}_2$ and these connections will use

$$p_1^3 = 2 \times \sum_{i=0}^{t-j-2} 2^i = 2 \times \left(2^{t-j-1} - 1\right) = 2^{t-j} - 2 \tag{6.32}$$

planes. All outputs in $\mathbb{SBW}_2$ are now used and we have still free outputs in $\mathbb{SBW}_3$. So we may consider the next $\mathbb{MBC}$ in stage $\lfloor n/2 \rfloor + 2$.

Let stage $t - x$, $1 \leqslant x \leqslant j - 2$ is considered. In this stage we have $|\mathbb{SI}_{t-x}| = 2^{t-x-1}$ and $|\mathbb{SO}_{t-x}| = 2^{n-t+x-1}$. Connections in $\mathbb{MBC}$s of previous stages will occupy 2^{t-x-3} inputs accessible from stage $t - x$ so we have still $2^{t-x-1} - 2^{t-x-3} = 3 \cdot 2^{t-x-3}$ free inputs. These inputs may be used for constructing $\mathbb{MBC}$s in stage $t - x$. We may construct $\dfrac{3 \cdot 2^{t-x-3}}{2^{n-t+x-1}} = 3 \cdot 2^{2t-n-2x-2} = 3 \cdot 2^{2j-2x-2}$ such $\mathbb{MBC}$s. Each $\mathbb{MBC}$ in stage $t - x$ will occupy $\sum_{i=0}^{n-t+x-1} 2^i = 2^{n-t+x} - 1$ output terminals in $\mathbb{SBW}$ and one output terminal in $\mathbb{Y}$. In one $\mathbb{SBW}$ we may construct $\dfrac{2^{\lfloor n/2 \rfloor}}{2^{n-t+x}} = 2^{j-x}\,(t = \lfloor n/2 \rfloor + 1)$ $\mathbb{MBC}$s. So $\mathbb{MBC}$s in stage $t - x$ will occupy output terminals in next $\dfrac{3 \cdot 2^{2t-n-2x-2}}{2^{j-x}} = 3 \cdot 2^{j-x-2}$ $\mathbb{SBW}$s. These $\mathbb{SBW}$s are numbered from $3 \cdot 2^{j-x-2}$ to $3 \cdot 2^{j-x-1} - 1$. In this way next

$$p_1^{j-x+2} = 3 \times 2^{2j-2x-2} \times \left(2^{t-2j+x} - 1\right) = 3 \times \left(2^{t-x-2} - 2^{2j-2x-2}\right) \tag{6.33}$$

planes will be inaccessible by connection $\langle x, \mathbb{Y} \rangle$.

Finally, in stage t number of $\mathbb{SBW}$s which can be occupied by $\mathbb{MBC}$s is $3 \cdot 2^{j-2}$, however, in this case we have only $2^j - 3 \cdot 2^{j-2} = 2^{j-2}$ $\mathbb{SBW}$s with free outputs. So, the $\mathbb{MBC}$s in stage t will occupy

$$p_1^{j+2} = 2^{j-2} \times 2^j \times \sum_{i=0}^{t-2j-1} 2^i = 2^{j-2} \times \left(2^{t-j} - 2^j\right) \tag{6.34}$$

planes, and all outputs in the blocking window are assigned except 2^{n-t-1} outputs in $\mathbb{SBW}_1$.

Through the nodes of stage i, $1 \leqslant i \leqslant n - t - 1$ we may now construct $|\mathbb{SI}_i|$ different $\mathcal{MMC}$s and these connections may occupy

$$p_2 = \sum_{i=1}^{n-t-1} 2^{i-1} \times 2^{n-t-1} = (n - t - 1) \times 2^{n-t-1}. \tag{6.35}$$

$t=$	1	2	3	4	5	6	7	8	9	10	11
$n=2$	[2]	2									
3	[3]	[3]	4								
4	6	[4]	6	8							
5	12	9	[8]	12	16						
6	24	18	[13]	15	24	32					
7	48	36	25	[20]	29	48	64				
8	96	72	50	[33]	35	57	96	128			
9	192	144	100	65	[48]	65	113	192	256		
10	384	288	200	130	81	[79]	125	225	384	512	
11	768	576	400	260	161	[112]	141	245	449	768	1024
12	1536	1152	800	520	322	193	[175]	265	485	897	1536

Table 6.3. The number of planes as a function of n and t in $\log_2(N, 0, p)$ switching fabrics

In the worst case these sets are disjoint and one more plane is needed for connection $\langle x, \mathbb{Y} \rangle$, so in general we can write:

$$p \geqslant p_1^1 + p_1^2 + p_1^3 + \sum_{x=0}^{j-2} p_1^{j-x+2} + p_1^{j+2} + p_2 + 1. \tag{6.36}$$

After putting equations (6.30) – (6.35) to inequality (6.36) and taking into account special cases for $j = 1$ and $j = 2$ we will obtain equation (6.29).

For n odd construction of the worst case scenario is similar to that for n even. However, in this case, output terminals in two $\mathbb{SBW}$s will be occupied by two $\mathbb{MBC}(y, \lfloor n/2 \rfloor + 1)$, so $2 \cdot p_1^1$ planes can be inaccessible for the new connection. Also for n odd we have $n = 2 \cdot \lfloor n/2 \rfloor + 1$. The total number of planes is in this case also given by (6.29).

Necessity can be proved by showing the set of events leading to occupancy of p planes, where p is given by (6.29). □

From Theorems 6.11 and 6.13 we can derive the value of t, for which p is minimum. For $1 \leqslant t \leqslant \lfloor n/2 \rfloor$ the minimum number of planes is obtained for $t = \lfloor n/2 \rfloor$, since (6.20) is growing when t is getting smaller. For $n = 3$ and $\lfloor n/2 \rfloor \leqslant t \leqslant n$ the value of p is the same for $t = \lfloor n/2 \rfloor$ and $t = \lfloor n/2 \rfloor + 1$. For n odd and $n > 3$ lower number of planes is needed for $t = \lfloor n/2 \rfloor + 1$. When n is even lower number of planes is needed for $t = \lfloor n/2 \rfloor + 1$ and $n \geqslant 10$. The number of planes for different n and t are given in Table 6.3.

Similar conditions for $\log_2(N, m, p)$ switching fabrics was derived by Danilewicz and Kabaciński [30, 31]. These conditions were later improved by Hwang and Lin [58], who have shown that for $m > 2$ different worst case scenario can be encountered at stages from $n-1$ to $n-t$. For $1 \leqslant t \leqslant \lfloor n/2 \rfloor$ the theorem is as follows:

THEOREM 6.14 *The* $\log_2(N, m, p)$ *switching fabric is wide-sense non-blocking for* $1 \leqslant t \leqslant \lfloor n/2 \rfloor$ *provided that blocking window algorithm is used if and only if:*

$$p \geqslant \begin{cases} \lfloor t \cdot 2^{n-t-1} + 2^{n-2t-1} - 2^{-m} \rfloor + 1, \\ \qquad \text{for } m = 1 \text{ and } 1 \leqslant t < \lfloor \frac{n}{2} \rfloor, \\ \qquad \text{for } m = 1,\ t = \lfloor \frac{n}{2} \rfloor \text{ and } n \text{ odd}, \\ \qquad \text{for } m = 2 \text{ and } t \geqslant 2, \\ \qquad \text{and for } 2 < m \leqslant t; \\ \lfloor t \cdot 2^{n-t-1} \rfloor + 1, \\ \qquad \text{for } m = 1,\ t = \lfloor \frac{n}{2} \rfloor \text{ and } n \text{ even}; \\ \lfloor t \cdot 2^{n-t-1} + 2^{n-2t-1} - 2^{-(t+1)} \rfloor + 1, \\ \qquad \text{for } m \geqslant 2 \text{ and } t = 1, \\ \qquad \text{and for } m > 2 \text{ and } t = 2; \\ \lfloor 2^{t-2} + t \cdot 2^{n-t-1} - \frac{t}{2} + 2^{n-2t-1} - 2^{-(t+1)} \rfloor + 1, \\ \qquad \text{for } m > 2 \text{ and } 2 < t < m; \\ \lfloor 2^{t-2} + t \cdot 2^{n-t-1} - \frac{m}{2} - 2^{t-m} + 2^{n-2t-1} - 2^{-m} + 1 \rfloor + 1, \\ \qquad \text{for } m > 2 \text{ and } 2 < m \leqslant t. \end{cases} \tag{6.37}$$

Proof. In stage $n-i$, $1 \leqslant i \leqslant t$, the new connection $\langle x, y \rangle$ can be blocked by connections to output terminals in $|\mathbb{SO}_{n-i}|$ and these connections may occupy p_1 planes, where

$$p_1 = \begin{cases} \dfrac{t}{2}, & \text{for } m \leqslant 2 \text{ and } t \leqslant m \\ & \text{or } m > 2 \text{ and } t \leqslant 2; \\ \dfrac{m}{2} + 2^{t-m} - 1, & \text{for } m \leqslant 2 \text{ and } t > m; \\ 2^{t-2}, & \text{for } m > 2 \text{ and } t > 2. \end{cases} \tag{6.38}$$

In stage i, $1 \leqslant i \leqslant t$, this connection can be blocked by multicast connections from input terminals in $|\mathbb{SI}_i|$ to output terminals in accessible

blocking windows, i.e.,

$$p_2 = \begin{cases} t \cdot 2^{n-t-1} - \dfrac{t}{2}, & \text{for } m \geqslant t; \\ t \cdot 2^{n-t-1} - \dfrac{m}{2} - 2^{t-m} + 1, & \text{for } m < t \end{cases} \tag{6.39}$$

planes may be unavailable for the connection. Finally, if stage $t+1$ can be considered, one more multicast connection can be set up from any input terminal in $|\mathbb{SI}_{t+1}|$. This means that next

$$p_3 = \begin{cases} 2^{n-2t-1} - 2^{-(t+1)}, & \text{for } t \leqslant m-1 \text{ and } 2t < n+m-1, \\ 2^{n-2t-1} - 2^{-m}, & \text{for } m-1 < t \text{ and } 2t < n+m-1, \\ 0, & \text{for } 2t \geqslant n+m-1 \end{cases} \tag{6.40}$$

planes may be unavailable by the connection $\langle x, y \rangle$. When all these sets of planes are disjoint, the number of planes required is given by

$$p \geqslant \lfloor p_1 + p_2 + p_3 \rfloor + 1. \tag{6.41}$$

After putting (6.38), (6.39), and (6.40) in (6.41) we obtain (6.37). □

When $\lfloor n/2 \rfloor < t \leqslant \lceil (m+n)/2 \rceil$, the respective theorem is as follows:

THEOREM 6.15 *The* $\log_2(N, m, p)$ *switching fabric is wide-sense nonblocking for* $\lfloor n/2 \rfloor < t \leqslant \lceil (m+n)/2 \rceil$ *provided that blocking window algorithm is used if and only if:*

$$p \geqslant \begin{cases} \left\lfloor 2^{t-m} + (m+n-t-2) \cdot 2^{n-t-1} \right\rfloor + 1, \\ \qquad\qquad \text{for } m \leqslant 2 \text{ and } t > m; \\ \left\lfloor 2^{t-2} + t \cdot 2^{n-t-1} - \frac{t}{2} + 2^{n-2t-1} - 2^{-(t+1)} \right\rfloor + 1, \\ \qquad\qquad \text{for } m > 2 \text{ and } 2 < t < m; \\ \left\lfloor 2^{t-2} + t \cdot 2^{n-t-1} - \frac{m}{2} - 2^{t-m} + 2^{n-2t-1} - 2^{-m} + 1 \right\rfloor, \\ \qquad\qquad \text{for } m > 2 \text{ and } 2 < m \leqslant t. \end{cases} \tag{6.42}$$

Proof. The proof is very similar the the proof of the previous theorem. In this case p_1 is also given by equation (6.38), while p_2, and p_3 are given

by [30]:

$$p_2 = \begin{cases} (m+n-t-1)\cdot 2^{n-t-1} - \dfrac{m+n-t-1}{2}, \\ \qquad \text{for } t \geqslant n-1 \text{ and } 2t \geqslant n+m-1; \\ (m+n-t-2)\cdot 2^{n-t-2} - \dfrac{m}{2} + 1, \\ \qquad \text{for } t < n-1 \text{ and } 2t \geqslant n+m-1; \\ t\cdot 2^{n-t-1} - \dfrac{t}{2}, \\ \qquad \text{for } t \leqslant m \text{ and } 2t < n+m-1; \\ t\cdot 2^{n-t-1} - \dfrac{m}{2} - 2^{t-m} + 1, \\ \qquad \text{for } t > m \text{ and } 2t < n+m-1; \end{cases} \tag{6.43}$$

$$p_3 = \begin{cases} 0, & \text{for } 2t \geqslant n+m-1; \\ 2^{n-2t-1} - 2^{-(t+1)}, & \text{for } 2t < n+m-1 \\ & \text{and } t < m-1; \\ 2^{n-2t-1} - 2^{-m}, & \text{for } 2t < n+m-1 \\ & \text{and } m-1 < t < n-1; \\ 2^{-t} - 2^{t-m-n+1}, & \text{for } 2t < n+m-1 \\ & \text{and } n-1 < t \leqslant n+m-2. \end{cases} \tag{6.44}$$

□

6.3.3 Rearrangeable Conditions

Rearrangeable $\log_2(N,0,p)$ switching fabrics for unicast connections were considered by Lea [92]. In the proof of the next theorem matrix $\mathbf{H}_r^{(c)}$, $c = 2^{\lfloor n/2 \rfloor}$, $r = 2^{\lceil n/2 \rceil}$, will be used to represent the maximal assignment in $\log_2(N,0,p)$ switching fabric which is defined as follows:

DEFINITION 6.16

$$\mathbf{H}_r^{(c)} = \left[h_{i,j} = \begin{cases} k, & \text{if there exist } k \text{ connections } \langle x,y \rangle \\ & \text{were } x \in \mathbb{I}_i \text{ and } y \in \mathbb{O}_j \\ 0, & \text{if there is no such connection} \end{cases} \right] \tag{6.45}$$

From the above definition we know that matrix $\mathbf{H}_r^{(c)}$ contains $r = 2^{\lceil n/2 \rceil}$ rows and columns. Row i represents set $\mathbb{I}_i$, column j corresponds to set $\mathbb{O}_j$ (see Def. 3.4 and 3.5), and entry $h_{i,j}$ represents the number of connections between input terminals in $\mathbb{I}_i$ and output terminals in

$\mathbb{O}_j$. This matrix $\mathbf{H}_r^{(c)}$ is similar to the matrix defined for three-stage switching fabrics (see Def. 5.36), but instead of first and third stage switches sets $\mathbb{I}_i$ and $\mathbb{O}_j$ are used. This matrix has following properties:

$$\sum_{i=1}^{r} h_{i,j} = \sum_{j=1}^{r} h_{i,j} = c = 2^{\lfloor \frac{n}{2} \rfloor}, \tag{6.46}$$

since there are $2^{\lfloor n/2 \rfloor}$ input and output terminals in each $\mathbb{I}_i$ and $\mathbb{O}_j$, respectively.

