Fast Packet Switching for Integrated Services

Peter Newman

Wolfson College

University of Cambridge

A dissertation submitted for the degree of
Doctor of Philosophy

December 1988

Abstract

As the communications industry continues to expand two current trends are becoming apparent: the desire to support an increasing diversity of communications services (voice, video, image, text, etc.) and the consequent requirement for increased network capacity to handle the expected growth in such multi-service traffic. This dissertation describes the design, performance and implementation of a high capacity switch which uses fast packet switching to offer the integrated support of multi-service traffic. Applications for this switch are considered within the public network, in the emerging metropolitan area network and within local area networks.

The Cambridge Fast Packet Switch is based upon a non-buffered, multi-path switch fabric with packet buffers situated at the input ports of the switch. This results in a very simple implementation suitable for construction in current gate array technology. A simulation study of the throughput at saturation of the switch is first presented to select the most appropriate switch parameters. Then follows an investigation of the switch performance for multi-service traffic. It is shown, for example, that for an implementation in current CMOS technology, operating at 50 MHz, switches with a total traffic capacity of up to 150 Gbits/sec may be constructed. Furthermore, if the high priority traffic load is limited on each input port to a maximum of 80% of switch port saturation, then a maximum delay across the switch of the order of 100 μsecs may be guaranteed, for 99% of the high priority traffic, regardless of the lower priority traffic load.

An investigation of the implementation of the switch by the construction of the two fundamental components of the design in 3 μm HCMOS gate arrays is presented and close agreement is demonstrated between the performance of the hardware implementation and the simulation model. It is concluded that the most likely area of application of this design is as a high capacity multi-service local area network or in the interconnection of such networks.

Preface

I wish to thank my supervisor, Roger Needham, for his encouragement and valuable advice during the course of my research. I would also like to thank Ian Leslie, David Wheeler, David Tennenhouse and David Milway for many useful discussions.

I am also indebted to the following people who have read and commented upon this dissertation: Roger Needham, Ian Leslie, David Wheeler and Brian Robertson.

I would like to make a special note of thanks to my parents and to Dr. Elizabeth Bates for the support and encouragement that they have given me during the period of my research.

Throughout my research I have been supported by a grant from the Science and Engineering Research Council for which I am also grateful.

Except where otherwise stated in the text, this dissertation is the result of my own work and is not the outcome of any work done in collaboration. Furthermore, this dissertation is not substantially the same as any that I have submitted for a degree, diploma or any other qualification at any other university. No part of this dissertation has already been or is being concurrently submitted for any such degree, diploma or any other qualification.

Contents

List of Figures

List of Tables

Glossary of Terms

ATD: Asynchronous Time Division, (see page 5).

ATM: Asynchronous Transfer Mode, (see page 5).

BiCMOS: A high speed implementation technology integrating both bipolar and CMOS devices on the same integrated circuit, (see page 110 and table 8.2).

B-ISDN: Broadband ISDN, (see page 5).

CAD: Computer Aided Design.

CAM: Computer Aided Manufacture.

CATV: Community Antenna Television, i.e. cable TV.

CCITT: The International Telegraph and Telephone Consultative Committee.

CMOS: Complementary Metal Oxide Semiconductor — An implementation technology, (see page 110 and table 8.2).

DTDM: Dynamic TDM, (see page 11).

ECL: Emitter Coupled Logic — A high speed implementation technology, (see page 110 and table 8.2).

FDDI: Fiber Distributed Data Interface — A high speed local area network.

FIFO: First In First Out — A queueing discipline.

GaAs: Gallium Arsenide — A very high speed implementation technology, (see page 110 and table 8.2).

HCMOS: High speed CMOS — An implementation technology, (see page 99).

HDLC: High-level Data Link Control — A popular data link layer protocol.

I/O: Input/Output.

ISDN: Integrated Services Digital Network, (see page 5).

LAN: Local Area Network, (see page 3).

MAN: Metropolitan Area Network, (see page 4).

PABX: Private Automatic Branch Exchange — A private telephone exchange.

STM: Synchronous Transfer Mode, (see page 5).

TASI: Time Assignment Speech Interpolation, (see page 13).

TDM: Time Division Multiplexing, (see page 9).

TTL: Transistor Transistor Logic — An implementation technology.

VCI: Virtual Circuit Indicator, (see page 107).

VLSI: Very Large Scale Integration — An integrated circuit containing a large number of active devices.

Chapter 1

Introduction

For many years voice telephony remained the dominant service supported by the telecommunications networks throughout the world. To support a single class of traffic only a single switching mechanism was required: circuit switching, which was well suited to the characteristics of voice traffic. Within the past twenty years the growth of the computer industry has led to the requirement that telecommunications networks offer computer communications services, originally aimed mainly at file transfer and terminal to host interconnection. Although the circuit switched telephony network has been used to support such applications it has proved inefficient and inadequate thus we have seen the development of packet switched networks to support computer communications traffic. Recently, with the rapid drop in the cost of computer technology has come the requirement to support many more communications services. Examples include voice, high quality audio, broadcast quality video, compressed video, image, facsimile, text and many forms of data transfer. The existing practice of providing a separate network, with its own switching mechanism, for every class of traffic cannot be extended to support the demand forecast for many new communications services. The most flexible solution is that of an integrated communications network with a single switching mechanism capable of handling all classes of traffic. Fast packet switching has been suggested as a possible switching mechanism to support integrated services.

If fast packet switching is to support the growth in demand for communications services for the foreseeable future, two major problems require attention. First the design and implementation of a fast packet switch must be investigated, capable of expansion from small sizes of switch up to structures of very high capacity, to handle the expected growth in traffic (especially if video services are contemplated). Secondly the switch must be capable of satisfying the delay requirements of the delay sensitive classes of multi-service traffic. These form the two major issues addressed in this dissertation.

1.1 Objectives

There are many possible approaches to the design of a fast packet switch. The work presented in this dissertation investigates a design which leads to a very simple hardware implementation. A simple implementation offers flexibility, a wide range of potential applications and operation at both conventional speeds and also at possibly very high speeds. The design features a number of parameters that have an effect upon the performance of the switch. Several techniques are also described to enhance the performance beyond that of the basic design. The first objective is thus to characterise the effect of the various parameters upon the performance of the switch. This allows the selection of the preferred design parameters.

The second objective is to characterise the performance of the switch for multi-service traffic. This objective has proven more difficult to satisfy mainly because of the problem of adequately defining the characteristics of multi-service traffic. The performance of the switch for telephony voice traffic, which is well characterised, has been investigated in detail. A simple model of multi-service traffic has also been used to investigate a statistical upper bound on the delay performance of the switch. The effect of packet length on the delay and throughput performance has been measured and some observations made on the packet loss probability due to buffer overload.

A third objective of this work has been the implementation of the two fundamental components of the switch design in current gate array technology. This provides an insight into the complexity of the design and its suitability for implementation in the various available logic families. It also lays the foundation for the development of an experimental model of the switch which will be required for future experimental work in the use of fast packet switching techniques for communications applications.

1.2 Outline

For a concise presentation of the design of the Cambridge Fast Packet Switch and of the major results discussed in detail in this dissertation the reader is referred to [118] or [117]. The work has also been presented at the conference "IEEE Infocom '88" [116] and at the 'European Telecommunications Workshop' [115].

The remainder of this chapter presents an introduction to the growth of telecommunications networks and discusses the requirement for the integrated support of multi-service traffic. Chapter two considers the switching mechanisms capable of supporting this requirement within a high capacity switch implementation and presents the argument for selecting fast packet switching. Some of the fundamental characteristics of a fast packet switch are also discussed. In chapter three a simple classification of fast packet switch designs is introduced followed by a review of many of the major designs of fast packet switch available in the literature. Many of these designs are based upon the use of a multi-stage interconnection network which also forms a central feature of the design of the Cambridge Fast Packet Switch. Chapter four therefore presents a review of interconnection networks concentrating upon those

most relevant to the design of a fast packet switch. The design of the Cambridge Fast Packet Switch is presented and discussed in chapter five. Chapter six presents the results of a simulation study of the performance of the switch fabric in order to gain an insight into the effect of the various switch parameters on performance and to select the appropriate switch design parameters. Chapter seven introduces the requirements that multi-service traffic imposes upon the switch and presents the simulation results of various aspects of a simple model of multi-service traffic applied to the switch. A detailed simulation study of the switch performance for telephony voice traffic in the presence of saturated data traffic is also investigated as a specific example of multi-service traffic of practical interest. In chapter eight the details of the experimental hardware implementation of the two major components of the switch design in current gate technology are presented. Performance measurements of the hardware model are compared to the simulation results and the extension of the model to a full-scale switch implementation is discussed. Finally, chapter nine summarises the insight gained from this study, introduces some ideas for the support of multicast traffic, and discusses some of the problems involved in the networking of fast packet switches that remain for further study.