THEOREM 6.17 *The* $\log_2(N, 0, p)$ *switching fabric is rearrangeable nonblocking for unicast connections if and only if:*

$$p \geqslant 2^{\lfloor \frac{n}{2} \rfloor}. \tag{6.47}$$

Proof. Necessity. When we consider a graph of intersecting paths in one plane shown in Fig. 3.26, we can conclude, that at most $2^{\lfloor n/2 \rfloor}$ paths meet in one node of this graph. Each path has to be set up through different plane, so at least $2^{\lfloor n/2 \rfloor}$ planes are needed.
Sufficiency. Assume that a maximum assignment is to be realized in the switching fabric. Connections in this assignment can be represented in the form of matrix $\mathbf{H}_r^{(c)}$, which can be decomposed into $2^{\lfloor n/2 \rfloor}$ elementary permutation matrices $\mathbf{E}_{r \times r}$. A set of connections corresponding to one matrix $\mathbf{E}_{r \times r}$ can be set up in one plane without conflict. □

Rearrangeable $\log_2(N, m, p)$ switching fabrics were considered by Lea and Shyy [95].

THEOREM 6.18 *The* $\log_2(N, m, p)$ *switching fabric is rearrangeable nonblocking for unicast connections if and only if:*

$$p \geqslant 2^{\lfloor \frac{n-m}{2} \rfloor}. \tag{6.48}$$

Proof. There are 2^m connecting paths in each plane. One plane can be decomposed into 2^m subplanes with a single path between any input-output pair of any subplane. Each suplane has 2^{n-m} inputs. A maximal assignment can be divided into 2^m subassignments such that no conflict will occur in stages from 1 to m and from $n+1$ to $n+m$. This can be done using for instance the looping algorithm described in section 5.1.6.2. Each subassignment needs $2^{\lfloor (n-m)/2 \rfloor}$ subplanes according to Theorem 6.17. Since there are 2^m subassignments and 2^m subplanes in one plane, then $p \geqslant 2^{\lfloor (n-m)/2 \rfloor}$ planes is sufficient. This number of planes is also necessary since there are maximal assignments which

cannot be realized in $\log_2(N, m, p)$ switching fabric with $p < 2^{\lfloor (n-m)/2 \rfloor}$, for example identity assignment:

$$\Pi = \begin{pmatrix} 0, & 1, & 2, & \ldots & N-1 \\ 0, & 1, & 2, & \ldots & N-1 \end{pmatrix}. \tag{6.49}$$

□

An example of the $\log_2(16, 2, 1)$ switching fabric is shown in Fig. 6.11. Since $m = 2$, this switching fabric has $2^2 = 4$ connecting paths between any input-output terminals pair. It is decomposed into four subplanes, each subplane has $2^{n-m} = 4$ inputs. Rearrangeable condition requires two planes. Let consider the identity assignment. Connections $\langle 0, 0 \rangle$, $\langle 1, 1 \rangle$, $\langle 2, 2 \rangle$, and $\langle 3, 3 \rangle$, for instance, may be set up through subplanes 1, 2, 3, and 4 of one plane, respectively. Any two of these connections cannot be set up in one subplane. Connections $\langle 4, 4 \rangle$, $\langle 5, 5 \rangle$, $\langle 6, 6 \rangle$, and $\langle 7, 7 \rangle$ will have to use another plane.

The above results were extended to $\log_{2^d}(N, m, p)$ switching fabrics in [149].

THEOREM 6.19 *The* $\log_d(N, m, p)$ *switching fabric is rearrangeable nonblocking for unicast connections if and only if:*

$$p \geqslant (2^d)^{\lfloor \frac{n-m}{2} \rfloor}, \tag{6.50}$$

where $n = \log_{2^d} N$.

The number of planes in rearangeable nonblocking $\log_2(N, m, p)$ switching fabrics as a function of n and m is compared in Table 6.4. It can be seen, that the lowest number of planes is needed for $m = n - 1$, and this number of planes is equal to 1. The $\log_2(N, n-1, 1)$ switching fabric is the Beneš switching fabric composed of 2×2 switches. The minimum number of 2×2 switches is also obtain for $m = n - 1$.

6.3.4 Rearranging Algorithms

For one-at-a-time connection model the *Paull* algorithm can be used. In this algorithm input terminals are divided into sets $\mathbb{I}_i$, $1 \leqslant i \leqslant 2^{\lfloor n/2 \rfloor}$, and output terminals are grouped into sets $\mathbb{O}_j$, $1 \leqslant j \leqslant 2^{\lfloor n/2 \rfloor}$. The state of the switching fabric can be presented in the form of state matrix $\mathbf{S}$ of size $r \times r$, $r = 2^{\lceil n/2 \rceil}$; rows correspond to sets $\mathbb{I}_i$; columns correspond to sets $\mathbb{O}_j$; and the matrix is defined as follows:

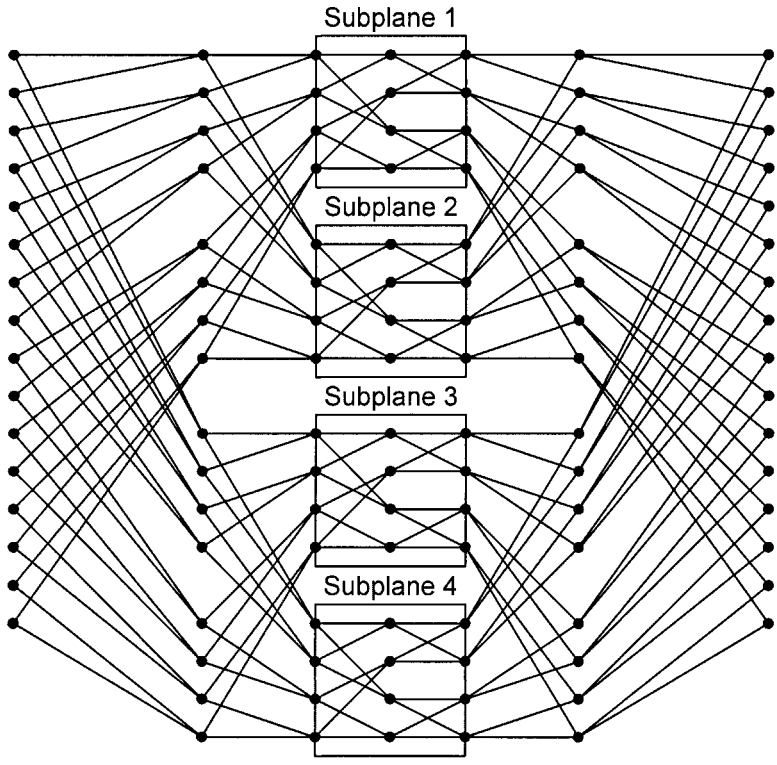


Figure 6.11. Decomposition of the $\log_2(16, 2, 1)$ switching fabric

$m=$	0	1	2	3	4	5	6	7	8	9	10	11
$n=2$	2	1										
3	2	2	1									
4	4	2	2	1								
5	4	4	2	2	1							
6	8	4	4	2	2	1						
7	8	8	4	4	2	2	1					
8	16	8	8	4	4	2	2	1				
9	16	16	8	8	4	4	2	2	1			
10	32	16	16	8	8	4	4	2	2	1		
11	32	32	16	16	8	8	4	4	2	2	1	
12	64	32	32	16	16	8	8	4	4	2	2	1

Table 6.4. The number of planes as a function of n and m in rearrangeable nonblocking $\log_2(N, m, p)$ switching fabrics with unicast connections

DEFINITION 6.20

$$\mathbf{S} = \left[s_{i,j} = \begin{cases} k, & \text{if there exist connection } \langle x, y \rangle \text{ set up} \\ & \text{through plane } k, \text{ were } x \in \mathbb{I}_i \text{ and } y \in \mathbb{O}_j \\ 0, & \text{if there is no such connection} \end{cases} \right] \quad (6.51)$$

In each cell there may be more than one entry, since more than one connection can be set up between input-output terminal pairs of the same $\mathbb{I}_i$ and $\mathbb{O}_j$. Since $|\mathbb{I}_i| = |\mathbb{O}_j| = 2^{\lfloor n/2 \rfloor}$, the number of entries in each row and each column should be not greater than this number. In the *Paull* algorithm any entry representing a plane may appear in one column or one row only once, similar to Clos switching fabrics (see section 5.1.6.1). This assumption is not always necessary and sometimes leads to unnecessary rearrangements. For example, let consider $\log_2(16, 0, 4)$ switching fabric with connections $\langle 0, 0 \rangle$, $\langle 1, 1 \rangle$, $\langle 2, 2 \rangle$, $\langle 4, 4 \rangle$, $\langle 5, 5 \rangle$, and $\langle 6, 6 \rangle$. The bipartite graph of this switching fabric is shown in Fig. 6.12a, and the state matrix is given in Fig. 6.12b. In this Figure, connections set up through plane 1 (i.e., connection $\langle 0, 0 \rangle$), plane 2 (connections $\langle 1, 1 \rangle$ and $\langle 4, 4 \rangle$), plane 3 (connections $\langle 2, 2 \rangle$ and $\langle 5, 5 \rangle$), and plane 4 (connection $\langle 6, 6 \rangle$) are shown in bold lines. Let connection $\langle 3, 7 \rangle$ be set up, $3 \in \mathbb{I}_1$, $7 \in \mathbb{O}_2$. When we look at the state matrix we conclude that this connection is blocked, since planes 1, 2, and 3 are in row 1, and planes 2, 3, and 4 are in column 2. We can unblock this new connection by re-switching connection $\langle 0, 0 \rangle$ from plane 1 to plane 4, and plane 1 becomes available for connection $\langle 3, 7 \rangle$. But in fact connection $\langle 3, 7 \rangle$ may be also set up through plane 1 or 2 without moving connection $\langle 0, 0 \rangle$. The algorithm which does not do such unnecessary rearrangements was proposed in [83].

In rearrangeable $\log_2(N, 0, p)$ switching fabrics with simultaneous connections model, an assignment can be set up using any of matrix decomposition or graph coloring algorithms. Another approach to setting up simultaneous connections was recently proposed in [79]. In this approach connections of an assignment are routed one-by-one using sequential routing. The algorithm can be described as follows:

ALGORITHM 6.9 *Sequential assignment routing*

Step 1 Take the first not connected input, starting from input 0.

Step 2 Set up the connection for this input through the first available plane, starting from plane 1.

Step 3 Repeat steps 1 and 2 until all inputs are connected.

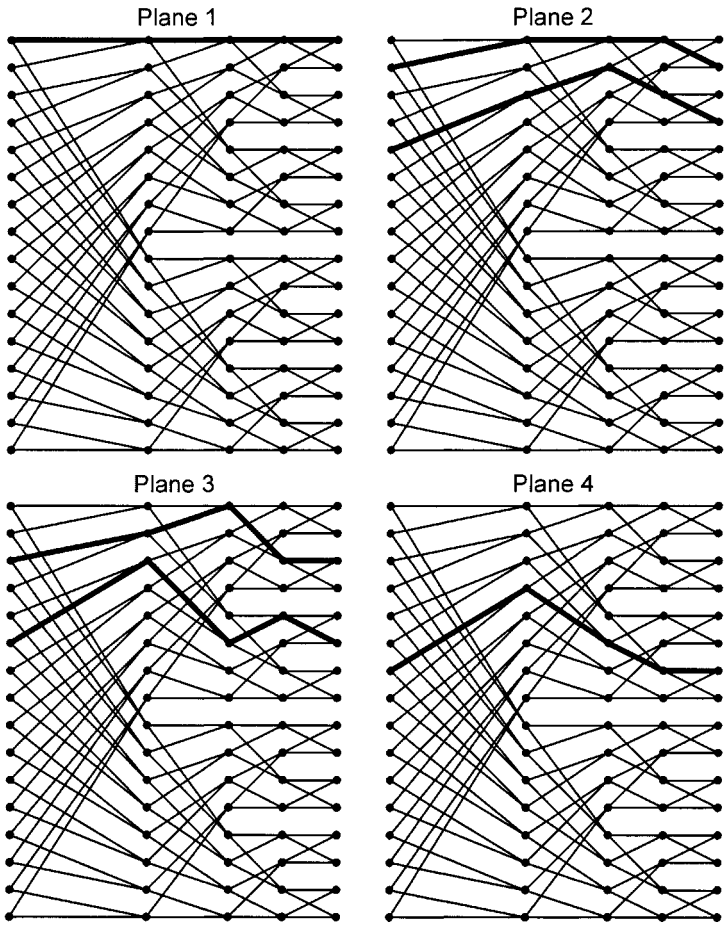


Figure 6.12a. Examples of connections in $\log_2(16, 0, 4)$ switching fabric

THEOREM 6.21 *The* $\log_2(N, 0, p)$ *switching fabric with* n *even, routes all possible permutations when algorithm 6.9 is used if and only if*

$$p \geqslant 2^{\frac{n}{2}}. \tag{6.52}$$

Proof. Let the permutation Π is to be set up and this permutation contains at least one set of connections represented by $\mathcal{G}(\mathbb{I}_i, \mathbb{O}_j)$. Without lost of generality, we can assume, that this is $\mathcal{G}(\mathbb{I}_0, \mathbb{O}_0)$. It is clear that each connection in this configuration will require a separate plane. Input terminal 0 will be connected with output terminal $\pi(0)$ through plane 1. Input terminal 1 will be connected with output terminal $\pi(1)$ through plane 2, etc. Finally, input terminal $2^{n/2} - 1$ will be connected with out-

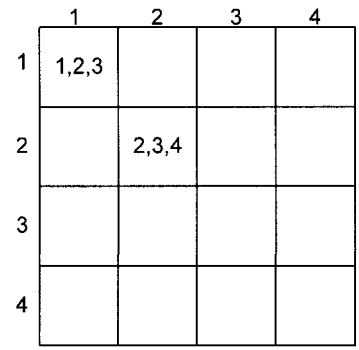


Figure 6.12b. Examples of connections in $\log_2(16, 0, 4)$ switching fabric - state matrix

put terminal $\pi(2^{n/2} - 1)$ through plane $2^{n/2}$, $\pi(i) \in \mathbb{O}_0, 0 \leqslant i \leqslant 2^{n/2} - 1$. These connections will not be in conflict with any other connection in Π. It is also clear that if there is another such set of connections in Π, it will use the same $2^{n/2}$ planes. So at least $2^{n/2}$ planes are needed.