1.3 Growth of Communications Networks

Local Area Networks

A local area network (LAN) connects computers, workstations, terminals, printers and related peripheral equipment across a distance of up to approximately 1 km [72]. Although research into local area network design began over ten years ago, to some extent stimulated by distributed computing applications [109], major commercial exploitation has developed over the last five years or so. As the cost of computer based equipment has fallen, so the requirement for interconnection within the local area has grown with the major area of commercial application being office automation. The vast majority of traffic currently carried on commercial LANs consists of file transfer and interactive data traffic but much research has been aimed at supporting other classes of traffic on the LAN especially voice traffic [121, 106, 38, 144, 3]. The local area network is privately owned and maintained which has encouraged work aimed at the integration of the LAN and the private telephone exchange (PABX) with varying levels of success [42, 151, 152, 46].

The traffic capacity of current commercial LANs is in the region 1 – 10 Mbits/sec. With the geographical constraint of about 1 km, LANs must be interconnected via bridges to extend the capacity and area of coverage. Work on the transparent interconnection of LANs via bridges is well advanced [10, 133]. The extent to which the traffic capacity may be increased by bridging between multiple LAN segments is limited. Thus to interconnect large numbers of LANs and to support emerging wideband services, such as image and video, high speed local area networks are being developed for use as a backbone network [68, 67, 90, 129, 130]. These networks offer a bandwidth of the order of 100 Mbits/sec. This represents a considerable bandwidth

for conventional computer communications applications but if the cost of workstations that support graphics and image applications continues to fall, and with the possible growth of video applications, a requirement to interconnect such high speed networks may develop. Unless traffic is highly localised, the interconnection of high speed LANs in a mesh topology, using simple bridges, will not greatly increase the capacity of the overall network. Thus the use of a fast packet switch as a high capacity multi-port bridge, to support the interconnection of high speed LANs, (and also as a high speed local area network itself,) forms a possible area of application for the use of fast packet switching technology. Furthermore, the high speed LANs that require interconnection need not be local to each other. Thus fast packet switching provides a mechanism whereby widely separated local area networks may be interconnected, at high capacity, to give the impression of a single virtual LAN spanning a very large area.

Metropolitan Area Networks

The metropolitan area network (MAN) is a public network which may be considered as an extension of the high speed local area network to encompass an urban area with a diameter of up to about 50 km [81]. It must also be capable of supporting multi-service traffic with reasonable efficiency and particularly the voice service. A number of network designs have been proposed [138, 62, 98], but the design most likely to be selected for public service is based on a dual bus arranged as a loop. It has the property that access to the network emulates the action of a single queue although sources are distributed across the network [103, 119]. The integration of multi-service traffic is currently proposed in a hybrid manner. Integration is achieved at the access and transmission level but separate circuit and packet switches are used for compatibility with the existing digital telephony network. Packet traffic, however, is fragmented and transported within short, fixed length packets which permits evolution to the support of multi-service traffic upon a single integrated switching mechanism should this prove desirable.

A single network segment cannot possibly support the evolving needs of an entire urban community thus many segments must be interconnected by means of a switch [29]. Thus as the demand for packet based communications traffic grows we find another application for a fast packet switch that offers the flexibility to support growth to a very high capacity and the ability to handle multi-service traffic.

Public Wide Area Networks

For many years the public telephone network has been evolving from analogue to digital transmission and switching techniques [127]. In the developed countries penetration of digital techniques into the trunk transmission and switching network is now very high and attention is being focussed upon the local telephone network. Digital access from the subscriber to a digital local exchange is forecast to encourage the development of new telecommunications services at an acceptable cost. The in-

tegrated services digital network (ISDN) [125] promises to provide integrated access through a common, standard interface to both circuit and packet switched networks. The circuit switched channels will offer a bandwidth of 64 kbits/sec and access to the packet network will be at up to 64 kbits/sec. The basic rate interface will offer two 64 kbits/sec circuit switched channels with a 16 kbits/sec packet switched signalling channel. In Europe, primary rate access will offer 30 circuit switched channels with a 64 kbits/sec packet switched signalling channel. Wideband access to (N×64 kbits/sec) channels is being considered for later implementation and standards have been developed for packet based user-network signalling.

The philosophy of the ISDN is to employ the existing copper cable between the subscriber and the local exchange for digital access. The possibility of gradually replacing the copper connection with optical fibre is currently under consideration with the opportunity of increasing the bandwidth between the subscriber and the local exchange to many hundreds of Mbits/sec. Such a network is referred to as the broadband ISDN (B-ISDN) [157, 91] and may be required to support services such as: video telephony; image, video and hi-fi audio retrieval services; and the distribution of high definition television [135, 9]. The economic feasibility of the evolution to a widespread broadband network is at present uncertain. However, active consideration is being given to developments in the switching and transmission technology that would be required.

Two approaches have been proposed for the realisation of the broadband ISDN: synchronous transfer mode (STM) and asynchronous transfer mode (ATM), also called new transfer mode [101]. STM is an extension of traditional circuit switching principles and provides channels of fixed bandwidth with a packet switched signalling mechanism. ATM, however, is based upon a fast packet switching mechanism which can provide channels with a bandwidth that is highly variable throughout the lifetime of a connection. In the parlance of broadband ISDN the term 'fast packet switch' implies the use of variable length packets, whereas fast packet switching with short, fixed length packets is called asynchronous time division (ATD). The Cambridge Fast Packet Switch is equally suited to handling both short, fixed length packets and variable length packets of any reasonable size without loss of efficiency. Thus the more general term 'fast packet switching' is used in this dissertation to cover both applications.

ATM offers the major advantage of flexibility over the STM approach. The traffic characteristics of future service requirements cannot be predicted in advance thus the more flexible the network, the easier it becomes for a network administration to offer new services. It is not certain, however, that all services may be supported across an ATM switching mechanism, e.g. distribution video. Also, ATM technology may not be fully developed within the timescale which may be required by early broadband implementations. Hybrid solutions have therefore also been proposed in which both STM and ATM are supported over the same access interface and transmission link [161]. One solution proposes STM channels at about 150 Mbits/sec for the distribution of entertainment video services with a full-duplex ATM channel also at 150 Mbits/sec for all other services [91]. Broadband ISDN therefore forms a

further application of fast packet switching techniques with the dual requirement of very high capacity and the ability to support a large number of services.

Integration

Three distinct levels of integration may be recognised in the evolution of telecommunications networks: access, transmission and switching. The current ISDN provides integrated access to circuit and packet switching networks that remain totally separate. Thus no sharing of resources between the networks is possible, two networks must be separately maintained and the support of future services is limited by the characteristics of the individual networks. Integration at the transmission level continues to require separate circuit and packet switches but the bandwidth of the transmission links connecting the switches is shared dynamically between the two switching mechanisms. Switches of this nature are referred to as hybrid switches and much work has been done on the integrated transmission of circuit voice and packet data services [32, 48, 93, 155]. The circuit switched component offers low delay and low variance of delay which is a requirement of the voice service in current public networks [102] while the packet switched component handles bursty services.

Only when a single switching mechanism handles all classes of traffic is integration at the switching level achieved [146]. Such a network offers integration of access, transmission and switching and may be considered fully integrated. The greatest advantage of full integration is the flexibility to adapt quickly to the changing traffic requirements of new communications services. Other advantages include transmission efficiency, independence of the switching mechanism from the characteristics of the source traffic, and the need to support and maintain only a single integrated network. Fast packet switching offers one possible solution for a fully integrated network.

1.4 Multi-Service Traffic

A simple classification of multi-service traffic is presented in table 1.1 which is adapted from [84] and [83] which itself reflects current CCITT[1] thinking. The natural rate indicates the source bit rate of the traffic class and in some cases assumes a certain amount of compression to reduce redundancy in the signal. Some sources emit traffic continuously at a single bit rate but many exhibit bursty behaviour in which traffic is emitted in bursts interspersed with idle periods. The burstiness of a source is expressed as the ratio of the peak to average bit rates. If the communications channel is fast enough to avoid being a bottleneck then most forms of data traffic become bursty due to user behaviour and the need to share processors amongst applications. Most forms of non-data traffic are also bursty if coded by efficient signal processing technology as the information content of the signal varies with time. A further parameter of the source traffic is the holding time of a connection. The relationship between the range of bit rates and holding times for various classes of traffic and switching

[1] The International Telegraph and Telephone Consultative Committee.

Service Class	*Service*	*Natural Rate bits/sec*	*Burstiness*
Conversation	Telephony	4–64k	2–3
	Video Telephony	2–34M	1–5
	Interactive Data	1k–1M	>10
	Telemetry	<10k	>10
Mail	Voice Mail	4–64k	2–3
	Video Mail	2–34M	2–3
	Text	1k–1M	1–10
	Facsimile	10k–1M	1–10
	Mixed Mode	100k–10M	1–10
File Transfer	Bulk Data	1M	1–10
	Program Download	1M	1–10
	CAD/CAM	1–40M	1–10
Retrieval	Hi-Fi Audio	1–2M	(2)
	Video	2–34M	2–3
	Mixed Mode Document	100k–10M	1–10
	Data	1M	1–10

Table 1.1: Multi-service traffic characteristics.

mechanisms is illustrated in fig. 1.1 which is taken from [91].

Each class of source traffic also exerts various requirements on the performance of the communications network: set-up delay, bit error rate and information delay. The set-up delay is the time required to establish a connection across the network. Estimates of the bit error rate required to support the various services vary widely but for a data service a residual error rate of better than 10^{-12} may be required. For the various real-time services such as voice and video the delay requirements may not permit the use of an error detection and correction protocol but due to the redundancy of the signal a higher bit error rate may be tolerated. The information delay is the delay requirement from source to destination across the network of which several measures may be significant: mean, jitter and percentile. Some services may tolerate high and variable delay across the network. Other services, however, such as telephony, place stringent requirements upon the upper bound of delay and the delay jitter, (variance of delay across the network.) These must be maintained throughout the duration of the connection else the establishment of the connection should be refused. This requires that the network be aware of the bandwidth required by a connection request and be capable of ensuring that this bandwidth is available before granting the connection. This problem will be considered in greater detail in chapter 7. Further discussion of multi-service traffic and of the services under consideration for support by the broadband ISDN may be found in [135, 9, 8, 92, 156, 142].

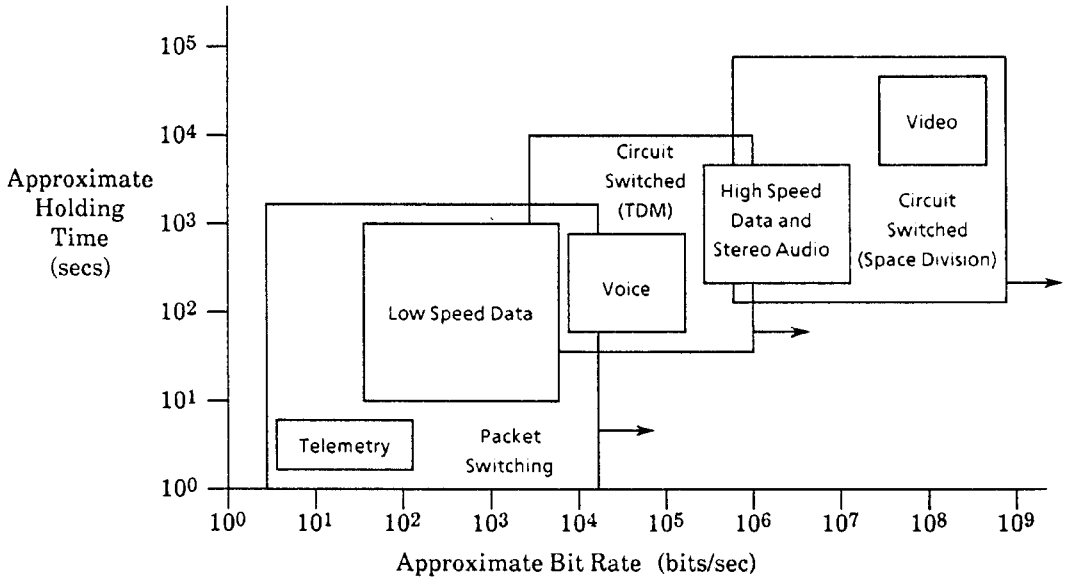

Figure 1.1: Relationship between bit rate and holding time for various classes of traffic and switching mechanisms.

Chapter 2

Statistical Switching Mechanisms

In this chapter a review of the available switching mechanisms is presented and it is argued that a statistical switching mechanism is best suited to the requirements of the high capacity switching of multi-service traffic. Fast packet switching is selected for further study largely on the grounds of its flexibility and the development of the fast packet switch is traced. Some of the fundamental characteristics common to all designs of fast packet switch are then introduced.

2.1 Multiplexing

Multiplexing is the technique whereby two or more separate communications channels are supported across a single transmission medium. A well known example from the telephone network is the support of multiple telephone conversations on a single high bandwidth trunk [22]. Early multiplexing systems for use in the analogue telephone network employed frequency division multiplexing (FDM) in which each separate channel was transmitted at a different carrier frequency. An analogous technique currently being developed for use in optical communications systems is that of wavelength division multiplexing (WDM) in which the various channels are carried on different optical wavelengths. In digital communications systems by far the most common form of multiplexing is that of time division multiplexing (TDM). In this technique the entire capacity of the shared transmission medium is allocated to each source in turn for a short duration sufficient for the source to transmit a brief burst of information of fixed length. As an example, a current European TDM transmission standard employs a 2.048 Mbits/sec digital carrier divided into frames of length 125 μsec, fig. 2.1. Each frame is divided into 32 timeslots each of length 8 bits (one octet). In every frame, timeslot 0 is used for synchronisation and maintenance purposes, timeslot 16 is allocated to signalling and all other timeslots may be allocated to traffic sources. When allocated a channel, the source is given the timeslot number and it fills the appropriate timeslot in every frame with 8 bits of data. Each channel

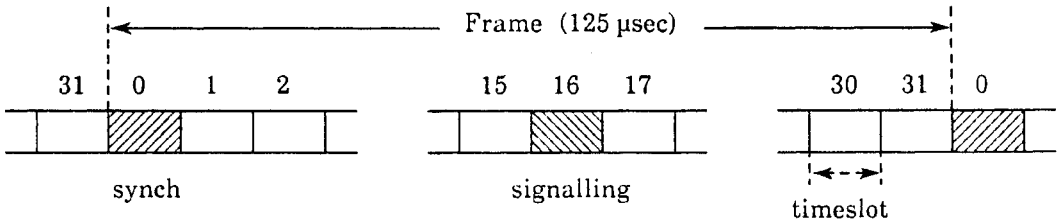

Figure 2.1: An example of time division multiplexing (TDM).

thus carries 64 kbits/sec of traffic.

Time division multiplexing offers channels of fixed bandwidth and is well suited to continuous traffic sources of fixed bit rate, e.g. 64 kbits/sec voice, but many traffic sources are bursty and offer an instantaneous bit rate that is widely variable. TDM is very inefficient in its use of bandwidth for bursty services or for variable bandwidth services so statistical multiplexing has been introduced to overcome this inefficiency. In statistical multiplexing the channels are no longer of fixed bandwidth but each source receives as much transmission capacity as it requires instantaneously. As in TDM, sources continue to transmit information in bursts but these bursts are not necessarily of equal length and sources may submit bursts of information in any order and at a rate that reflects the instantaneous bandwidth required. Sources generally queue for access to the shared transmission medium on a first come first served basis but some sources may be allocated priority. In TDM, when a bursty source is temporarily idle the bandwidth allocated to it is unused but is not available to other sources whereas with statistical multiplexing a bursty source only consumes transmission bandwidth when it has information to send. In conventional TDM the identity of every channel is implicitly conveyed in the position of its timeslot within the frame but with statistical multiplexing the identity of each channel must be explicitly prefaced to every burst of information. This additional overhead tends to require that the information bursts of statistical multiplexing be much longer than those of conventional TDM.

In conventional TDM the offered traffic load can never exceed the capacity of the shared transmission medium, a utilisation of 100% may be supported indefinitely, delay is deterministic and jitter is very low. With statistical multiplexing, however, for short periods the offered traffic load can exceed the capacity of the transmission medium which may result in loss of information, delay or both. Statistical multiplexing cannot support an average utilisation of 100% on the transmission medium and 80% is a maximum utilisation frequently quoted. Delay is dependent upon the mean traffic load and the source traffic characteristics and jitter may be high. Despite these apparent disadvantages, statistical multiplexing is very flexible, supports traffic sources which vary widely in their bandwidth requirements and source traffic characteristics and handles bursty sources efficiently [149, 50].