Sufficiency will be proved by showing that there is no other configuration of connections in Π which will require more planes. Let us consider connections realized between $\mathbb{I}_i$ and $\mathbb{O}_j$. All paths from input terminals in $\mathbb{I}_i$ to output terminals in $\mathbb{O}_j$ go through the node in stage $n/2$. Assume now, that some input terminals in $\mathbb{I}_i$ are to be connected with output terminals in $\mathbb{O}_k$, $k \neq j$. Say there is one such connection, and it uses the same node, say in stage $(n/2) - 1$ as some other connections represented by $\mathcal{G}(\mathbb{I}_i, \mathbb{O}_j)$, and the same node in stage $(n/2) + 1$ as some other connections represented by $\mathcal{G}(\mathbb{I}_l, \mathbb{O}_k)$. Connections represented by $\mathcal{G}(\mathbb{I}_i, \mathbb{O}_j)$ and $\mathcal{G}(\mathbb{I}_l, \mathbb{O}_k)$ use different nodes in stage $n/2$, so they are not in conflict in this stage. When $i < l$, connections from input terminals in $\mathbb{I}_i$ to output terminals in $\mathbb{O}_j$ will be set up through planes numbered from 1 to $2^{n/2} - 1$, and the connection from one input terminal in $\mathbb{I}_i$ to one output terminal in $\mathbb{O}_k$ will also be set up through one of these $2^{n/2} - 1$ planes, since it goes through the different node in stage $n/2$ and connections from input terminals in $\mathbb{I}_l$ are not set up yet. When $i > l$, connections from input terminals in $\mathbb{I}_l$ are already set up. Therefore, connection $(\mathbb{I}_i, \mathbb{O}_k)$ may be in conflict with connections from $2^{n/2-1} - 1$ input terminals in $\mathbb{I}_i$ and $2^{(n/2)-1} - 1$ planes will be unavailable since paths from $2^{(n/2)-1}$ input terminals meet in the node of stage $(n/2) - 1$ (one of these paths is the path from the considered input terminal which is to be connected). This connection may be also in conflict with connections to $2^{(n/2)-1} - 1$ output terminals in $\mathbb{O}_k$ and other $2^{(n/2)-1} - 1$ planes will be unavailable. If these sets of planes are disjoint we have

$2 \cdot \left(2^{(n/2)-1} - 1\right) = 2^{n/2} - 2$ planes unavailable for connection $(\mathbb{I}_i, \mathbb{O}_k)$, but 2 planes are still available and one of them will be used. Similarly, in $\mathbb{O}_j$ we have now one output terminal free and say the connection to this output terminal is to be set up from input terminal in $\mathbb{I}_h$, $h \neq i$. By analogy, we may have $2^{n/2} - 2$ planes unavailable for connection $(\mathbb{I}_h, \mathbb{O}_j)$, but 2 planes are still available.

We have assumed, that in $\mathbb{I}_i$ one input terminal was connected to one output terminal in $\mathbb{O}_k$, while other input terminals in this set were connected to output terminals in $\mathbb{O}_j$. Similar arguments show that when there are more than one connection $(\mathbb{I}_i, \mathbb{O}_k)$ and they use the same nodes in stages lower than $(n/2) - 1$ and higher than $(n/2) + 1$, the number of planes needed to successfully route all connections in Π are even less than $2^{n/2}$. So all together not more than $2^{n/2}$ planes are needed to realize all possible permutations when connections are set up using the sequential assignment routing algorithm. □

Algorithm 6.9 can be also used to route maximal assignment in the $\log_2(N, 0, p)$ switching fabric with n odd, however, such switching fabric requires more planes than the rearrangeable nonblocking one [79].

THEOREM 6.22 *The* $\log_2(N, 0, p)$ *with* n *odd, routes all possible permutations when Algorithm 6.9 is used if and only if:*

$$p \geqslant \begin{cases} 2^{\frac{n-1}{2}} & \text{for } n = 3 \\ 1.25 \cdot 2^{\frac{n-1}{2}} & \text{for } n \geqslant 5 \end{cases}. \tag{6.53}$$

Proof. Similarly as in Theorem 6.21, let us consider connections represented by $\mathcal{G}(\mathbb{I}_i, \mathbb{O}_j)$. It is clear that such configuration needs $2^{(n-1)/2}$ planes, and all paths go through nodes in stages $(n-1)/2$ and $(n+1)/2$. Let us now assume that some input terminals in $\mathbb{I}_i$ are to be connected with output terminals in $\mathbb{O}_j$ and $\mathbb{O}_k$, $k \neq j$. For $n = 3$ it is easy to all possible classes of path configurations with conflicts. These classes are shown in Fig. 6.13. As can be seen from this Figure, connecting paths in each of these classes can be set up using two planes. Connections set up through plane 1 are marked in dashed lines, while connections routed through plane 2 – in bold lines. So, for $\log_2(8, 0, p)$ switching fabric, $p = 2^{(n-1)/2}$ planes are sufficient to set up all possible permutations.

When n is odd and $n \geqslant 5$, all paths in $\mathcal{G}(\mathbb{I}_i, \mathbb{O}_j)$ and $\mathcal{G}(\mathbb{I}_i, \mathbb{O}_{\widehat{j}})$ use the same node in stage $(n-1)/2$, while path in $\mathcal{G}(\mathbb{I}_i, \mathbb{O}_j)$ and $\mathcal{G}\left(\mathbb{I}_{\widehat{i}}, \mathbb{O}_j\right)$ use the same node in stage $(n+1)/2$. Let assume that there are $2^{(n-1)/2} - c$ connections $(\mathbb{I}_i, \mathbb{O}_j)$ and c connections $(\mathbb{I}_i, \mathbb{O}_{\widehat{j}})$. Say these c connections are set up through planes numbered from 1 to c, and connections $(\mathbb{I}_i, \mathbb{O}_j)$

Figure 6.13. State classes for $\log_2(8,0,p)$ switching fabric

are set up through $2^{(n-1)/2} - c$ planes numbered from $c+1$ to $2^{(n-1)/2}$, since all these connections go through the same node in stage $(n-1)/2$. Connections to c outputs in $\mathbb{O}_j$ are not set up yet, and say these outputs will be used by connections $(\mathbb{I}_{\widehat{i}}, \mathbb{O}_j)$. They will go through the same node in stage $(n+1)/2$ as connections $(\mathbb{I}_i, \mathbb{O}_j)$. Since connections $(\mathbb{I}_i, \mathbb{O}_j)$ occupy planes numbered from $c+1$ to $2^{(n-1)/2}$, only planes numbered from 1 to c can be used by $(\mathbb{I}_{\widehat{i}}, \mathbb{O}_j)$. In $\mathbb{I}_{\widehat{i}}$ there may be c other connections, say $(\mathbb{I}_{\widehat{i}}, \mathbb{O}_k)$, $k \neq j$, $k \neq \widehat{j}$, and these connections may use planes numbered from 1 to c (if these $(\mathbb{I}_{\widehat{i}}, \mathbb{O}_k)$ are to be set up earlier than $(\mathbb{I}_{\widehat{i}}, \mathbb{O}_j)$). In the worst case, connections $(\mathbb{I}_{\widehat{i}}, \mathbb{O}_k)$ and $(\mathbb{I}_{\widehat{i}}, \mathbb{O}_j)$ go through the same node in stage $(n-3)/2$, but they use different nodes in stage $(n-1)/2$. Therefore, for c connections $(\mathbb{I}_{\widehat{i}}, \mathbb{O}_j)$ planes numbered from 1 to c will be not accessible in the node of stage $(n-1)/2$, and planes numbered from $c+1$ to $2^{(n-1)/2}$ will be not accessible in the node of stage $(n+1)/2$. So these connections will require additional planes numbered from $2^{(n-1)/2}+1$ to $2^{(n-1)/2}+c$. We have c connections $(\mathbb{I}_{\widehat{i}}, \mathbb{O}_k)$ and c connections $(\mathbb{I}_{\widehat{i}}, \mathbb{O}_j)$ which go through the same node in stage $(n-3)/2$. Since $2^{(n-3)/2}$ input terminals are accessible from the node in this stage, we have $2c \leqslant 2^{(n-3)/2}$. The maximum number of additional planes is obtained when $c = 2^{(n-5)/2}$, and the total number of planes for $n \geqslant 5$ is

$$p \geqslant 2^{\frac{n-1}{2}} + 2^{\frac{n-5}{2}} = 5 \cdot 2^{\frac{n-5}{2}} = 1.25 \cdot 2^{\frac{n-1}{2}}. \tag{6.54}$$

□

6.4 Multirate Switching

6.4.1 Switching Fabrics

In multirate connections the model described earlier and bipartite graph representation of the switching fabric can be used. The difference in multirate $\log_2(N,m,p)$ switching fabrics is only in that each of input and output terminals has the capacity equal to β, and each of interstage links has the capacity of 1. In bipartite graph representation these capacities are assign to respective nodes. Each node in stages 0 and $n+m$ has the capacity of β, while in remaining stages each node has the ca-

pacity equal to 1. More connections can use the same connecting path and the same nodes, provided that the sum of their weights is lower than its capacity. Some architectures like extended delta and Cantor switching fabrics were considered earlier in [114, 115] and [22]. Multirate $\log_d(N, m, p)$ switching fabrics were considered in [93, 94], where necessary and sufficient conditions were given for discrete bandwidth case when $1/b$ is an integer, and sufficient condition, as well as necessary condition for $B \in (1-b, \beta]$ for continuous bandwidth case was proved. Better upper bounds, which in some cases are also lower bounds, were given in [81, 82]. Multirate multicast switching fabrics were considered in [80]. In the next section these main results will be described.

6.4.2 Strict-sense Nonblocking

6.4.2.1 Unicast Connections

In the case of discrete bandwidth case the strict-sense nonblocking conditions for the $\log_2(N, 0, p)$ switching fabric are as follows:

THEOREM 6.23 *The* $\log_2(N, 0, p)$ *switching fabric is strict-sense nonblocking for the discrete bandwidth case if and only if:*

$$p \geqslant 2 \left\lfloor \frac{L_1 \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - B}{b} \right\rfloor}{\left\lfloor \frac{1-B+b}{b} \right\rfloor} \right\rfloor + 1, \tag{6.55a}$$

for n odd, and

$$\begin{aligned} p \geqslant\ & 2 \left\lfloor \frac{L_1 \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - B}{b} \right\rfloor}{\left\lfloor \frac{1-B+b}{b} \right\rfloor} \right\rfloor + \\ & + \left\lfloor \frac{L_2 \left\lfloor \frac{\beta}{b} \right\rfloor}{\left\lfloor \frac{1-B+b}{b} \right\rfloor} \right\rfloor + \left\lfloor \frac{L_3 + L_4}{\left\lfloor \frac{1-B+b}{b} \right\rfloor} \right\rfloor + 1, \end{aligned} \tag{6.55b}$$

for n even, where

$$L_1 = \begin{cases} 2^{\frac{n-1}{2}} - 1, & \text{for } n \text{ odd}, \\ 2^{\frac{n}{2}-1} - 1, & \text{for } n \text{ even}, \end{cases} \tag{6.55c}$$

$$L_2 = 2^{\frac{n}{2}-1}, \tag{6.55d}$$

$$L_3 = L_1 \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - B}{b} \right\rfloor - \left\lfloor \frac{L_1 \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - B}{b} \right\rfloor}{\left\lfloor \frac{1 - B + b}{b} \right\rfloor} \right\rfloor \left\lfloor \frac{1 - B + b}{b} \right\rfloor, \tag{6.55e}$$

and

$$L_4 = L_2 \left\lfloor \frac{\beta}{b} \right\rfloor - \left\lfloor \frac{L_2 \left\lfloor \frac{\beta}{b} \right\rfloor}{\left\lfloor \frac{1 - B + b}{b} \right\rfloor} \right\rfloor \left\lfloor \frac{1 - B + b}{b} \right\rfloor. \tag{6.55f}$$

Proof. Sufficiency. Let connection $\langle x, y, \omega \rangle$ is to be set up. The connecting path will be inaccessible by this connection if any node on this path carries connections of total weight greater than $1 - \omega$. In the discrete bandwidth case we should have at least $\lfloor (1 - \omega + b)/b \rfloor$ connections of weight b to block any node in connection path. At each of input terminals other than x we may have up to $\lfloor \beta/b \rfloor$ connections of such weight. Additionally, $\lfloor (\beta - \omega)/b \rfloor$ connections of weight b may be set up at input terminal x. For n even in nodes of stages from 1 to $(n/2) - 1$ we can set up $\sum_{i=1}^{(n/2)-1} |\mathbb{SI}_i| \lfloor \beta/b \rfloor$ connections of weight b, and $\lfloor (\beta - \omega)/b \rfloor$ such connections at input terminal x. These connections may occupy

$$p_1 = \left\lfloor \frac{\sum_{i=1}^{(n/2)-1} |\mathbb{SI}_i| \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - \omega}{b} \right\rfloor}{\left\lfloor \frac{1 - \omega + b}{b} \right\rfloor} \right\rfloor = \left\lfloor \frac{L_1 \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - \omega}{b} \right\rfloor}{\left\lfloor \frac{1 - \omega + b}{b} \right\rfloor} \right\rfloor \tag{6.56}$$

planes in such way, that they will be inaccessible by the new connection. At input terminals in $\mathbb{SI}_i$ there may still be a free bandwidth available

for L_3 connections of weight b,

$$L_3 = L_1 \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - \omega}{b} \right\rfloor - \left\lfloor \frac{L_1 \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - \omega}{b} \right\rfloor}{\left\lfloor \frac{1 - \omega + b}{b} \right\rfloor} \right\rfloor \left\lfloor \frac{1 - \omega + b}{b} \right\rfloor, \tag{6.57}$$

but, since $L_3 < \lfloor (1 - \omega + b)/b \rfloor$, these connections cannot block the additional plane. By analogy, in nodes of stages from $(n/2) + 1$ to $n - 1$ we can set up $\sum_{i=1}^{(n/2)-1} |\mathbb{SO}_{n-i}| \lfloor \beta/b \rfloor$ connections of weight b to output terminals in $\mathbb{SO}_{n-i}$, and $\lfloor (\beta - \omega)/b \rfloor$ such connections to output terminal y. These connections will make

$$\begin{aligned} p_2 &= \left\lfloor \frac{\sum_{i=1}^{(n/2)-1} |\mathbb{SO}_{n-i}| \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - \omega}{b} \right\rfloor}{\left\lfloor \frac{1 - \omega + b}{b} \right\rfloor} \right\rfloor \\ &= \left\lfloor \frac{L_1 \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - \omega}{b} \right\rfloor}{\left\lfloor \frac{1 - \omega + b}{b} \right\rfloor} \right\rfloor, \end{aligned} \tag{6.58}$$

planes inaccessible by the new connection. At output terminals in $\mathbb{SO}_{n-i}$ there may be also a free bandwidth which can be used by L_3 of weight b. Additionally, in the node of stage $n/2$, we may have a set of connections from input terminals in $\mathbb{SI}_{n/2}$ to output terminals in $\mathbb{SO}_{n/2}$, and since $|\mathbb{SI}_{n/2}| = |\mathbb{SO}_{n/2}|$, these connections may occupy

$$p_3 = \left\lfloor \frac{|\mathbb{SI}_{n/2}| \left\lfloor \frac{\beta}{b} \right\rfloor}{\left\lfloor \frac{1 - \omega + b}{b} \right\rfloor} \right\rfloor = \left\lfloor \frac{L_2 \left\lfloor \frac{\beta}{b} \right\rfloor}{\left\lfloor \frac{1 - \omega + b}{b} \right\rfloor} \right\rfloor \tag{6.59}$$

planes. At input terminals in $\mathbb{SI}_{n/2}$ and output terminals in $\mathbb{SO}_{n/2}$ there may be the remaining bandwidth which can be used by L_4 connections of weight b, where

$$L_4 = L_2 \left\lfloor \frac{\beta}{b} \right\rfloor - \left\lfloor \frac{L_2 \left\lfloor \frac{\beta}{b} \right\rfloor}{\left\lfloor \frac{1 - \omega + b}{b} \right\rfloor} \right\rfloor \left\lfloor \frac{1 - \omega + b}{b} \right\rfloor. \tag{6.60}$$

We can set up L_3 connections of weight b from input terminals in $\mathbb{SI}_i$ to output terminals in $\mathbb{SO}_i$, $1 \leqslant i < n/2$ and L_4 connections of weight b from input terminals in $\mathbb{SI}_{n/2}$ to output terminals in $\mathbb{SO}_{n/2}$. When $L_3 + L_4 > \lfloor (1 - \omega + b)/b \rfloor$ these connections may occupy next

$$p_4 = \left\lfloor \frac{L_3 + L_4}{\left\lfloor \dfrac{1 - \omega + b}{b} \right\rfloor} \right\rfloor \tag{6.61}$$

planes. In the worst case these sets of planes are disjoint and one more plane is needed for the new connection $\langle x, y, \omega \rangle$. Thus, for n even, we have

$$\begin{aligned} p &\geqslant p_1 + p_2 + p_3 + p_4 + 1, \\ p &\geqslant 2 \left\lfloor \frac{L_1 \left\lfloor \dfrac{\beta}{b} \right\rfloor + \left\lfloor \dfrac{\beta - \omega}{b} \right\rfloor}{\left\lfloor \dfrac{1 - \omega + b}{b} \right\rfloor} \right\rfloor + \\ &\quad + \left\lfloor \frac{L_2 \left\lfloor \dfrac{\beta}{b} \right\rfloor}{\left\lfloor \dfrac{1 - \omega + b}{b} \right\rfloor} \right\rfloor + \left\lfloor \frac{L_3 + L_4}{\left\lfloor \dfrac{1 - \omega + b}{b} \right\rfloor} \right\rfloor + 1. \end{aligned} \tag{6.62}$$

Inequality (6.62) must be maximized through all ω, and this maximum is reached for $\omega = B$. Putting $\omega = B$ in (6.62) we obtain (6.55b).