One proposal of statistical multiplexing for use in the asynchronous transfer mode of broadband ISDN has been termed asynchronous time division (ATD) [141, 31]. The

capacity of the shared medium is divided into short fixed length blocks called cells of 128 bits each. Cells are allocated to traffic sources statistically on demand and each cell contains a short header to identify the source. No framing is applied to the transmission medium but empty cells are filled with a synchronisation pattern by which synchronisation across the transmission link is maintained. A sufficient supply of empty cells is guaranteed as on average the transmission medium will not be utilised beyond about 80%. Another proposal named dynamic TDM (DTDM) retains the frame and timeslot structure of TDM, with each timeslot containing a single cell, but allocates the timeslots statistically as in ATD [161]. A more flexible TDM multiplexing strategy for optical fibre links is under consideration named SONET [15]. It is capable of supporting both synchronous transfer mode (conventional TDM) and asynchronous transfer mode (statistically multiplexed) payloads. In a similar manner the multiplexing strategy under consideration for the STM proposal for broadband ISDN is also capable of supporting both STM and ATM payloads [37].

2.2 Switching Mechanisms

Multiplexing allows the sharing of a high capacity communications link between many channels; but in order to achieve communication between source and destination across a network, a switching function is necessary. A range of switching mechanisms is available to accompany the various multiplexing techniques [84, 127, 25], from circuit switching for use with conventional time division multiplexing (or synchronous transfer mode) to packet switching which mates with the extreme end of statistical multiplexing (or asynchronous transfer mode). Between these two extremes lies a spectrum of available switching mechanisms illustrated in fig. 2.2 which is adapted from [84] and [37]. Switching mechanisms towards the left of the diagram offer channels with fixed bandwidth but a constant and small delay whereas those towards the right of the diagram offer variable bandwidth channels but with a variable delay which can be quite high [59]. Towards the centre of the diagram the statistical switching mechanisms attempt to provide the variable bandwidth required by bursty and variable rate sources but at low delay and low variance of delay compared with conventional packet switching.

Circuit Switching

Circuit switching is based upon the concept of a connection. A connection is an association between a source and its destination across a switched network. A connection may support communication in only a single direction or may offer both forward and reverse channels. A unicast, or point-to-point connection, is established between a single source and a single destination whereas a multicast, or distributive, connection may connect a single source to many destinations. Multicast connections will not be considered further until chapter 9. In circuit switched networks a communications channel of fixed bandwidth is exclusively allocated to a connection throughout the lifetime of that connection. A circuit switch, in general, connects input channels to

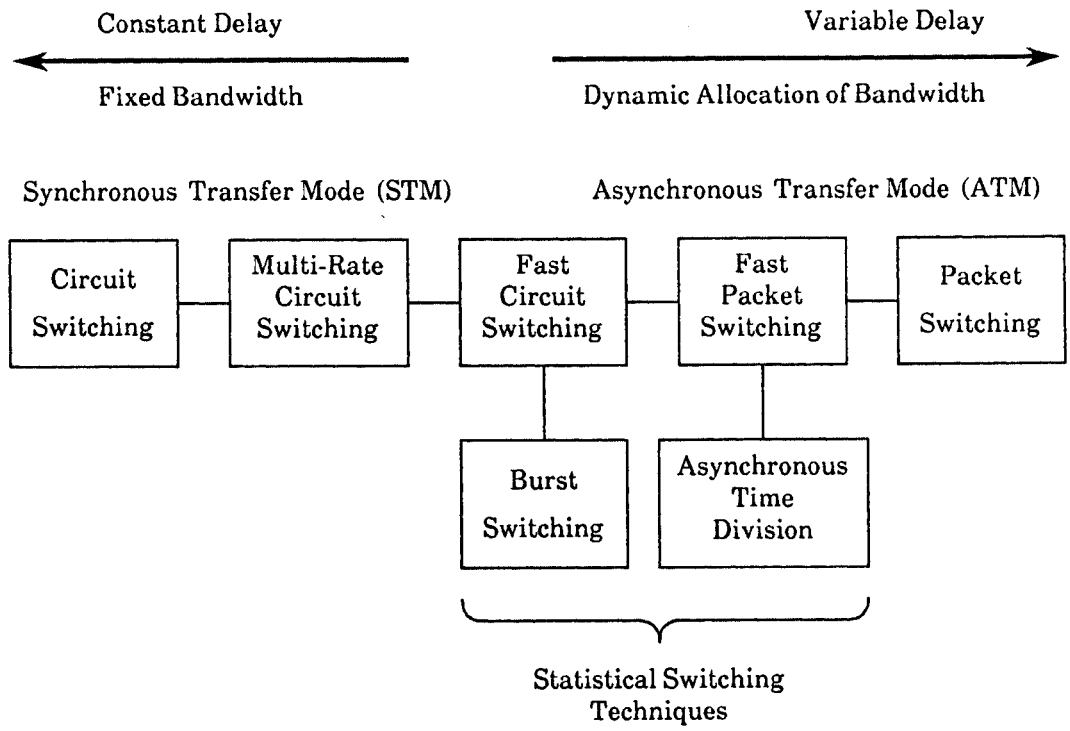

Figure 2.2: The spectrum of switching mechanisms.

output channels, (although the interconnection of bi-directional channels is also possible.) If each channel is presented to the switch on a separate transmission medium the switching is said to occur in the space domain and a connection is allocated a physical path across the switch [162, 105]. If all of the channels are presented to the switch in a TDM multiplex on a single incoming and outgoing transmission medium, see fig. 2.1, then switching occurs in the time domain. In this case a connection represents an association between a timeslot on the incoming link and a timeslot on the outgoing link. During each frame the relevant timeslot is copied from the incoming link into a buffer and thence to the required timeslot on the outgoing link. This requires the buffering of one complete frame and inserts a constant delay of up to one frame length in the connection. In general, a digital circuit switch will be required to interconnect a number of separate incoming and outgoing transmission links, each of them TDM multiplexed with a number of channels. Thus both time and space domain switching is involved [127]. Small switches of up to 1000 or so 64 kbits/sec channels may be implemented using shared memory whereas larger switches will require an interconnection network [160, 34].

Circuit switching offers channels of fixed bandwidth with a low and constant delay. It is transparent in that once a connection is established, the network takes no notice of the information it carries. In particular the network makes no attempt to correct transmission errors that may occur. Overload is handled by refusing to accept further connection requests once the system capacity is fully allocated. This means

that a connection once established will not suffer degradation of delay or bandwidth due to overload. Circuit switching is best suited to applications that require a fixed bandwidth, a low delay, and in which the call holding time is long compared to the call set-up time. It cannot effectively support the widely varying bit rates of many communications services at their natural rate. It does not exploit the burstiness of many forms of information and is inefficient for bursty traffic.

Multi-Rate Circuit Switching

Multi-rate circuit switching is a slight enhancement of circuit switching in that channels of different but fixed bandwidth may be formed by combining one or more integer multiples of some basic channel rate. Selecting the basic channel rate, however, poses a problem in order to satisfy the needs of both low and high bandwidth services. Multiple basic channel rates may be employed but this tends to complicate the design and control of the switch. Synchronising all of the basic rate streams that form a multi-rate channel also poses a significant problem as in general with a circuit switched network no guarantee is offered as to the relative delay between timeslots switched across the network. Neither is there any guarantee that any such delay will remain constant for the duration of a connection. The main disadvantage, however, is that multi-rate circuit switching does not handle bursty sources any more efficiently than does circuit switching.

Fast Circuit Switching

Fast circuit switching has been proposed as a means of handling bursty traffic. If the set-up of a connection across the switch is sufficiently fast, then a connection may be set up for each burst of traffic as it arrives and released immediately the burst ends. Thus the bandwidth of the switch is only allocated to active sources. The technique is similar to that of time assignment speech interpolation (TASI) [20, 153] which was a multiplexing technique used on an expensive analogue transmission link to allocate voice sources to transmission channels only during periods of speaker activity (or talkspurts). The method provided a significant increase in the number of voice sources that could be handled by a transmission link without substantial loss of quality provided a sufficient number of sources were multiplexed.

It is inefficient to set up the entire connection on the arrival of each burst of information, thus burst switching [6, 61] introduces the virtual circuit. A virtual circuit is a logical connection between source and destination which dissociates the concept of the connection from the bandwidth allocated to it. In burst switching a virtual circuit is set up at the beginning of a call which defines the connection but bandwidth is only allocated to that connection at the arrival of each burst of traffic. Buffering is also introduced so that if bandwidth is not available on the arrival of a burst it may be delayed until bandwidth becomes available. For bursts of voice traffic, information is discarded once a burst becomes delayed for longer than 2 msec as it is no longer of any use. As a burst is always transmitted at the same bit rate as

that at which it is received, there is no need to store the complete burst. It can be forwarded as soon as transmission bandwidth becomes available.

The interest in burst switching has so far proven somewhat limited with most of the work being undertaken by a single telecommunications manufacturer. Emphasis appears to be directed towards the switching of 64 kbits/sec voice in the presence of data traffic [122, 123]. The switching mechanism could be made more flexible if multiple channel rates were available for burst switching, as in multi-rate circuit switching, but this would complicate the design and operation of the switch.

Packet Switching

Turning to the other end of the spectrum of switching mechanisms we find conventional packet switching [140, 27]. In packet switching the bandwidth of the transmission medium is no longer divided into channels but the bandwidth of the entire medium is available to every burst of information from each source. Each information burst is constrained to a maximum length and additional fields of control information are added to identify source and destination and to support flow and error control etc. The resulting unit of information is called a packet. The maximum length of a packet is limited by the buffering requirements of packet switches and the packet delay requirements. It should not be too small, however, due to reasons of bandwidth efficiency as the overhead of control information can be quite considerable and is added to every packet. Packets are generally stored in every packet switch in the path and are not forwarded until completely received although suggestions such as cut-through and virtual cut-through [77, 73] have been made to forward packets before they are completely received. Error checking and flow control protocol operations are performed on a link-by-link basis between every packet switch in the path and error correction may be performed both by retransmission from the preceding switch in the path and also on an end-to-end basis.

Packet switched networks offer two fundamental modes of operation: connection-oriented and connectionless. In connection-oriented mode a virtual circuit is established across the path between source and destination. In general, all packets belonging to the same virtual circuit follow the same route across the network which means that the routing operation only has to be performed once when the virtual circuit is set up. The processing of subsequent packets may thus be simplified, the packet header may be simplified, and flow control may be applied more efficiently and selectively to virtual circuits. In connectionless operation each packet, called a datagram, is handled individually and bears enough control information to completely identify it, its source and its destination. Packets between the same source and destination may follow different routes and packets may not be guaranteed to arrive in the same sequential order in which they left. Connectionless operation requires more processing for every packet and flow control is less selective but it is less vulnerable to node failures and more easily adapted to changing traffic patterns. Connection-oriented mode is favoured by telecommunications administrations while connectionless operation is generally preferred by computer communications manufacturers.

Various experiments in supporting the voice service over wide area packet switched networks have been reported [154, 49] but the high delay and high variance of delay over such networks requires complex resequencing procedures to reconstruct the voice signal which themselves insert further delay [11, 104, 108]. In addition, the public voice service requires a number of very large switches both in the total traffic capacity and in the number of switch ports, the support of which is beyond the ability of current designs of conventional packet switch. The support of the voice service over local area packet switched networks has perhaps been more successful [39, 46, 94, 106, 7] but even here the large maximum packet length permitted in many local area networks can introduce a large variance of delay for the voice signal reconstruction algorithm to handle.

Packet switching offers a very flexible communications facility supporting any arbitrary data rate up to the full rate of the transmission medium by selecting the size of the packet and the frequency with which packets are sent. It is also very efficient for handling bursty services and does not consume switching or transmission bandwidth during the idle periods of a call. It responds very rapidly to variations in the bandwidth required by sources during the active phases of a call and can interconnect sources and destinations operating at different data rates. Due to the large amount of processing per packet at every switch, conventional packet switches in general offer a much lower maximum capacity than circuit switches of comparable complexity. They also suffer from high delays across the network and a high variance of delay and it is to answer these drawbacks that fast packet switching has been proposed.

Fast Packet Switching

Fast packet switching attempts to retain the flexibility of conventional packet switching while reducing the delay and increasing the maximum switch capacity to approach that offered by circuit switching [50, 149, 79, 146, 147]. Recent advances in optical fibre transmission technology provide very high bandwidth links with very low bit error rates. With a low error rate on each transmission link, error control is no longer required on a link-by-link basis at every switch in the path. Also, at high transmission rates it may prove impractical to attempt to provide the functions of flow control and error control on every link in the path due to delay and buffering requirements. Therefore, in fast packet switching, the functions of flow control and error control are implemented on an end-to-end basis, or on entry to and exit from the network [66]. Thus services that require error detection and correction may implement a retransmission strategy on an end-to-end basis whereas services, such as voice, that may tolerate a certain degree of error may take advantage of the low delay. As the protocol requirements of each switch are reduced, packets may be processed entirely in hardware. Thus switches of much greater capacity may be constructed and the switch may become more transparent to the data it carries than for conventional packet switching. Fast packet switching is in general connection-oriented. Thus once a virtual circuit is established across the network very short packet headers may be

used to distinguish between each of the virtual circuits multiplexed over a single link. Also the routing of each packet may be performed in hardware by table look-up. As the packet overhead has been significantly reduced, very short fixed length packets may be used to reduce the delay across the switch to levels comparable with that of circuit switching. Fast packet switching with short fixed length packets is often referred to as asynchronous time division (ATD) in the context of broadband ISDN.

Both fast circuit switching and fast packet switching offer statistical switching mechanisms that handle bursty traffic efficiently and are capable of supporting high capacity switch implementations. Fast packet switching requires a header on every packet whereas fast circuit switching requires a header only on every burst. Fast packet switching therefore carries a greater overhead, perhaps 10% of the available bandwidth or more in a typical application, but with high capacity optical fibre transmission links, bandwidth efficiency may not be the most critical parameter. In both forms of statistical switching, overload occurs when the incoming information exceeds the transmission capacity resulting in delay, loss of information or both. Fast packet switching, however, is able to spread the effects of delay and loss over all calls or over a selected class or classes of calls. With fast circuit switching the effect must be absorbed by at most a few selected calls and can thus result in more severe delay or loss effects. Fast packet switching is also able to vary the allocation of bandwidth to individual sources instantaneously and can thus allow much greater flexibility. Fast packet switching may also give a better performance for data traffic as end-to-end retransmissions are carried using the entire bandwidth of the transmission links rather than across the narrowband channels of fast circuit switching. Thus fast packet switching has been selected for further study partly because of its flexibility but also because a very simple design of fast packet switch was envisaged and considered to be worthy of detailed study (see chapter 5).

2.3 Evolution of the Packet Switch

Early Switch Architecture

In the early days of packet switching, computer processing power was an expensive commodity so packet switches were designed with a single central processor handling all of the switching, routing and protocol functions of the entire packet switch. Thus the throughput of the switch was limited by the processing capacity of the central processor and the complexity of the packet switching protocol. With the growth of VLSI technology the cost of processing fell rapidly until it became possible to provide some processing capacity on each switch port. Thus the lower level protocol functions, such as flow control and error detection and correction, could be handled independently by each switch port while the central processor provided higher level protocol functions such as routing. This increased the throughput by an order of magnitude, but as the central processor continued to interconnect all of the switch ports it remained a bottleneck.

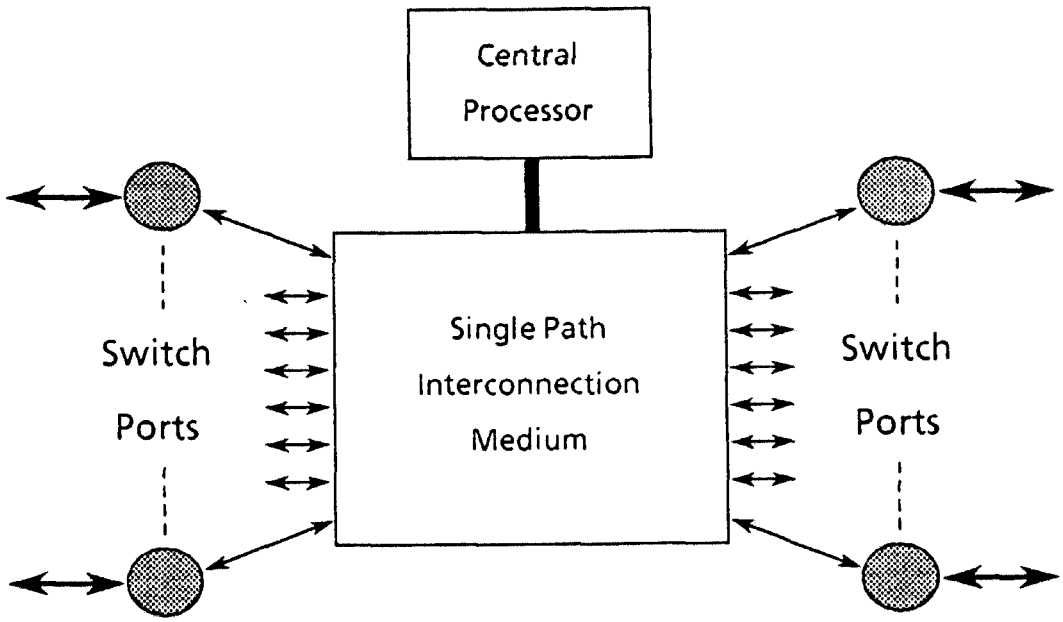

Figure 2.3: A single path decentralised packet switch.