When n is odd the proof is similar, but we count sections from 1 to $(n-1)/2$ in both cases (p_1 and p_2), and there is no p_3 since all stages are already taken into account. Similarly, the remaining connections of weight b cannot block any additional plane. So we have:

$$\begin{aligned} p_1 &= \left\lfloor \frac{\displaystyle\sum_{i=1}^{(n-1)/2} |\mathbb{SI}_i| \left\lfloor \dfrac{\beta}{b} \right\rfloor + \left\lfloor \dfrac{\beta - \omega}{b} \right\rfloor}{\left\lfloor \dfrac{1 - \omega + b}{b} \right\rfloor} \right\rfloor \\ &= \left\lfloor \frac{L_1 \left\lfloor \dfrac{\beta}{b} \right\rfloor + \left\lfloor \dfrac{\beta - \omega}{b} \right\rfloor}{\left\lfloor \dfrac{1 - \omega + b}{b} \right\rfloor} \right\rfloor, \end{aligned} \tag{6.63}$$

$$p_2 = \left\lceil \frac{\sum_{i=1}^{(n-1)/2} |\mathbb{SO}_{n-i}| \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta-\omega}{b} \right\rfloor}{\left\lfloor \frac{1-\omega+b}{b} \right\rfloor} \right\rceil$$

$$= \left\lceil \frac{L_1 \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta-\omega}{b} \right\rfloor}{\left\lfloor \frac{1-\omega+b}{b} \right\rfloor} \right\rceil, \tag{6.64}$$

and

$$p \geqslant p_1 + p_2 + 1 = 2 \left\lceil \frac{L_1 \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta-\omega}{b} \right\rfloor}{\left\lfloor \frac{1-\omega+b}{b} \right\rfloor} \right\rceil + 1. \tag{6.65}$$

Putting $\omega = B$ in (6.65) we obtain (6.55).

Following the same pattern analysis, it is possible to construct the set of events showing that when less number of planes is used, the considered switching fabric is blocking. □

Let us consider the $\log_2(16, 0, p)$ switching fabric with $\beta = 0.8$, $b = 0.2$ and $B = 0.6$. The new connection is $\langle 0, 0, 0.6 \rangle$. At input terminal 0 we may set up one connection of weight 0.2 (the same is true for output terminal 0). In each of the remaining input (output) terminals, four connections of weight 0.2 can be set up. Connections from input terminals 0 and 1 ($\mathbb{SI}_1 = \{1\}$), will occupy plane 1 ($p_1 = 1$) as shown in Fig. 6.14. These are connections numbered 1, 2 and 3. At input terminal 1 it is still possible to set up 2 connections of weight b ($L_3 = 2$). Similarly, connections to output terminals 0 and 1 ($\mathbb{SO}_3 = \{1\}$) will occupy plane 2 ($p_2 = 1$). Connections numbered from 7 to 12 from input terminals in ($\mathbb{SI}_2 = \{2, 3\}$) to output terminals in ($\mathbb{SO}_2 = \{2, 3\}$) will occupy planes 3 and 4 ($p_3 = 2$). We have still free bandwidth in these terminals for $L_4 = 2$ connections. So these L_3 and L_4 connections may occupy the next plane ($p_4 = 1$). One more plane is needed for connection $\langle 0, 0, 0.6 \rangle$.

Strictly nonblocking conditions for $\log_2(N, 0, p)$ switching fabric with continuous bandwidth connections are given by the following theorem. Functions (4.27), (4.28), (4.29), (4.30), and (4.31) defined in section 4.2.3.2 are used in this theorem.

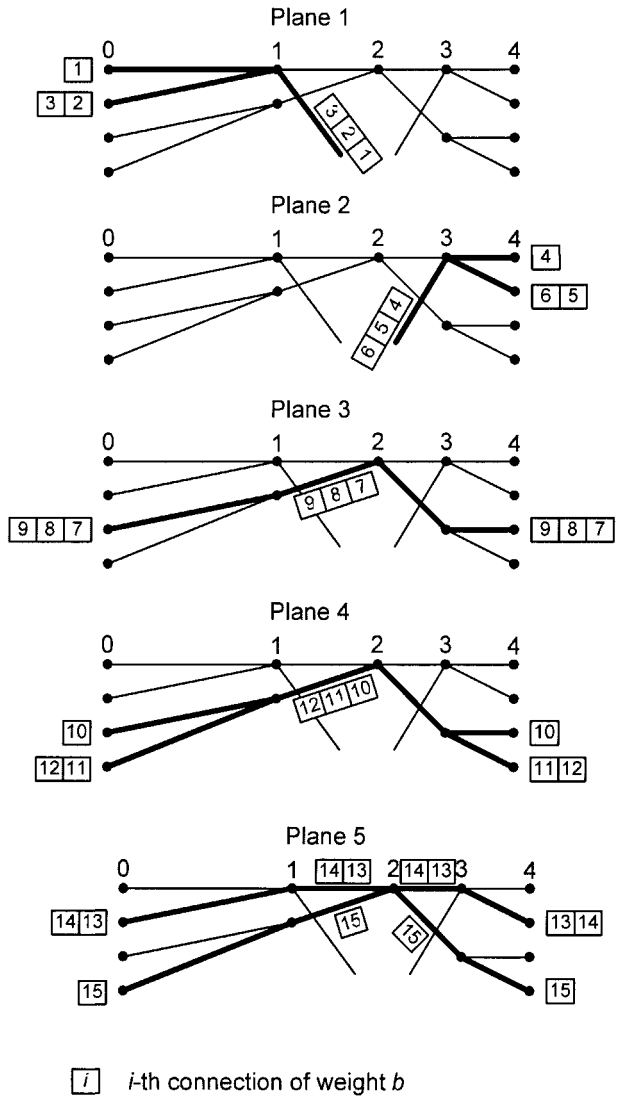


Figure 6.14. The worst state in the $\log_2(16, 0, p)$ switching fabric

THEOREM 6.24 *The* $\log_2(N, 0, p)$ *switching fabric is strict-sense nonblocking for the continuous bandwidth case, if:*

$$p \geqslant \begin{cases} 2L_1 \left\lfloor \dfrac{\beta}{b} \right\rfloor + 1, & \text{for } n \text{ odd}, \\ 2L_1 \left\lfloor \dfrac{\beta}{b} \right\rfloor + L_2 \left\lfloor \dfrac{\beta}{b} \right\rfloor + 1, & \text{for } n \text{ even}, \end{cases} \tag{6.66a}$$

when $B \in (1-b,\ \beta]$*;*

$$p \geqslant \begin{cases} 2\left\lfloor \dfrac{L_1 \left\lfloor \dfrac{\beta}{b} \right\rfloor + \left\lfloor \dfrac{\beta - B}{b} \right\rfloor}{2} \right\rfloor + 1, & \text{for } n \text{ odd}, \\ 2\left\lfloor \dfrac{L_1 \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \dfrac{\beta - B}{b} \right\rfloor}{2} \right\rfloor + \left\lfloor \dfrac{L_2 \left\lfloor \dfrac{\beta}{b} \right\rfloor}{2} \right\rfloor + 1, & \text{for } n \text{ even}, \end{cases} \tag{6.66b}$$

when $B \in \left(1-2b, \frac{1}{2}\right]$ *and* $\frac{1}{4} < b < \frac{1}{2}$*;*

$$p \geqslant \begin{cases} 2L_1 P(\beta; 1-B) + 1, & \text{for } n \text{ odd}, \\ 2L_1 P(\beta; 1-B) + L_2 P(\beta; 1-B) + 1, & \text{for } n \text{ even}, \end{cases} \tag{6.66c}$$

for all other B *and* $R_1(\beta; 1-B) < b$*;*

$$p \geqslant \begin{cases} 2\left[L_1 P(\beta; 1-B) + \lfloor R_3(L_1; a(\beta; B)) \rfloor + \right. \\ \left. + P(\alpha_1(L_1; \beta; B) + R_5(\beta - B); 1-B)\right] + 1, \\ \qquad \text{for } n \text{ odd}, \\ 2\left[L_1 P(\beta; 1-B) + \lfloor R_3(L_1; a(\beta; B)) \rfloor + \right. \\ \left. + P(\alpha_1(L_1; \beta; B) + R_5(\beta - B); 1-B)\right] + \\ + L_2 P(\beta; 1-B) + \lfloor R_3(L_2; a(\beta; B)) \rfloor + \\ + P(\alpha_1(L_2; \beta; B); 1-B) + \\ P(R_1(\alpha_1(L_1; \beta; B) + R_5(\beta - B); 1-B) + \\ + R_1(\alpha_1(L_2; \beta; B); 1-B); 1-B) + 1, \\ \qquad \text{for } n \text{ even}, \end{cases} \tag{6.66d}$$

for all other B *and* $b \leqslant R_1(\beta; 1-B) < 2b$*; and*

$$p \geqslant \begin{cases} 2\left[L_1 P(\beta; 1-B) + \right. \\ \left. P(\alpha_2(L_1; \beta; B) + R_5(\beta - B); 1-B)\right] + 1, \\ \qquad \text{for } n \text{ odd}, \\ 2\left[L_1 P(\beta; 1-B) + \right. \\ \left. P(\alpha_2(L_1; \beta; B) + R_5(\beta - B); 1-B)\right] + \\ + \left(L_2 P(\beta; 1-B) + P(\alpha_2(L_2; \beta; B); 1-B)\right) \\ + P(R_1(\alpha_2(L_1; \beta; B) + R_5(\beta - B); 1-B) + \\ + R_1(\alpha_2(L_2; \beta; B); 1-B); 1-B) + 1, \\ \qquad \text{for } n \text{ even}, \end{cases} \tag{6.66e}$$

for all other B *and* $R_1(\beta; 1-B) \geqslant 2b$*; where*

$$L_1 = \begin{cases} 2^{\frac{n-1}{2}} - 1, & \text{for } n \text{ odd}, \\ 2^{\frac{n}{2}-1} - 1, & \text{for } n \text{ even}, \end{cases} \tag{6.66f}$$

$$L_2 = 2^{\frac{n}{2}-1}, \tag{6.66g}$$

$$a(\beta; B) = R_2(\beta; 1-B) + 1, \tag{6.66h}$$

$$\alpha_1(L; \beta; B) = [L - a(\beta; B)\lfloor R_3(L; a(\beta; B))\rfloor] R_1(\beta; 1-B), \tag{6.66i}$$

$$\alpha_2(L; \beta; B) = LR_1(\beta; 1-B). \tag{6.66j}$$

Proof. Sufficient conditions will be proved by showing the worst state in the switching fabric. Suppose we want to add the new connection $\langle x, y, \omega \rangle$, $0 < b \leqslant \omega \leqslant B \leqslant \beta \leqslant 1$. Any path from input terminal x will be inaccessible for the new connection of weight ω if there is a node on the connecting path, which already carries connections of the total weight greater than $1-\omega$. In the worst case this sum of weights should be as small as possible, say $1-\omega+\varepsilon$, where ε is close to but greater than 0. However, when $1-\omega < b$ a path is inaccessible if a connections of weight b is set up through at least one of its nodes. When $1-\omega \geqslant B$, it is not possible that one connection will occupy a weight $1-\omega+\varepsilon$. In this case we have to set up at least two connections. When $1-\omega < 2b$, the total weight in the path will be equal to $2b$. In other cases it is always possible to set up one or more connections with a total weight $1-\omega+\varepsilon$. Similarly as in the two-stage one-sided and three-stage two-sided switching fabrics we have three cases:

1 $1-\omega < b$,

2 $b \leqslant 1-\omega < 2b$ and $1-\omega \geqslant B$,

3 other values of $1-\omega$.