The Single Path Decentralised Switch

To further improve the capacity of the packet switch it became necessary to remove the central processor completely from the transmission path of every packet. To achieve this, some form of single path interconnection medium was inserted to interconnect all of the intelligent, peripheral switch ports while the central processor took on more of a supervisory role, as illustrated in fig. 2.3. Hence, although some form of action may still have been required of the central processor on a per packet basis, it was decentralised by being removed from the task of physically transmitting each packet between the switch ports. The majority of packet switches have used shared memory as the single path interconnection medium with direct memory access in each of the switch ports but some designs have used serial bus [52, 26], parallel bus [16, 28] or ring [68, 67, 53] based structures.

When the majority of the per packet processing is removed from the central processor the throughput of the packet switch is determined by the bandwidth of the interconnection medium and the rate at which the processors in the switch ports can handle the protocol functions required. From an architectural perspective there is little difference between this class of packet switch and a local area network (LAN). The switch port of the packet switch corresponds to the media access controller of the LAN. The only major difference is that the switching function in the LAN is distributed across the local area. This requires a more complex media access protocol than for the packet switch for which access to the interconnection medium is contained within the confines of the switch. A parallel may also be drawn between this class of packet switch and digital circuit switches that handle up to about 1000 telephony

channels of 64 kbits/sec bandwidth. These also use a shared memory interconnection medium with a central processor that is in general only required at the set-up and clearing down of a connection.

Hybrid switch structures have also been proposed with a single path interconnection medium. These offer separate packet and circuit switching functions with integrated access and transmission facilities. Many such designs exist in the literature covering both discrete switches [80, 16, 151] and distributed switches, i.e. local area or metropolitan area network designs, both ring [18, 23, 138] and CATV bus [97].

The Multi-Path Switch

In considering switches of very high capacity, the bandwidth of a single path interconnection medium imposes a limit upon the switch capacity that may be achieved. To overcome this fundamental restriction it is clear that some form of multi-path interconnection medium is required that is capable of supporting communication between a large number of switch ports concurrently. Thus with a multi-path interconnection medium the total capacity of the switch is no longer limited to the bandwidth of the paths forming the interconnection medium but may grow as the number of switch ports increases. In this manner a much higher total switch capacity may be attained than for a single path interconnection medium using the same implementation technology. Conversely a high capacity switch no longer requires high speed and expensive device technology. The multi-path architecture applies equally to circuit, packet and hybrid switches. Circuit switches have used analogue multi-path switching networks for many years but more recently high capacity digital TDM circuit switches have been designed around a non-blocking, multi-path interconnection network [34, 160]. Hybrid multi-path switches have also been proposed [150, 139, 96] and the majority of current fast packet switch designs are multi-path switches, examples of which will be discussed in the following chapter. Many forms of multi-path interconnection medium are possible, e.g. multiple rings [139, 2], but the most general class, and the one which yields the highest switch capacities, is that of the multi-stage interconnection network which will be examined in detail in chapter 4.

2.4 Fundamentals of Fast Packet Switch Design

There are some basic concepts that are common to many designs of fast packet switch and these will now be introduced prior to the detailed discussion of existing fast packet switch designs presented in the following chapter.

A fast packet switch will in general consist of a set of input lines each arriving at an input port, a set of output lines each departing from an output port, with input and output ports interconnected via a switch fabric, fig. 2.4. A switch controller will also be interfaced to the switch fabric and may control the input and output ports either directly or via packets across the switch fabric. External connections to the switch are generally required in the form of bi-directional links which are formed by

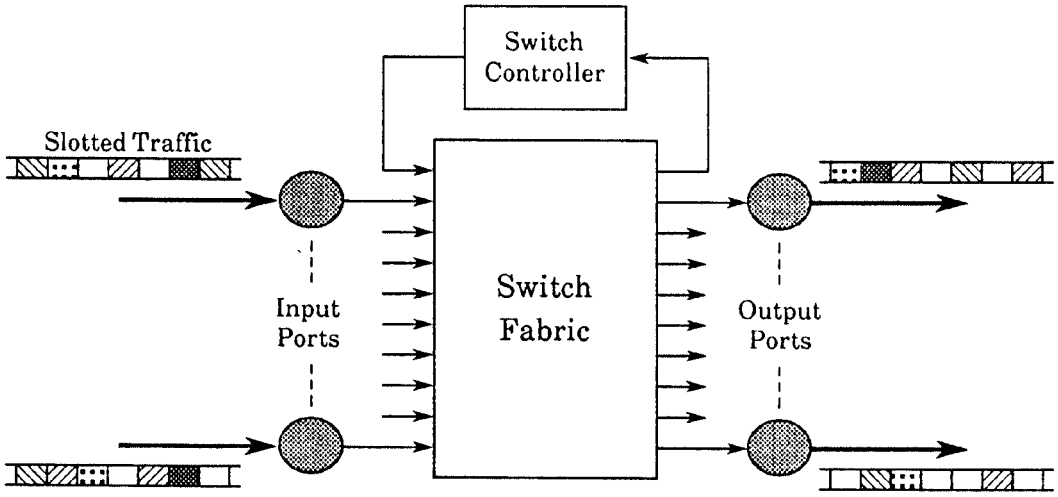

Figure 2.4: General structure of a fast packet switch.

grouping an input and an output line together. Many designs of fast packet switch are only capable of handling short fixed length packets. In such designs the bandwidth on both input and output lines is divided up into timeslots each of which may carry a cell (packet) or may be empty. All lines must be synchronised and this is accomplished either by means of a frame structure with a synchronisation pattern in every frame, as in TDM, or by filling empty cells with a synchronisation pattern. A multiplexing scheme of this nature is often referred to as 'slotted'.

Each packet includes a packet header which must contain a label that identifies the connection to which the packet belongs. In general the address space from which these labels are selected is specific to each input port of the fast packet switch. The label is selected by the control processor of the switch when the connection is established from the pool of unused addresses on the relevant input port of the switch. If the address space of the label field were not localised the problem of allocating a globally unique label in a large network would be time consuming and would limit the number of virtual circuits that could be supported within the network. Thus to support a connection across a number of fast packet switches a different label is required to traverse each link within the path. One function of the input port of a fast packet switch is therefore to replace the label field of the incoming packet with an outgoing label. It does this by means of a look-up table which is set up when the connection is established. In a very large hierarchically structured network a two level labelling technique may be required, one label for local switching and another for trunk switching. Conversely in smaller networks a simpler scheme may be adopted possibly based on a globally unique destination name or upon a unique area code with a local destination name [51].

If a high capacity fast packet switch is to be constructed, a multi-path design is required. This may be achieved either by interconnecting a number of complete fast packet switches to form a larger structure or by implementing the switch fabric as

a multi-stage interconnection network of simple switching devices. In both cases the switches that are interconnected will be referred to as switching elements. In the first case each switching element is a complete fast packet switch; complete with control processor, connection tables and label manipulation in the input ports. This allows flexibility in the choice of interconnection network but causes unnecessary replication of the control functions in a large switch. The second method, which is the more popular, does not require replication of the control processor or input port functions but implements the switch fabric as a multi-stage interconnection network of simple switching elements. Examples of multi-stage interconnection networks may be found in figs. 4.8 and 5.3.

The multi-stage interconnection networks generally selected have the property that a simple algorithm exists whereby each switching element can forward an incoming packet towards the correct output port. This algorithm usually requires that a tag specifying the required output port number be prefixed to each packet on entry to the switch. This function is performed in the input port by table look up on the label field in the packet header. One class of networks that display this property are commonly called banyan networks in the literature, although they have been more accurately defined as delta networks which refers to a specific sub-class of banyan networks. Switch fabrics are generally formed from square switching elements which have the same number of inputs as outputs and the degree of a square switching element is the number of its input (or output) ports. Most interconnection networks are constructed from identical switching elements. The degree of the switching element is important because it determines the number of stages of switching required in the interconnection network and hence the total number of interconnections required to form a given size switch. The number of interconnections required is a major factor in determining the maximum size of the switch due to implementation considerations.

2.5 Summary

Time division multiplexing (TDM) offers fixed bandwidth channels with a constant and low delay. Statistical multiplexing is much more flexible, offers variable bandwidth connections and handles bursty traffic much more efficiently but may suffer from high delay, high variance of delay and also loss of information under overload conditions. Conventional circuit switching supports the interconnection of TDM channels while conventional packet switching handles the interconnection of statistically multiplexed channels. A switching mechanism is required that combines the benefits of circuit switching: low delay, low variance of delay and high capacity switch structures; with the flexibility and efficiency for bursty traffic that is offered by statistical multiplexing. Two statistical switching mechanisms have been reviewed: fast circuit switching and fast packet switching. Both appear capable of offering a delay performance close to that of circuit switching while being much more efficient in handling bursty traffic. Fast packet switching has been selected for further study as it appears to be the more flexible switching mechanism and also for performance and implementation considerations. From a brief review of the evolution of the packet switch a

multi-path design has been suggested in order to achieve high capacity switch structures. Some of the basic concepts that underly many of the multi-path designs of fast packet switch have been introduced.