First, the case with n even will be considered.
Case 1: $1-\omega < b$. In this case, a plane is inaccessible for the new connection of weight ω if there is a node on the connecting path, which already carries a connection of weight b. In the worst case, each of such connections can be set up through different planes. At one input terminal $\lfloor \beta/b \rfloor$ connections of weight b can be set up. If β/b is not an integer, there is some free bandwidth at this input terminal, but its

weight is less than b and it cannot be used by the next connection. In the node of stage i, $1 \leqslant i < \lfloor n/2 \rfloor$, the connecting path of connection $\langle x, y, \omega \rangle$ may be blocked by connecting path from input terminals in $\mathbb{SI}_i$. Similarly, in the node of stage $n-i$, this connection may be blocked by connecting paths to output terminals in $\mathbb{SO}_{n-i}$. We have:

$$p_1 = \sum_{i=1}^{(n/2)-1} |\mathbb{SI}_i| \left\lfloor \frac{\beta}{b} \right\rfloor = \left(2^{\frac{n}{2}-1} - 1\right) \left\lfloor \frac{\beta}{b} \right\rfloor = L_1 \left\lfloor \frac{\beta}{b} \right\rfloor \tag{6.67}$$

and

$$p_2 = \sum_{i=1}^{(n/2)-1} |\mathbb{SO}_{n-i}| \left\lfloor \frac{\beta}{b} \right\rfloor = \left(2^{\frac{n}{2}-1} - 1\right) \left\lfloor \frac{\beta}{b} \right\rfloor = L_1 \left\lfloor \frac{\beta}{b} \right\rfloor. \tag{6.68}$$

All bandwidth at these input and output terminals is used an no other connection of weight b can be set up. In the node of stage $n/2$ connection from input terminals in $\mathbb{SI}_{n/2}$ to output terminals in $\mathbb{SO}_{n/2}$ will occupy

$$p_3 = |\mathbb{SI}_{n/2}| \left\lfloor \frac{\beta}{b} \right\rfloor = 2^{\frac{n}{2}-1} \left\lfloor \frac{\beta}{b} \right\rfloor = L_2 \left\lfloor \frac{\beta}{b} \right\rfloor. \tag{6.69}$$

In the worst case, one more plane is needed for setting up connection $\langle x, y, \omega \rangle$. So, for $1 - \omega < b$, we obtain:

$$p \geqslant p_1 + p_2 + p_3 + 1 = 2L_1 \left\lfloor \frac{\beta}{b} \right\rfloor + L_2 \left\lfloor \frac{\beta}{b} \right\rfloor + 1. \tag{6.70}$$

Case 2: $b \leqslant 1 - \omega < 2b$ and $1 - \omega \geqslant B$. In this case plane is inaccessible for a new connection of weight ω, if two connections of weight b already intersect in one node. Similarly as in the previous case, we may have $\lfloor \beta/b \rfloor$ connections of weight b at each input terminal other than x, and $\lfloor \beta - \omega/b \rfloor$ connections of this weight at input terminal x. So all together we may have $\sum_{i=1}^{(n/2)-1} |\mathbb{SI}_i| \lfloor \beta/b \rfloor + \lfloor (\beta - \omega)/b \rfloor$ connections, and two connections make a node inaccessible for the new connection $\langle x, y, \omega \rangle$. Thus,

$$\begin{aligned} p_1 &= \left\lfloor \frac{\sum_{i=1}^{(n/2)-1} |\mathbb{SI}_i| \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - \omega}{b} \right\rfloor}{2} \right\rfloor \\ &= \left\lfloor \frac{L_1 \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - \omega}{b} \right\rfloor}{2} \right\rfloor, \end{aligned} \tag{6.71}$$

and by analogy

$$p_2 = \left\lfloor \frac{\sum_{i=1}^{(n/2)-1} |\mathbb{SO}_{n-i}| \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta-\omega}{b} \right\rfloor}{2} \right\rfloor = \left\lfloor \frac{L_1 \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta-\omega}{b} \right\rfloor}{2} \right\rfloor . \tag{6.72}$$

If the number of connections of weight b is odd, there is a possibility to set up one more connection, but it would not block any plane. In the node of stage $n/2$ next

$$p_3 = \left\lfloor \frac{|\mathbb{SI}_{n/2}| \left\lfloor \frac{\beta}{b} \right\rfloor}{2} \right\rfloor = \left\lfloor \frac{2^{\frac{n}{2}-1} \left\lfloor \frac{\beta}{b} \right\rfloor}{2} \right\rfloor = \left\lfloor \frac{L_2 \left\lfloor \frac{\beta}{b} \right\rfloor}{2} \right\rfloor \tag{6.73}$$

planes may be blocked, and since the number of connections of weight b, which can be set up from these input terminals is always even, no bandwudth for any next connection is left. Therefore, we finally get:

$$p \geqslant 2 \left\lfloor \frac{L_1 \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta-\omega}{b} \right\rfloor}{2} \right\rfloor + \left\lfloor \frac{L_2 \left\lfloor \frac{\beta}{b} \right\rfloor}{2} \right\rfloor + 1. \tag{6.74}$$

Case 3: other values of $1-\omega$. For other values of $1-\omega$, a plane is inaccessible for the new connection $\langle x, y, \omega \rangle$ if there is a node on its connecting path which already carries connections of total weight greater than $1-\omega$. For $1-\omega < B$ only one connection of such weight may be set up. In the other case, at least two connections are to be set up, and their weights should be greater than $1-\omega$. At each of input or output terminals we may set up $P(\beta; 1-\omega)$ such connections. Each of sets $\mathbb{SI}_i$ ($\mathbb{SO}_{n-i}$) results in $\sum_{i=1}^{(n/2)-1} |\mathbb{SI}_i| P(\beta; 1-\omega)$ ($\sum_{i=1}^{(n/2)-1} |\mathbb{SO}_{n-i}| P(\beta; 1-\omega)$) such connections. There is still a free bandwidth of weight $\beta-\omega$ at input terminal x and at output terminal y, but since $\beta-\omega \leqslant 1-\omega$, then we cannot set up a connection of weight greater than $1-\omega$ at these terminals. There is still a free bandwidth of weight $R_1(\beta; 1-\omega)$ in each of input and output terminals except x and y. It may be used by next connections, provided that $R_1(\beta; 1-\omega) \geqslant b$. If $b \leqslant R_1(\beta; 1-\omega) < 2b$, several connections of

such weight, which passes through one node may make the plane inaccessible by the new connection. The minimum number of these connections is denoted by $a(\beta;\omega)$. Therefore, next $\left\lfloor R_3\left(\sum_{i=1}^{(n/2)-1}|\mathbb{SI}_i|;a\left(\beta;\omega\right)\right)\right\rfloor$ planes will be inaccessible by the new connection. In nodes accessible form stages 1 to $(n/2)-1$ we may still have free bandwidth of weight

$$\alpha_1\left(L_1;\beta;\omega\right)=\left[L_1-a\left(\beta;\omega\right)\left\lfloor R_3\left(L_1;a\left(\beta;\omega\right)\right)\right\rfloor\right]R_1\left(\beta;1-\omega\right) \quad (6.75)$$

and free bandwidth of weight $R_5(\beta-\omega)$ at input terminal x. The same is true for output terminals in $\mathbb{SO}_{n-i}$ and y. When this weight is greater than $1-\omega$, it may block $P\left(\alpha_1\left(L_1;\beta;\omega\right)+R_5(\beta-\omega);1-\omega\right)$ planes. We have now only $R_1\left(\alpha_1\left(L_1;\beta;\omega\right)+R_5(\beta-\omega);1-\omega\right)$ bandwidth left, and if it is greater than b it may be used by next connections but they will not block the plane since this bandwidth is not greater than $1-\omega$. If $R_1(\beta;1-\omega)\geqslant 2b$ the total weight of free bandwidth at input terminals in $\mathbb{SI}_i$ is equal to

$$\alpha_2\left(L_1;\beta;\omega\right)=L_1R_1\left(\beta;1-\omega\right), \quad (6.76)$$

and connections using this bandwidth and free bandwidth of weight $R_5\left(\beta-\omega\right)$ at input terminal x may occupy not more than $P\left(\alpha_2\left(L_1;\beta;\omega\right)+R_5\left(\beta-\omega\right);1-\omega\right)$ additional planes. The weight of remaining bandwidth is now $R_1\left(\alpha_2\left(L_1;\beta;\omega\right)+R_5\left(\beta-\omega\right);1-\omega\right)$. Again, the same is true for output terminals in $\mathbb{SO}_{n-i}$ and y. So, for other $1-\omega$ we obtain,

$$p_1=p_2=\begin{cases} L_1P\left(\beta;1-\omega\right), & \text{for } R_1\left(\beta;1-\omega\right)<b; \\ L_1P\left(\beta;1-\omega\right)+, & \text{for } b\leqslant R_1\left(\beta;1-\omega\right)<2b; \\ +\left\lfloor R_3\left(L_1;a\left(\beta;\omega\right)\right)\right\rfloor+ & \\ +P\left(\alpha_1\left(L_1;\beta;\omega\right)+R_5\left(\beta-\omega\right);1-\omega\right), & \\ L_1P\left(\beta;1-\omega\right)+ & \text{for } R_1\left(\beta;1-\omega\right)\geqslant 2b; \\ +P\left(\alpha_2\left(L_1;\beta;\omega\right)+R_5\left(\beta-\omega\right);1-\omega\right). & \end{cases} \quad (6.77)$$

By analogy to the above considerations and from Theorem 6.23 we obtain

$$p_3=\begin{cases} L_2P\left(\beta;1-\omega\right), & \text{for } R_1\left(\beta;1-\omega\right)<b; \\ L_2P\left(\beta;1-\omega\right)\left\lfloor R_3\left(L_2;a\left(\beta;\omega\right)\right)\right\rfloor+ & \\ +P\left(\alpha_1\left(L_2;\beta;\omega\right);1-\omega\right), & \text{for } b\leqslant R_1\left(\beta;1-\omega\right)<2b; \\ L_2P\left(\beta;1-\omega\right)+P\left(\alpha_2\left(L_2;\beta;\omega\right);1-\omega\right), & \\ & \text{for } R_1\left(\beta;1-\omega\right)\geqslant 2b; \end{cases} \quad (6.78)$$

At input terminals in $\mathbb{SI}_{n/2}$ there may be a free bandwidth of weight $R_1(\alpha_1(L_2;\beta;\omega);1-\omega)$ or $R_1(\alpha_2(L_2;\beta;\omega);1-\omega)$, depending on $R_1(\beta;1-\omega)$ being greater or less than $2b$. In combination with the remaining bandwidth from input terminals accessible from nodes in stages from 1 to $(n/2)-1$ we may have additionally

$$\begin{aligned} p_4 &= P(R_1(\alpha_1(L_1;\beta;\omega)+R_5(\beta-\omega);1-\omega) \\ &\quad +R_1(\alpha_1(L_2;\beta;\omega);1-\omega);1-\omega) \end{aligned} \tag{6.79}$$

planes occupied when $b \leqslant R_1(\beta;1-\omega) < 2b$, or

$$\begin{aligned} p_4 &= P(R_1(\alpha_2(L_1;\beta;\omega)+R_5(\beta-\omega);1-\omega) \\ &\quad +R_1(\alpha_2(L_2;\beta;\omega);1-\omega);1-\omega) \end{aligned} \tag{6.80}$$

planes occupied when $R_1(\beta;1-\omega) \geqslant 2b$. The total number of planes is in this case is determined by inequality

$$p \geqslant \lfloor p_1+p_2+p_3+p_4 \rfloor + 1. \tag{6.81}$$

The number of planes p must be maximized through all ω and it reaches maximum at $\omega = B$, then putting B in respective formulae we obtain conditions given in Theorem 6.24 for n even. For n odd equations for p_1 and p_2 can be derived by analogy, and in this case p_3 and p_4 are not considered. □

It should be noted, that for cases 1 and 2 the sufficient conditions given by theorem 6.24 are also necessary. The same is true for case 3 when $R_1(\beta;1-\omega) < 2b$. These conditions can be proved by showing the set of events leading to the occupancy of the required number of planes.

We will now show some of examples for switching fabrics with different b, B, and β. First, let us consider the $\log_2(32,0,p)$ switching fabric with $\beta = 1$, and let connection weights be between $b = 0.2$ and $B = 0.9$. Since $B \in (0.8;1]$, nonblocking conditions are determined by case 1 and we have $p \geqslant 31$. The worst state of this switching fabric is shown in Fig. 6.15. In each input or output terminal in $\mathbb{SI}_i$ or $\mathbb{SO}_{n-i}$ we can set up five connections of weight b. Each of such connections makes a plane inaccessible for the connection $\langle 0,0,B\rangle$. We have $p_1 = p_2 = 15$ and one more plane is needed for connection $\langle 0,0,0.9\rangle$.

Let us assume now that in this switching fabric connection weights are limited by $b = 0.2$ and $B = 0.8$. Since $B \notin (0.8;1]$, nonblocking conditions a determined by case 3. According to Theorem 6.24: $p \geqslant 25$. The worst case in the switching fabric is shown in Fig. 6.16. The plane will be inaccessible for the connection of weight 0.8 if there is a node on the connecting path, which already carries a connection of

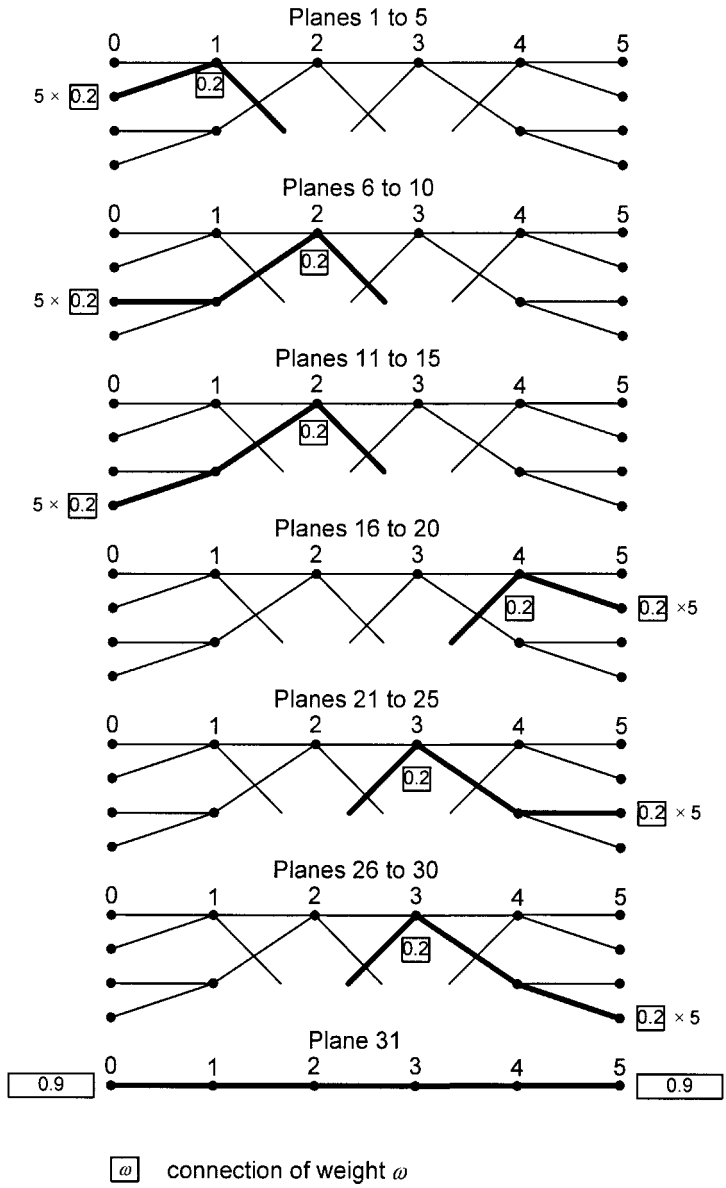


Figure 6.15. The worst state in $\log_2(32, 0, 31)$ with $\beta = 1$, $b = 0.2$ and $B = 0.9$

weight 0.2^+ (0.2^+ denotes a connection of weight $0.2 + \varepsilon$, $\varepsilon \to 0$). Four such connections can be set up in one input terminal ($P(1; 0.2) = 4$), and there is a free bandwidth of weight less than 0.2 ($R_1(1; 0.2) = 0$) and it cannot be used by any other connection at this input terminal.

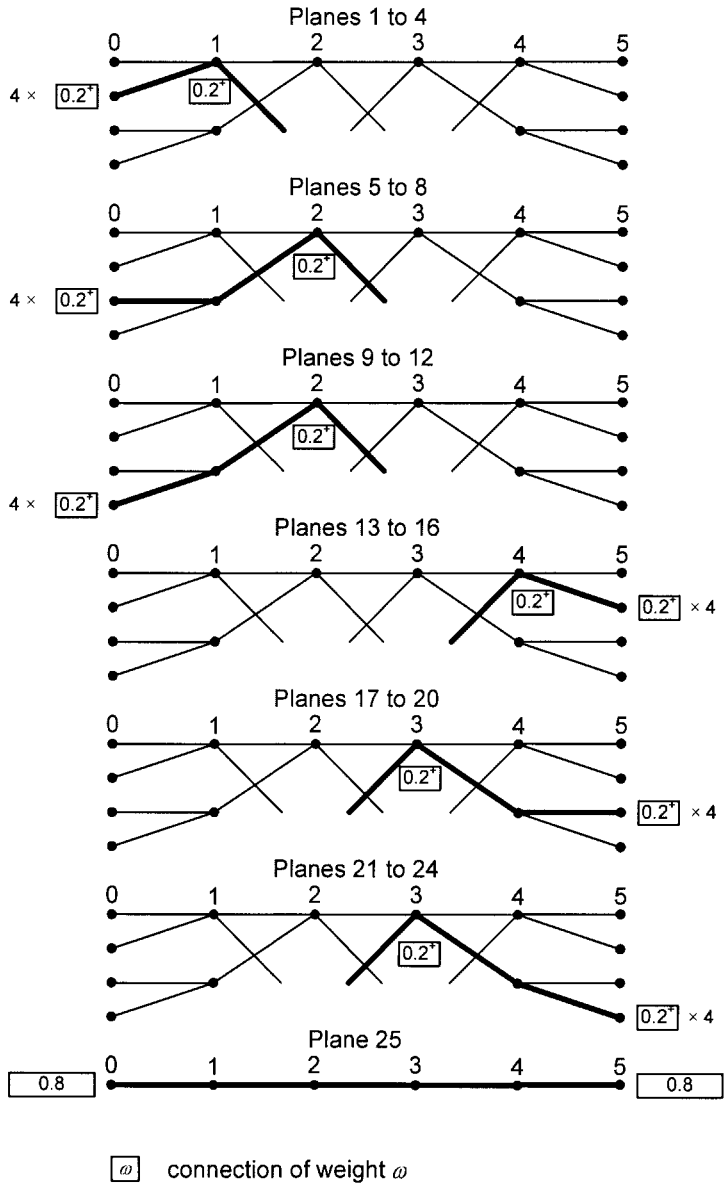


Figure 6.16. The worst state in $\log_2(32, 0, 25)$ with $\beta = 1$, $b = 0.2$ and $B = 0.8$

At both input terminal x and output terminal y, it is possible to set up a connection of weight 0.2, but the plane will be still accessible for the connection of weight 0.8.

Let us now consider the $\log_2(16, 0, p)$ switching fabric with $\beta = 1$, and let connection weights be between $b = 0.1$ and $B = 0.6$. Since $B \notin (0.9; 1]$, nonblocking conditions are determined by case 3. Let the new connection is $\langle 0, 0, 0.6\rangle$. This connection will be blocked in any plane when one of the nodes on connecting path is already used by a connection of weight 0.4^+. At each input terminal we can set up two connections of such weight ($P(1; 0.4) = 2$). We can consider now only $\mathbb{SI}_1 = \{1\}$, and since $\sum_{i=1}^{(n/2)-1} |\mathbb{SI}_i| = L_1 = 1$ than these connections will occupy two planes (planes 1 and 2). The same is true for $\mathbb{SO}_3 = \{1\}$, i.e., connections to output terminal 1 will also occupy two planes (planes 3 and 4). At input terminal 1 we have now free bandwidth available of weight $R_1(1; 0.4) = 0.2$ and it can be used by the next connection (only one since in fact it is less than $2b$). We need 3 such connections to block a plane ($a(1; 0.6) = 3$), since there is only one input terminal with available bandwidth of weight $R_1(1; 0.4)$, it will not block the next plane ($\lfloor R_3(1; 3)\rfloor = 0$). We now have $\alpha_1(1; 1; 0.6) = 0.2$, $R_5(0.4) = 0.4$ and $P(\alpha_1(1; 1; 0.6) + R_5(0.4); 0.4) = 1$. So one more plane may be occupied, and it is plane 5. Such connections to output terminals 0 and 1 may also occupy the same number of planes (plane 6). We have $p_1 = p_2 = 3$, and $R_1(\alpha_1(1; 1; 0.6) + R_5(0.4); 0.4) = 0.2$. At input terminals in $\mathbb{SI}_2 = \{2, 3\}$ four connections of weight 0.4^+ can be set up to output terminals in $\mathbb{SO}_2 = \{2, 3\}$. At each input terminal in $\mathbb{SI}_2$ we have free bandwidth of weight $R_1(1; 0.4) = 0.2$, so two connections of such weigh can be set up, but three such connections are needed to block a plane, since $a(1; 0.6) = 3$ and we have $\lfloor R_3(2; 3)\rfloor = 0$). Now we have $\alpha_1(2; 1; 0.6) = 2 \cdot 0.2 = 0.4$, $P(\alpha_1(2; 1; 0.6); 0.4) = 0$, $p_3 = 4$ (planes from 7 to 10 occupied), and $R_1(\alpha_1(2; 1; 0.6); 0.4) = 0.4$. Since $R_1(\alpha_1(1; 1; 0.6) + R_5(o.4); 0.4) + R_1(\alpha_1(2; 1; 0.6); 0.4) = 0.6$ and $P(0.6; 0.4) = 1$, we have $p_4 = 1$ and one more plane is inaccessible by the new connections. So all together 11 planes are busy and one more plane is needed for the new connection.

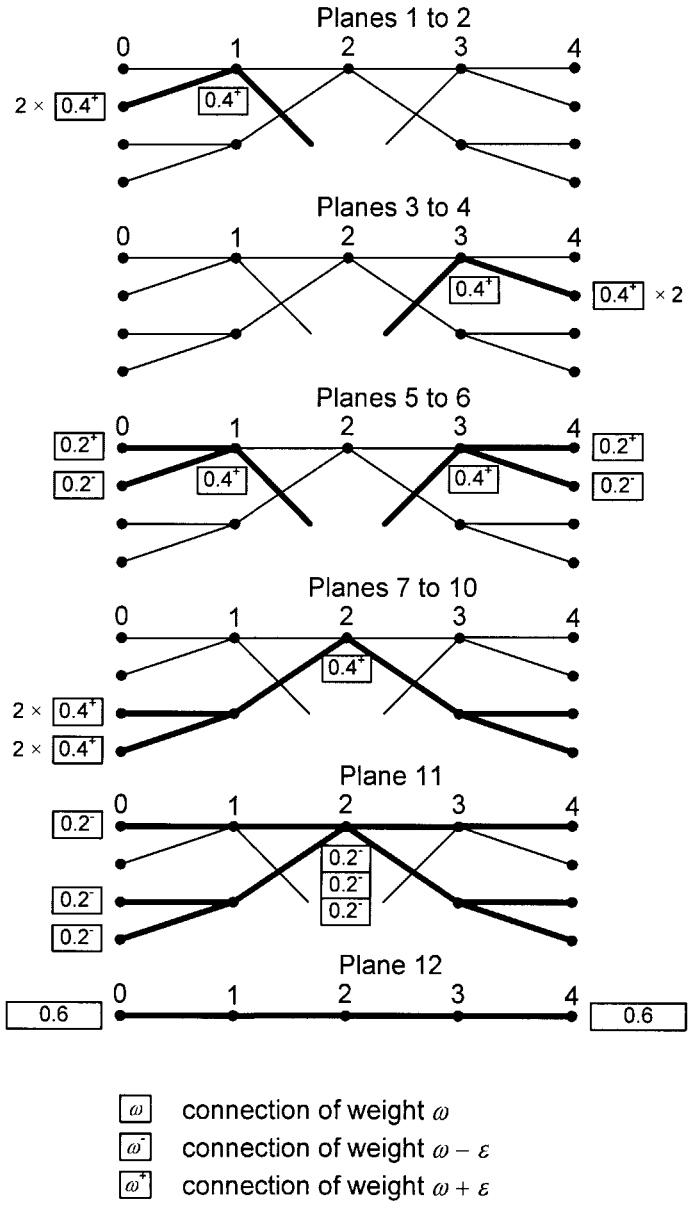


Figure 6.17. The worst state in $\log_2(16, 0, 12)$ with $\beta = 1$, $b = 0.1$ and $B = 0.6$

6.4.2.2 Multicast Connections

Strict-sense nonblocking conditions for multirate broadcast connections can be derived using arguments similar to those given in Theorems 6.23 and 6.24 for discrete and continuous bandwidth cases, respectively, and Theorem 6.8 for space-division switching. These conditions will result in the large number of planes, therefore they are not practical. More practical approach is to use wide-sense nonblocking switching fabrics.

6.4.3 Wide-sense Nonblocking Conditions

Wide-sense nonblocking $\log_2(N, m, p)$ switching fabrics with unicast connections can be obtained using the routing strategy based on functional divisions of planes. In this algorithm one set of planes is used for connections of weights less than or equal to 0.5, and another set of planes is used for connections of weights greater than 0.5. In the case of connections with weights greater than 0.5, only one connection can be set up in one link, and it correspond to the space-division case. The required number of planes is therefore given by Theorem 6.5. For connections with smaller weights, the required number of planes can be obtained by putting $B = 0.5$ in conditions given by Theorem 6.23 or 6.24, depending on whether discrete or continuous bandwidth case is considered.

In the case of multirate multicast connections, similarly as in the space-division switching fabrics, the control algorithms based on blocking windows can be used for vertically replicated banyan-type and extended banyan-type switching fabrics. The first attempt to derive such conditions was made by Kabaciński and Wichary [80]. They consider some cases for continuous bandwidth case, however, the general results for both discrete and continuous bandwidth are still open problems.

References

[1] M. H. Ackroyd, "Call repacking in connecting networks," *IEEE Transactions on Communications*, vol. COM-27, no. 3, pp. 589–591, March 1979.

[2] D. P. Agrawal, "Graph theoretical analysis and design of multistage interconnection networks," *IEEE Transactions on Computers*, vol. C-32, pp. 637–648, 1983.

[3] D. Akopian, J. Takala, J. Saarinen, and J. Astola, "Multistage interconnection networks for parallel viterbi decoders," *IEEE Transactions on Communications*, vol. 51, no. 9, pp. 1536–1545, September 2003.

[4] N. Alon, "A simple algorithm for edge-coloring bipartite multigraphs," *Information Processing Letters*, vol. 85, no. 6, pp. 301–302, March 2003.

[5] S. Andersen, "The looping algorithm extended to base 2^t rearrangeable switching networks," *IEEE Transactions on Communications*, vol. COM-25, no. 10, pp. 1057–1063, October 1977.

[6] G. Balboni, M. Collivignarelli, L. Licciardi, A. Paglialunga, G. Rigolio, and F. Zizza, "From transport backbone to service platform: facing the broadband switch evolution," in *International Switching Symposium ISS'95*, 1995, p. 26.

[7] L. I. Bassalygo, I. I. Grushko, and V. I. Neiman, "The structure of one-sided connectiong networks," in *6th International Teletraffic Congress*, Munich, 1970, pp. 2451–2459.

[8] K. Basu, P. Maveddat, and J. Chen, "Multi-rate services: Challanges in network design," in *Proc. IEEE GLOBECOM '94*, San Francisco, CA, 1994, pp. 644–649.

[9] V. E. Beneš, *Mathematical Theory of Connecting Networks and Telephone Traffic*. New York: Academic Press, 1965.

[10] A. W. Berg and W. Whitt, "Extending the effective bandwidth concept to networks with priority classes," *IEEE Communications Magazine*, vol. 36, no. 8, pp. 78–83, August 1998.

[11] M. Beshai, L. Garret, and I. Steward, "Transposed switching of multi-bit-rate traffic," in *ITC Specialists Seminar*, Adelaide, Australia, 1989, p. paper 12.3.

[12] C. Cardot, "Comments on "A simple algorithm for the control of rearranfeable switching networks," *IEEE Transactions on Communications*, vol. COM-34, no. 4, p. 395, April 1986.

[13] T. Y. Chai, T. H. Cheng, S. K. Bose, C. Lu, and G. Shen, "Array interconnection for rearrangeable 2-D MEMS optical switch," *Journal on Lightwave Technology*, vol. 21, no. 5, pp. 1134–1140, May 2003.

[14] K. S. Chan, S. Chan, and K. L. Yeung, "Wide-sense non-blocking multicast ATM switches," *Electronic Letters*, vol. 33, no. 6, pp. 462–464, 1997.

[15] ——, "Wide-sense non-blocking multicast ATM switches," in *IEEE GLOBECOM '97*, Phoenix, AZ, 1997, p. paper S15.3.

[16] H. J. Chao, "Next generation routers," *Proceedings of the IEEE*, vol. 90, no. 9, pp. 1510–1558, September 2002.

[17] H. J. Chao, K.-L. Deng, and Z. Jing, "PetaStar: A petabit photonic packet switch," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 7, pp. 1096–1112, Spetember 2003.

[18] H. J. Chao, C. H. Lam, and E. Oki, *Broadband Packet Switching Technologies. A Practical Guide to ATM Switches and IP Routers.* John Wiley & Sons Ltd, 2001.

[19] P. Charransol, J. Hauri, C. Athènes, and D. Hardy, "Development of a time division switching network usable in a very large range of capacities," *IEEE Transactions on Communications*, vol. COM-27, no. 7, pp. 982–988, July 1979.

[20] F. M. Chiussi, J. G. Kneuer, and V. P. Kumar, "Low-cost scalable switching solutions for broadband networking: the ATLANTA architecture and chipset," *IEEE Communications Magazine*, vol. 35, no. 12, pp. 44–53, December 1997.

[21] C.-G. Choi, S.-P. Han, B. C. Kim, S.-H. Ahn, and M.-Y. Jeong, "Fabrication of large-core 1×16 optical power splitters in polymers using hot-embossing process," *IEEE Photonics Technology Letters*, vol. 15, no. 6, pp. 825–827, June 2003.

[22] S.-P. Chung and K. W. Ross, "On nonblocking multirate interconnection networks," *SIAM Journal on Computing*, vol. 24, no. 4, pp. 726–736, August 1991.

[23] C. Clos, "A study of non-blocking switching networks," *The Bell System Technical Journal*, vol. 32, no. 2, pp. 406–424, March 1953.

[24] M. Collier and T. Curran, "The strictly non-blocking condition for three-stage networks," in *Proc. 14th International Teletraffic Congress - ITC'94*, Antibes Juan-les-Pins, France, 1994, pp. 635–644.

[25] J. P. Coudreuse, P. Grall, and C. Raphaien, "E 10 system: Time division tandem exchanges," *Commutation et Electronique*, no. Special Issue, pp. 43–56, June 1975.

[26] G. Danilewicz, "Nonblocking multicast switching fabrics (in Polish)," Ph.D. dissertation, Poznan University of Technology, Poznań, Poland, March 2001.