Chapter 3

Fast Packet Switch Architecture

Having established the basic concept of the fast packet switch and given an impression of the context into which it fits as a switching mechanism this chapter attempts to explore the architecture of the fast packet switch. A simple classification of fast packet switch designs is first introduced. A number of fast packet switch designs which have recently appeared in the literature are then reviewed and some comparisons drawn. Finally, from the existing literature, an elementary performance comparison between the three major classes of switch design is presented which will be developed in later chapters.

3.1 A Simple Classification of Switch Designs

Two fundamental components are required to construct a fast packet switch: switching and buffering; and the relative positioning of these components permits a simple classification of fast packet switch design, fig. 3.1. If the buffering remains external to the switch fabric the design is based upon a non-buffered switch fabric. Else, if the buffering is implemented within each of the switching elements forming the switch fabric a buffered switch fabric (or internally buffered) design results. Of the designs based upon a non-buffered switch fabric, if the buffering precedes the switch fabric the switch is classified as input buffered. Else, if the buffering follows the switch fabric the design is output buffered. An input buffered design requires much less hardware and fewer interconnections than a similar output buffered switch but its basic performance is only about half that of the ideal output buffered switch. This difference in performance results from an effect known as head of the line blocking [71, 76] which is discussed in the following section.

Input buffered switches may be classified according to whether the switch fabric is blocking or non-blocking. Blocking is said to occur when the transmission of an incident packet to a free output is temporarily prevented by other traffic within the switch fabric. A blocking switch will have a lower performance than an equivalent non-blocking fabric but will require fewer switching elements and interconnections. Various techniques are available to improve the basic performance of an input buffered

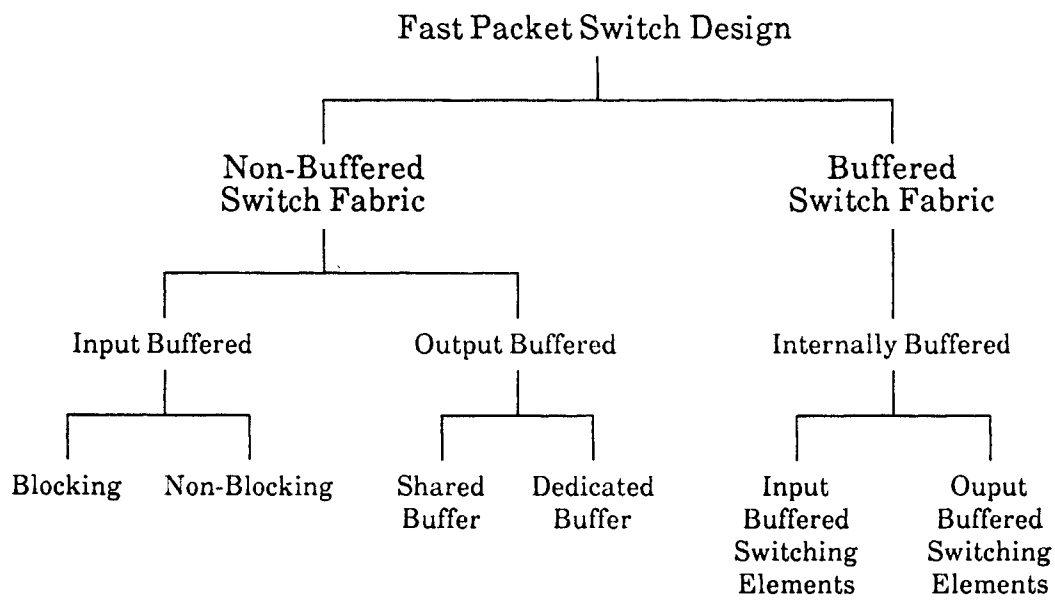

Figure 3.1: A simple classification of fast packet switch design.

switch towards that of the output buffered switch. The Cambridge Fast Packet Switch is an example of a blocking input buffered design while the three phase Batcher-banyan [71] is non-blocking.

Output buffered switches may share a pool of buffers between all output ports of the switch fabric else buffers may be dedicated to each output port. A shared design will require fewer buffers than a dedicated design. Starlite [70] provides an example of a shared buffer design while the Knockout switch [163] employs dedicated buffers.

In an internally buffered switch design each switching element within the switch fabric has its own buffers. These buffers may either be implemented on the input side of each switching element or upon the output side. Input buffered switching elements are much simpler to construct than are output buffered switching elements but they offer a lower performance. In general the performance of an internally buffered switch design using input buffered switching elements is similar to that of an input buffered switch design using a non-blocking switch fabric. Conversely, the performance of an internally buffered design with output buffered switching elements may approach that of an output buffered switch design for switching elements of large degree. Turner's switch [145, 148, 19] and the CSELT design [128, 56] offer examples of an internally buffered design using input buffered switching elements. Prelude [141, 31], the Bus Matrix Switch [120], the IBM Switch [4] and the TDM Bus design [36, 35] are all examples of designs based upon an output buffered switching element.

3.2 Input Buffered Switches

The banyan network is a multi-stage interconnection network employed as a switch fabric in both buffered and non-buffered designs. It is a blocking network but if all active incident packets are sorted into order based upon their destination output port number before being applied to the banyan network, non-blocking performance results [107, 88]. In a non-buffered switch fabric the Batcher sorting network may be used to provide this function and a non-blocking Batcher-banyan network results. This network requires that all packets be of the same length and that they be presented to the network in synchronism. It also requires any conflicts between packets contending for the same output port during the same timeslot to be resolved prior to the network. The Batcher-banyan switch fabric has been suggested for use both in input buffered and output buffered switch designs which will now be reviewed. A detailed discussion of the construction of Batcher, banyan and other interconnection networks will be presented in the following chapter.

The Three Phase Batcher-Banyan

The Batcher-banyan switch fabric will only offer non-blocking performance provided that no more than a single packet requests access to any switch output at the same time. The three phase Batcher-banyan switch [71] detects packets contending for the same output within the same timeslot by use of a three phase algorithm and causes requests that cannot be satisfied to be buffered at the input of the switch fabric. All packets are submitted to the switch fabric synchronously, and requests that cannot be satisfied in the current timeslot are re-submitted in the next. The basic structure of the switch is presented in fig. 3.2. Incoming packets are queued in the input buffers which act as first in first out (FIFO) queues. In phase I of the algorithm every port controller with a packet to transmit sends out a pilot packet which consists merely of the required switch output port number followed by the source input port number. These pilot packets are sorted into ascending order of output port number by the sorting network. This results in requests contending for the same output port emerging on adjacent ports of the sorting network. Every k^{th} output of the sorting network is fed back to every k^{th} port controller which allows the port controllers to decide which packets win the contention by comparing adjacent packets. At this stage, the input ports which originated the requests do not know the result of the arbitration, thus phase II of the algorithm, the acknowledgement phase, is necessary. In phase II every port controller that observed a successful pilot packet returns an acknowledgement to the originating port controller across the entire Batcher-banyan switch fabric. In phase III those input ports that receive an acknowledgement to their pilot request transmit the full packet across the switch fabric.