[27] G. Danilewicz and W. Kabaciński, "Nonblocking and rearrangeable symmetrical three-stage multicast switching networks," in *IV Polish Teletraffic Symposium*, Poznań, May 1997, pp. I.2–1–I.2–7.

[28] ——, "Wide sense non-blocking three-stage multirate Clos switching networks with multicast connections – new upper bounds," in *Third IEEE International Workshop on Broadband Switching Systems*, Kingston, Ontario, Canada, June 1999, pp. 75–79.

[29] ——, "Wide-sense non-blocking multi-$\log_2 N$ broadcast switching networks," in *International Conference on Communications ICC 2000*, New Orleans, LA, USA, June 2000, pp. 1440–1444.

[30] ——, "$\text{Log}_2(N, m, p)$ broadcast switching networks," in *International Conference on Communications ICC 2001*, vol. 2, Helsinki, Finland, June 2001, pp. 604–608.

[31] ——, "Wide-sense non-blocking multicast switching networks composed of $\log_2 N + m$ stages," in *IEEE International Conference on Telecommunications ICT 2001*, vol. 1, Bucharest, Romania, June 2001, pp. 519–524.

[32] ——, "Wide-sens and strict-sense non-blocking operation of multicast multi-$\log_2 N$ switching networks," *IEEE Transactions on Communications*, vol. 50, no. 6, pp. 1025–1036, June 2002.

[33] M. de Prycker, *Asynchronous Transfer Mode: Solution for Broadband ISDN*. London: Prentice Hall, 1991.

[34] I. B. Djordjevic, R. Varrazza, M. Hill, and S. Yu, "Packet switching performance at 10 Gb/s across a 4×4 optical crosspoint switch matrix," *IEEE Photonics Technology Letters*, vol. 16, no. 1, pp. 102–104, January 2004.

[35] D. Z. Du, P. C. Fishburn, B. Gao, and F. H. Hwang, "Wide-sense nonblocking for 3-stage Clos networks," in *Advances in Switching Networks*, D. Z. Du and H. Q. Ngo, Eds. Kluwer, 2001, pp. 89–100.

[36] D. Z. Du, B. Gao, F. K. Hwang, and J. H. Kim, "On multirate rearrangeable Cloc networks," *SIAM Journal on Computing*, vol. 28, no. 2, pp. 463–470, 1998.

[37] D.-Z. Du and F. K. Hwang, *Advances in Switching Networks*, ser. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, 1998, vol. 42.

[38] D.-Z. Du and H. Q. Ngo, *Switching Networks: Recent Advances.* Kluwer Academic Publishers, 2001.

[39] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks. An Engineering Approach.* Morgan Kaufmann Publishers, 2003.

[40] A. M. Duguid, "Structural properties of switching networks," Brown University, Progress Report BTL-7, 1959.

[41] J.-P. Faure, L. Noire, and E. Ollier, "A 8x8 optical space-switch based on a novel 8x1 MOEMS switching module," in *IEEE Optical Fiber Conference*, 2000, pp. WX5–1–WX5–4.

[42] J.-P. Faure, L. Noirie, and E. Ollier, "A 8x8 optical space-switch based on a novel 8x1 MOEMS switching module," in *OFC 2001*, 2001, pp. WX5–1–WX5–3.

[43] P. Fishburn, F. K. Hwang, D. Z. Du, and B. Gao, "On 1-rate wide-sense nonblocking for 3-stage Clos networks," *Discrete Applied Mathematics*, vol. 78, pp. 75–87, 1997.

[44] B. Gao and F. K. Hwang, "Wide-sense nonblocking for multirate 3-stage Clos networks," *Theoretical Computer Science*, vol. 182, pp. 171–182, 1997.

[45] P. Giacomazzi and V. Trecordi, "A study of non blocking multicat switching networks," *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 1163–1168, February/March/April 1995.

[46] P. Hall, "On representatives of subsets," *J. London Math. Soc.*, vol. 10, pp. 26–30, 1935.

[47] R. Hernandez, K. Carmichael, C. K. Chai, and G. Cole, *IP Storage Networking: IBM NAS and iSCSI Solutions.* IBM Redbooks, February 2002.

[48] H. Hinton, *An introduction to photonic switching fabrics.* Nowy Jork: Plenum Press, 1993.

[49] M. Hoffman, D. Nüsse, and E. Voges, "An electrostatically actuated 1×2 moving-fiber switch," *IEEE Photonics Technology Letters*, vol. 15, no. 1, pp. 39–41, January 2003.

[50] J. Y. Hui, *Switching and Traffic Theory for Integrated Broadband Networks.* Boston, MA: Kluwer, 1990.

[51] F. K. Hwang and A. Jajszczyk, "On nonblocking multiconnection networks," *IEEE Transactions on Communications*, vol. COM-34, no. 10, pp. 1038–1041, October 1986.

[52] F. K. Hwang, "Rearrangeability of multi-connection three-stage Clos networks," *Networks*, vol. 2, pp. 301–306, 1972.

[53] ——, "Three-stage multicaonnection networks which are nonblocking in the wide sense," *Bell System Technical Journal*, vol. 58, pp. 2183–2187, 1979.

[54] ——, "Control algorithms for rearrangeable Clos networks," *IEEE Transactions on Communications*, vol. COM-31, no. 8, pp. 952–954, August 1983.

[55] ——, "Choosing the best $\log_2(N, m, p)$ strictly nonblocking networks," *IEEE Transactions on Communications*, vol. 46, 1998.

[56] ——, *The Mathematical Theory of Nonblocking Switching Networks.* Singapore: World Scientific, 1998.

[57] F. K. Hwang and S. C. Liaw, "On nonblocking multicast 3-stage Clos networks," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 535–539, 2000.

[58] F. K. Hwang and B.-C. Lin, "Wide-sense nonblocking multicast $\log_2(N, m, p)$ networks," *IEEE Transactions on Communications*, vol. 51, no. 10, pp. 1730–1735, October 2003.

[59] H. Inoue, H. Nakamura, K. Morosawa, Y. Sasaki, T. Katsuyama, and N. Chinone, "An 8 mm length nonblocking 4×4 optical switch array," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 7, pp. 1262–1266, August 1988.

[60] A. Jajszczyk, "A simple algorithm for the control of rearrangeable switching networks," *IEEE Transactions on Communications*, vol. COM-33, no. 2, pp. 168–171, February 1985.

[61] ——, "Comments on three-stage multiconnection networks which are nonblocking in the wide sense," *Bell System Technical Journal*, vol. 62, pp. 2113–2114, September 1983.

[62] ——, "On nonblocking switching networks composed of digital symmetrical matrices," *IEEE Transactions on Communications*, vol. COM-31, no. 1, pp. 2–9, January 1983.

[63] ——, "One-sided switching networks composed of digital switching matrices," *IEEE Transactions on Communications*, vol. COM–35, no. 12, pp. 1383–1384, 1987.

[64] ——, "On combinatorial properties of broadband time-division switching networks," *Computer Networks and ISDN Systems*, vol. 20, pp. 377–382, 1990.

[65] ——, *Photonic Switching*, ser. The Froehlich/Kent Encyclopedia of Telecommunications. Marcel Dekker, Inc., 1997, vol. 14, pp. 355–385.

[66] ——, "Rearrangeability of multicast Clos networks with middle-stage fan-out," *IEEE Communications Letters*, vol. 7, no. 12, pp. 599–600, December 2003.

[67] A. Jajszczyk and G. Jekel, "A new concept - repackable networks," *IEEE Transactions on Communications*, vol. 41, no. 8, pp. 589–591, August 1993.

[68] S. A. Johnson, "A performance analysis of integrated communications systems," *British Telecomunication Technological Journal*, vol. 3, no. 4, pp. 36–45, 1985.

[69] H. S. Jung, "1 × 2 on/off optical power splitters utilizing strain-induced optical waveguides in $LiNbO_3$," *IEICE Transactions on Electronics*, vol. E86-C, no. 5, pp. 762–764, May 2003.

[70] W. Kabaciński, "Comments on 'two-stage nonblocking bidirectional switch networks'," *Electronics Letters*, vol. 25, no. 17, p. 1198, 1989.

[71] ——, "On nonblocking switching networks for multirate connections," in *Proc. 13th Int. Teletraffic Congress ITC'91*, vol. D, Copenhagen, Denmark, June 1991, pp. 885–889.

[72] ——, "On nonblocking switching networks for multichannel connections," *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 222–224, February/March/April 1995.

[73] ——, "Non-blocking asymmetrical three-stage multirate switching networks," in *International Conference on Communication Technology*, vol. 1, Beijing, China, October 1998, pp. S11–11/1–S11–11/5.

[74] ——, "Non-blocking three-stage multirate switching networks," in *Sixth IFIP Workshop on Performance, Modelling and Evaluation of ATM Networks*, Ilkley, UK, July 1998, pp. 26/1–26/10.

[75] ——, "Non-blocking folded architectures for multirate ATM switching fabrics," in *IEEE International Conference on Communications ICC 1999*, vol. 3, Vancouver, Canada, June 1999, pp. 1458–1462.

[76] W. Kabaciński and F. K. Liotopoulos, "Designing generalised, multirate & strictly non-blocking 3-stage Clos switches," in *International Conference on Telecommunications ICT 2000*, Acapulco, Meksyk, May 2000, pp. 400–404.

[77] ——, "Multirate non-blocking generalized 3-stage Clos switching networks," *IEEE Transactions on Communications*, vol. 50, no. 9, pp. 1486–1494, September 2002.

[78] W. Kabaciński and M. Michalski, *Lower Bounds for WSNB Multi-Log_2N Switching Networks*, Poznań University of Technology, Institute of Electronics and Telecommunications, Poznań, Poland, August 2004.

[79] W. Kabaciński and M. Michalski, "Simultaneous connections routing in multi-$\log_2 N$ switching fabrics," in *IEEE Workshop on High Performance Switching and Routing HPSR 2004*, Phoenix, AZ, April 2004, pp. 214–218.

[80] W. Kabaciński and T. Wichary, "Multi-$\log_2 N$ multirate switching networks with multicast connections," in *Polish Teletraffic Symposium*, Kraków, September 2003, pp. 297–317.

[81] W. Kabaciński and M. Żal, "Non-blocking operation of multi-$\log_2 N$ switching networks," in *Third IEEE International Workshop on Broadband Switching Systems*, Kingston, Ontario, Canada, June 1999, pp. 140–144.

[82] ——, "Non-blocking operation of multi-$\log_2 N$ switching networks," *System Science*, vol. 25, no. 4, pp. 83–97, 1999.

[83] ——, "A new control algorithm for rearrangeable multi-$\log_2 N$ switching networks," in *IEEE International Conference on Telecommunications ICT 2001*, vol. 1, Bucharest, Romania, June 2001, pp. 501–506.

[84] S. Kaczmarek, "Characteristics of vertically stacked switching networks," *Telecommunications Review*, vol. LVI, no. 2, pp. 54–56, 1983, (in Polish).

[85] W. H. Kautz, K. N. Levitt, and A. Waksman, "Cellular interconnection arrays," *IEEE Transactions on Computers*, vol. C-17, no. 5, pp. 443–451, May 1968.

[86] T. Kawamura, H. Ichino, M. Suzuki, K. Genda, and Y. Doi, "Over 20 Gbit/s throughput ATM crosspoint switch large scale integrated circiut using Si bipolar technology," *Electronics Letters*, vol. 30, pp. 854–855, 1994.

[87] D. S. Kim and D.-Z. Du, "Multirate multicast switching networks," *Theoretical Computer Science*, vol. 261, pp. 241–251, 2001.

[88] D. G. Kirkpatrick, M. Klawe, and N. Pippenger, "Some graph-coloring theorems with applications to generalized connection networks," *SIAM J. Algebr. Discrete Methods*, vol. 6, pp. 576–782, 1985.

[89] K. D. Kovarik and P. Maveddat, "Mulit-rate ISDN," *IEEE Communications Magazine*, vol. 32, no. 4, pp. 48–54, April 1994.

[90] D. H. Lawrie, "Access and alignment of data in an array processor," *IEEE Transactions on Computers*, vol. 25, pp. 1145–1155, 1976.

[91] C.-T. Lea, "Bipartite graph design principle for photonic switching network," *IEEE Transactions on Communications*, vol. 38, no. 4, pp. 529–538, April 1990.

[92] ——, "Multi-$\log_2 N$ networks and their applications in high-speed electronic and photonic switching systems," *IEEE Transactions on Communications*, vol. 38, no. 10, pp. 1749–1740, October 1990.

[93] ——, "Multirate $\log_2(N, e, p)$ networks," in *IEEE Globecom*, San Francisco, CA, 1994, pp. 319–323.

[94] ——, "Buffered or unbuffered: A case study based on $\log_d(N, e, p)$ networks," *IEEE Transactions on Communications*, vol. 44, no. 1, pp. 105–113, January 1996.

[95] C.-T. Lea and D.-J. Shyy, "Tradeoff of horizontal decompostion versus vertical stacking in rearrangeable nonblocking networks," *IEEE Transactions on Communications*, vol. 39, no. 6, pp. 899–904, June 1991.

[96] H. Y. Lee, F. K. Hwang, and J. D. Carpinelli, "A new decomposition algorithm for rearrangeable clos interconnection networks," *IEEE Transactions on Communications*, vol. 44, no. 11, pp. 1572–1578, November 1996.

[97] T. T. Lee, "Nonblocking copy networks for multicast packet switching," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1455–1467, 1988.

[98] J. Leuthold, R. Ryf, S. Chandrasekhar, D. T. Neilson, C. H. Joyner, and C. R. Giles, "All-optical nonblocking Terabit/s crossconnect based on low power all-optical wavelength converter and MEMS switch fabric," in *OFC'01*, 2001, pp. PD16-1–PD16-3.

[99] S.-Y. R. Li, *Algebraic Switching Theory and Broadband Applications*. Academic Press, 2001.

[100] L. Licciardi and F. Serio, "Technical solutions in a flexible and modular architecture for ATM switching," in *International Symposium on Communications ISCOM'95*, 1995, pp. 359–366.

[101] S. Liew, M.-H. Ng, and C. Chan, "Blocking and nonblocking multirate Clos switching networks," *IEEE/ACM Transactions on Networking*, vol. 6, no. 3, pp. 307–318, June 1998.

[102] G.-H. Lin, "Nonblocking routing properties of Clos networks," in *Switching Networks: Recent Advances*, D.-Z. Du and H. Ngo, Eds. Kluwer Academic Publishers, 2001, pp. 117–141.

[103] G.-H. Lin, D.-Z. Du, X.-D. Hu, and G. Xue, "On rearrangeability of mulitrate Clos networks," *SIAM Journal on Computing*, vol. 28, no. 4, pp. 1225–1231, 1999.

[104] F. K. Liotopoulos and S. Chalasani, "Strictly nonblocking operation of 3-stage Clos switching networks," in *Performance Modelling and Evaluation of ATM Networks*. London: Chapman & Hall, 1996, vol. II.

[105] F. K. Liotopoulos, "Split-connection rearrangeably nonblocking operation of three-stage multirate Clos networks," *Computer Communications*, vol. 25, pp. 1584–1595, 2002.

[106] F. K. Liotopoulos and S. Chalasani, "Semi-rearrangeably nonblocking operation of clos networks in the multirate environment," *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, pp. 281–291, April 1996.

[107] M. Listani and A. Roveri, "Switching structures for ATM," *Computer Communications*, vol. 12, pp. 349–358, 1989.

[108] M. Listanti and L. Veltri, "Non blocking generalized three-stage switching networks," in *IEEE Globecom*, San Antonio, TX, December 2001.

[109] K. H. Liu, *IP over WDM*. England: John Wiley & Sons Ltd, 2002.

[110] X. Ma and G.-S. Kuo, "Optical switching technology comparison: Optical MEMS vs. other technologies," *IEEE Optical Communications*, pp. S16–S23, November 2003.

[111] G. Maier and A. Pattavina, "Design of photonic rearrangeable networks with zero first-order switching-element-crosstalk," *IEEE Transactions on Communications*, vol. 49, no. 7, pp. 1268–1279, July 2001.

[112] G. M. Masson and J. B. W. Jordan, "Generalized multistage connection networks," *Networks*, vol. 2, pp. 191–209, 1972.

[113] C. M. Melas and A. Milewski, "The effect of call routing rules in non-blocking three-stage switching networks," *IEEE Transactions on Communications*, vol. COM-27, no. 1, pp. 150–152, January 1979.

[114] R. Melen and J. S. Turner, "Nonblocking multirate networks," *SIAM Journal on Computing*, vol. 18, no. 2, pp. 301–313, 1989.

[115] ——, "Nonblocking multirate networks," in *Proc. INFOCOM*, 1989.

[116] ——, "Multirate Clos networks," *IEEE Communications Magazine*, vol. 41, no. 10, pp. 38–44, October 2003.

[117] E. J. Murphy, T. O. Murphy, A. F. Ambrose, R. W. Irvin, B. H. Lee, P. Peng, G. W. Richards, and A. Yorinks, "16*times*16 strictly nonblocking guided-wave optical switching system," *Journal on Lightwave Technology*, vol. 14, no. 3, pp. 352–358, March 1996.

[118] V. I. Neiman, "Structure et commande optimales des réseaux de connexion sans blocage." *Annales des Télécommunication*, pp. 639–643, July-August 1969.

[119] P. Newman, "A fast packet switch for the integrated services backbne network," *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1468–1479, 1988.

[120] G. Niestegge, "Nonblocking multirate switching networks," in *Proc. the 5th ITC Seminar*, Lake Como, Italy, 1987.

[121] G. Niestegge and E. Wallmeier, "Traffic analysis of multirate switching networks for broadband ISDN," in *12th International Teletraffic Congress*, Torino, Italy, 1988, p. paper 5.1.A.3.

[122] S. Ohta, "A simple control algorithm for rearrangeable switching networks with time division multiplexed links," *IEEE Journal on Selected Areas in Communications*, vol. SAC-5, no. 8, pp. 1302–1308, October 1987.

[123] Y. Oie, M. Murata, K. Kubota, and H. Miyahara, "Effect of speedup in nonblocking packet switch," in *IEEE ICC '89*, 1989, p. 410.

[124] H. Okayama, Y. Okabe, T. Kamijoh, and N. Sakamoto, "Optical switch array using banyan network," *IEICE Transactions on Communications*, vol. E82-B, no. 2, pp. 365–372, February 1999.

[125] E. Oki and N. Yamanaka, "A high-speed ATM switch based on scalable distributed arbitration," *IEICE Transactions on Communications*, vol. E80-B, no. 9, pp. 1372–1376, 1997.

[126] ——, "Scalable crosspoint buffering ATM switch architecture using distributed arbitration scheme," in *IEEE ATM Workshop*, 1997, pp. 28–35.

[127] ——, "A high-speed tandem-crosspoint ATM switch architecture with input and output buffers," *IEICE Transactions on Communications*, vol. E81-B, no. 2, pp. 215–223, 1998.

[128] E. Oki, N. Yamanaka, and M. Nabeshima, "Performance of scalable-distributed-arbitration ATM switch supporting multiple QoS classes," *IEICE Transactions on Communications*, vol. E83-B, no. 2, pp. 204–213, 2000.

[129] E. Oki, N. Yamanaka, K. Nakai, and N. Matsuura, "Multi-stage switching system using optical WDM grouped links based on dynamic bandwidth sharing," *IEEE Communications Magazine*, vol. 41, no. 10, pp. 56–63, October 2003.

[130] E. Oki, N. Yamanaka, and S. Yasukawa, "Tandem-crosspoint ATM switch srchitecture and its cost-effective expansion," in *IEEE Workshop on Broadband Switching Systems BSS'97*, 1997, pp. 45–51.

[131] M. J. O'Mahony, D. Simeonidou, D. K. Hunter, and A. Tzanakaki, "The application of optical packet switching in future communication networks," *IEEE Communications Magazine*, vol. 39, no. 3, pp. 128–135, March 2001.

[132] D. C. Opferman and N. T. Tsao-Wu, "On a class of rearrangeable switching networks," *The Bell System Technical Journal*, vol. 50, no. 5, pp. 1579–1618, May-June 1971.

[133] P. O'Reilly, "Circuit switching – the switching technology for future broadband networks?" *IEEE Network Magazine*, April 1987.

[134] A. Y. Oruç and A. Thirumalai, "A systematic design of cellular permutation arrays," *IEEE Transactions on Computers*, vol. 38, no. 10, pp. 1447–1451, October 1989.

[135] M. Owen, M. Asghari, I. H. White, K. R. Poguntke, and M. J. Robertson, "Theoretical design of a novel $N \times N$ two-dimensional integrated optical crosspoint switch architecture," *Journal on Lightwave Technology*, vol. 16, no. 3, pp. 380–387, March 1998.

[136] G. I. Papadimitriou, C. Papazoglou, and A. S. Pomportsis, "Optical switching: Switch fabrics, techniques, and architectures," *Journal on Lightwave Technology*, vol. 21, no. 2, pp. 384–405, February 2003.

[137] D. S. Parker, "Notes on shuffle/exchange-type networks," *IEEE Transactions on Computers*, vol. 29, pp. 213–222, 1980.

[138] A. Pattavina, *Switching Theory - Architectures and Performance in Broadband ATM Networks*. John Wiley & Sons, England, 1998.

[139] A. Pattavina and G. Tesei, "Multicast nonblocking switching networks," *IEEE Transactions on Communications*, vol. 50, no. 8, pp. 1240–1243, August 2002.

[140] M. Paull, "Reswitching of connection networks," *The Bell System Technical Journal*, vol. 41, no. 3, pp. 833–855, May 1962.

[141] P. S. Raby and P. J. Banks, "Two-stage nonblocking bidirectional switch networks," *Electronics Letters*, vol. 24, no. 6, pp. 362–363, 1988.

[142] R. Ramaswami and K. N. Sivarajan, *Optical Networks. A Practical Perspective*, 2nd ed. Morgan Kaufmann Publishers, 2002.

[143] J. W. Roberts and A. V. Hoang, "Characteristics of services requiring multi-slot connections and their impact on ISDN design," in *Proc. the 5th ITC Seminar*, Lake Como, Italy, 1987, pp. 96–115.

[144] R. Ryf, J. Kim, J. Hickey, A. Gnauck, D. Carr, F. Pardo, C. Bolle, R. Frahm, N. Basavanhally, C. Yoh, D. Ramsey, R. Boie, R. Geirge, J. Kraus, C. Lichtenwalner, R. Papazian, J. Gates, H. R. Shea, A. Gasparyan, V. Muratov, J. E. Griffith, J. A. Prybyla, and S. Goyal, "1296-port MEMS transparent optical crossconnect with 2.07petabit/s switch capacity," in *OFC'01*, 2001, pp. PD28-1–PD28-3.

[145] M. Schienle, G. Wenger, S. Eichinger, J. Müller, L. Stoll, and G. Müller, "A 1×8 InP/InGaAsP optical matrix switch with low insertion loss and high crosstalk suppression," *Journal on Lightwave Technology*, vol. 14, no. 5, pp. 822–826, May 1996.

[146] R. Sedgewick, *Algorithms in C Part 5. Graph Algorithms.*, 3rd ed. Addison Wesley, 2002.

[147] G. Shen, T. H. Cheng, C. Lu, T. Y. Chai, and S. K. Bose, "A novel rearrangeable non-blocking architecture for 2D MEMS optical space switches," *Optical Networks Magazine*, no. 6, pp. 70–79, November/December 2002.

[148] T. Shibata, M. Okuno, T. Goh, T. Watanabe, M. Yasu, M. Itoh, M. Ishii, Y. Hibino, A. Sugita, and A. Himeno, "Silica-based waveguide-type 16×16 optical switch module incorporating driving circuits," *IEEE Photonics Technology Letters*, vol. 15, no. 9, pp. 1300–1302, September 2003.

[149] D.-J. Shyy and C.-T. Lea, "Rearrangeable nonblocking $\log_d(N, m, p)$ networks," *IEEE Transactions on Communications*, vol. 52, no. 5, pp. 1502–1510, May 1994.

[150] ——, "$\text{Log}_2(N, m, p)$ strictly nonblocking networks," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1502–1510, October 1991.

[151] D. Slepian, "Two theorems on a particular crossbar switching networks," 1952.

[152] M. Stasiak, "An approximate midel of a switching network carrying mixture of differmt multi-channel traffic streams," *IEEE Transactions on Communications*, vol. 41, no. 6, pp. 836–840, 1993.

[153] ——, "Blocking probability in a limited-availability group carrying mixture of different mulit-channel traffic streams," *Annales des Télécommunication*, vol. 48, no. 1–2, pp. 71–76, 1993.

[154] ——, "Combinatorial consideratins for switching systems carrying mulit-channel traffic streams," *Annales des Télécommunication*, vol. 51, no. 11–12, pp. 611–625, 1996.

[155] M. Stasiak, P. Zwierzykowski, and M. Głąbowski, "Blocking probability in the mulit-service switching networks with multicast traffic," in *10th IEEE Mediterranean Electrotechnical Conference MELECON*, vol. 2, Cypres, 2000, pp. 868–871.

[156] T. Takahashi, "Time-slot sequence integrity for $(n \times 64)$ kb/s connection," *Electronics and Communications in Japan, Part-1*, vol. 71, no. 3, pp. 79–88, 1988.

[157] J. Tate, A. Bernasconi, P. Mescher, and F. Scholten, *Introduction to Storage Area Networks*, 2nd ed. IBM Redbooks, March 2003.

[158] J. Tate, G. Colea, I. Gomilsek, and J. van der Pijll, *Designing an IBM Storage Area Network*. IBM Redbook, May 2000.

[159] K. Tawara, K. Hamazato, and Y. Iino, "A time divixion switching network based on time switches," *Review of Electrical Communications Laboratory*, vol. 27, pp. 758–772, September-October 1979.

[160] K.-H. Tsai, D.-W. Wang, and F. K. Hwang, "Lower bounds of wide-sense nonblocking Clos networks," in *Proc. 4th COCOON Conf.*, Taipei, Taiwan, August 1998.

[161] ——, "Lower bounds of wide-sense nonblocking Clos network," *Theoretical Computer Science*, vol. 261, pp. 323–328, 2001.

[162] N. Tsao-Wu, "On Neimans's algorithm for the control of rerrangeable switching networks," *IEEE Transactions on Communications*, vol. 22, pp. 737–742, 1974.

[163] Y. Tscha and K. H. Lee, "Non-blocking conditions for multi-$\log_2 N$ multiconnection networks," in *IEEE GLOBECOM*, 1992, pp. 1600–1604.

[164] Y. Tscha and K. Lee, "Yet another result on multi-$\log_2 N$ networks," *IEEE Transactions on Communications*, vol. 47, no. 9, pp. 1425–1431, September 1999.

[165] Y. Tscha, "Scheduling length for switching element disjoint multicasting in banyan-type switching networks," *Journal of Systems Architecture*, vol. 48, no. 6-7, pp. 175–191, January 2003, [32].

[166] E. Valdimarsson, "Blocking in multirate interconnection networks," *IEEE Transactions on Communications*, vol. 42, no. 2/3/4, pp. 2028–2035, February/March/April 1994.

[167] A. Varma and S. Chalasani, "Asymmetrical multiconnection three-stage Clos networks," *Networks*, vol. 23, pp. 427–439, 1993.

[168] T.-J. Wang, C.-F. Huang, and W.-S. Wang, "Wide-angle 1×3 optical power divider in linbo$_3$ for variable power splitting," *IEEE Photonics Technology Letters*, vol. 15, no. 10, pp. 1401–1403, October 2003.

[169] G. Wenger, M. Schienle, J. Bellermann, M. Heinbach, S. Eichinger, J. Müller, B. Acklin, L. Stoll, and G. Müller, "A completely packaged strictly nonblocking 8 × 8 optical matrix switch on InP/InGaAsP," *Journal on Lightwave Technology*, vol. 14, no. 10, pp. 2332–2337, October 1996.

[170] C.-L. Wu and T.-Y. Feng, "On a class of multistage interconnection networks," *IEEE Transactions on Communications*, vol. C-29, pp. 694–702, 1980.

[171] L. Xu, H. G. Perros, and G. Rouskas, "Techniques for optical packet switching and optical burst switching," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 136–142, January 2001.

[172] N. Yamanaka, K. Endo, K. Genda, H. Fukuda, T. Kishimoto, and S. Sasaki, "320 Gb/s high-speed ATM switching system hardware technologies based on copper-polyimide MCM," *IEEE Transactions on CPMT*, vol. 18, pp. 83–91, 1995.

[173] Y. Yang and M. Masson, "Non-blocking broadcast switching networks," *IEEE Transactions on Computers*, vol. 40, no. 9, pp. 1005–1015, September 1991.

[174] Y. Yang and J. Wang, "On blocking probability of multicast networks," *IEEE Transactions on Communications*, vol. 46, no. 7, pp. 957–968, July 1998.

[175] ——, "Wide-sense nonblocking Clos networks under packing strategy," *IEEE Transactions on Computers*, vol. 48, no. 3, pp. 265–284, March 1999.

[176] S. Yao, S. J. B. Yoo, and B. Mukherjee, "All-optical packet switching for metropolitan area networks: Opportunities and challenges," *IEEE Communications Magazine*, vol. 39, no. 3, pp. 142–148, March 2001.

[177] T.-W. Yeow, K. L. E. Law, and A. Goldenberg, "MEMS optical switches," *IEEE Communications Magazine*, vol. 39, no. 11, pp. 158–163, November 2001.

[178] S. Yorozu, Y. Hashimoto, Y. Kameda, H. Terai, A. Fujimaki, and N. Yoshikawa, "A 40GHz clock 160Gb/s 4x4 switch circuit using single flux quantum technology for high-speed packet switching systems," in *Workshop on High Performance Switching and Routing HPSR'04*, Phoenix, AZ, April 2004.

[179] T. Yoshimura, M. Ojima, Y. Arai, and K. Asama, "Architecture of 1024x1024 three-dimensional micro optical switching systems with self-organized lightwave network," in *OFC'03*, vol. 1, 2003, pp. 50–52.

[180] E. Zegura, "Design and analysis of practical switching systems," Ph.D. dissertation, Washington University, Department of Computer Science, June 1993.

Index