This fast packet switch design has two significant drawbacks, both of them a result of the size of the Batcher sorting network. A much larger number of switching elements are required to form a Batcher network than for a banyan network and although the switching elements of the Batcher network may be simpler to implement

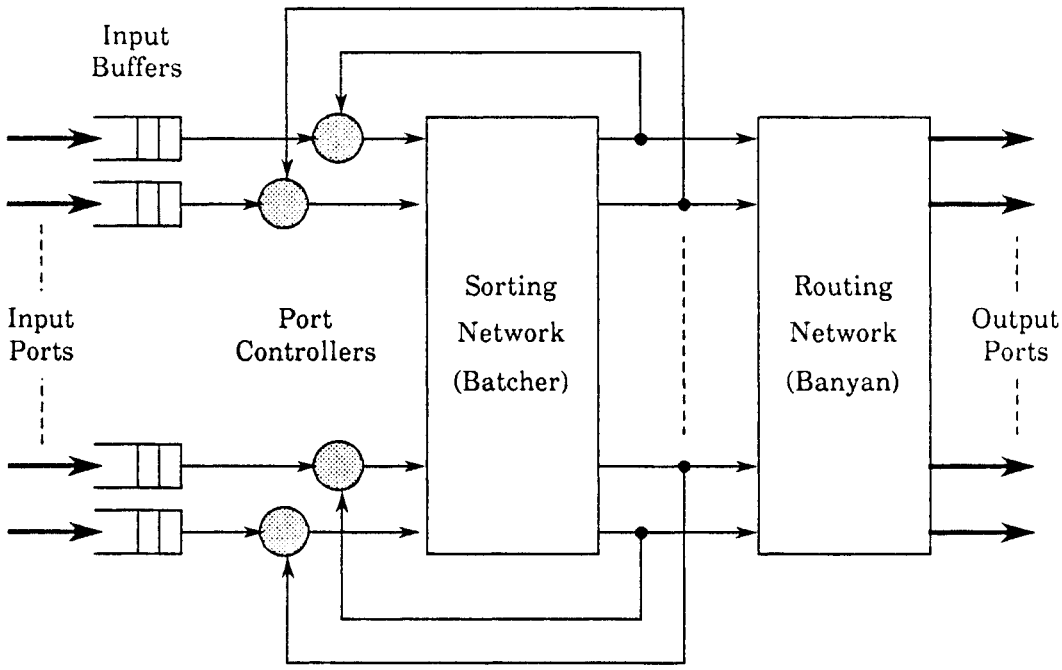

Figure 3.2: Basic structure of the three phase Batcher-banyan switch.

than those of the banyan, difficulties arise when partitioning the switch fabric for implementation in VLSI. One such implementation is discussed in [34] in which two VLSI chips have been designed for the switch fabric, one for the front end and one for the back end. In the 64×64 example switch discussed, the Batcher-banyan switch fabric has been partitioned into 9 stages, 7 of which are required for the Batcher sorting network and only two for the banyan routing network. Thus six stages of interconnection result from the Batcher network while only one is required for the banyan. Whilst an impressive example of layout and engineering permits the goal of a 256×256 Batcher-banyan switch fabric operating at 100 MHz with CMOS devices to look realistic, one is tempted to consider the performance of switch fabrics that suffer a small amount of blocking but are much smaller and simpler to partition and construct.

The second problem related to the size of the Batcher sorting network is that of the overhead incurred by the first two phases of the three phase algorithm. Phases I and II of the algorithm require a total of $\log_2 N(\log_2 N + 4)$ bit times to arbitrate between conflicting requests (where N is the size of the switch). The majority of this is due to delay through the Batcher sorting network. Thus for a large switch size, e.g. 1024×1024, a total of 140 bit times of overhead would be required for every packet. A popular packet size for use in the broadband ISDN is currently considered to be 128 bits, so such a switch would be operating at a basic efficiency of less than 50% at this packet size. A suggestion has been made to reduce the three phase algorithm to a two phase algorithm by the incorporation of two additional forms of

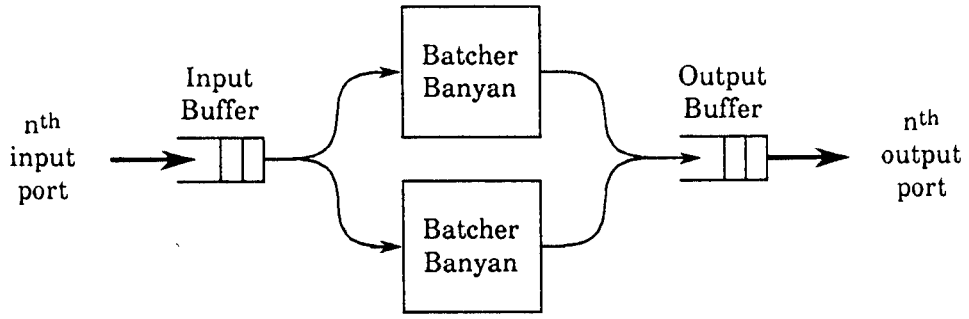

Figure 3.3: A two-plane switch structure.

interconnection network between the Batcher and banyan networks. This however considerably increases the size of an already large switch fabric.

Another problem, common to all designs of input buffered switch, is referred to as head of the line blocking. When a packet contends for an output port and loses, during a particular timeslot, it remains at the head of the FIFO queue at its input port until the following timeslot. It may be that there are other packets behind it on the queue destined for other output ports which could have been served during the current timeslot. These packets are blocked and this causes a reduction in throughput compared to output buffered and internally buffered switch designs. In input buffered switch designs that are asynchronous at the packet level, the transmission of packets other than the one at the head of the queue may be attempted on discovering that the first packet is blocked. This technique is often referred to as input queue by-pass. It improves the throughput of the switch and also reduces the sensitivity of the switch to the arrival statistics of the incident traffic. In a synchronous design, such as the above, this may only be achieved at the expense of repeating phases I and II of the algorithm on the second packet in the queue, after the arbitration of the first packet is completed, prior to phase III in every timeslot. This would significantly increase the overhead for every packet.

An alternative is to adopt a two-plane switch fabric as shown in fig. 3.3. The three phase algorithm of the second switch fabric is phased to commence just after the algorithm for the first switch fabric is completed so that packets blocked in the first fabric may be sent across the second. This requires each output port to be capable of handling two packets arriving at once. As there is no feedback from the output ports of the switch to the input ports, the output buffer must be dimensioned to reduce the probability of buffer overflow to acceptable proportions. Also the input port may be constructed to be capable of transmitting across both switch planes simultaneously, or alternatively only across a single plane at any one time. The most important aspect of adding the second switch plane is to provide redundancy for enhancing system reliability, but from the performance viewpoint it is interesting to question to what extent non-blocking operation is now required of the individual switch planes. Could the use of much simpler, blocking switch planes, in a two-plane structure achieve a similar performance?

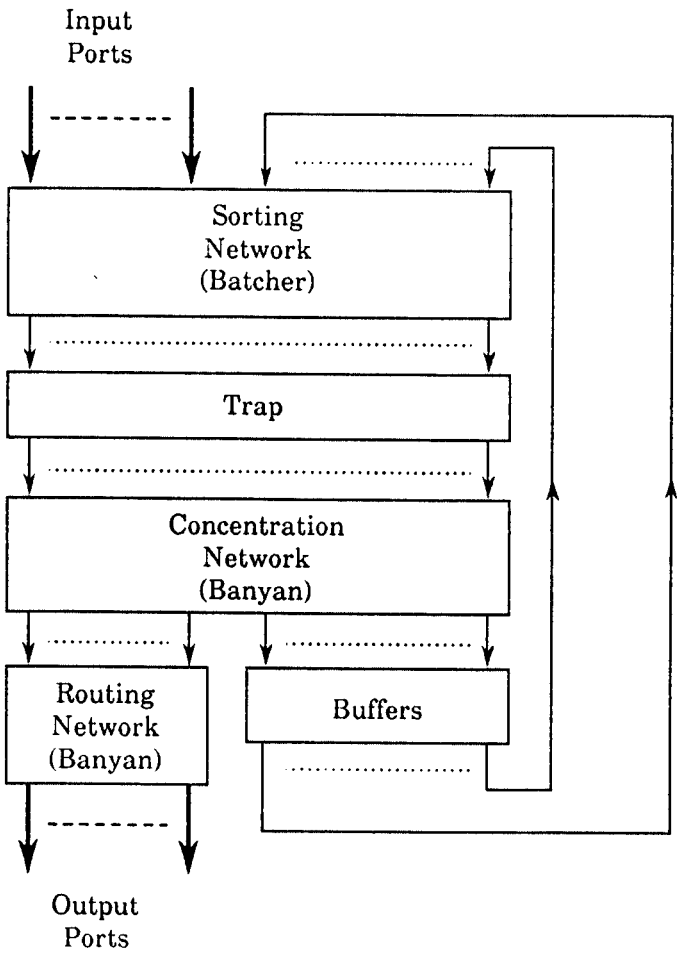

Figure 3.4: The Starlite fast packet switch.

3.3 Output Buffered Switches

Starlite

This is another fast packet switch design based upon the use of a Batcher-banyan switch fabric but overcomes the head of the line blocking problem by storing packets which fail arbitration in shared buffers at the output of the sorting network rather than at the input [70]. The basic structure of the Starlite switch is shown in fig. 3.4. Again the switch is synchronous at the packet level. On entry to the switch a tag is prefixed to each packet which contains the port number of the required output port. Incoming packets are sorted according to the tag by the Batcher network and the trap arbitrates between packets contending for the same output port which appear on adjacent ports on output from the Batcher network. No more than one packet may be directed to any single output port during the same timeslot thus conflicting packets which lose arbitration must be directed to the buffers in order to

be recirculated through the network during the next timeslot. To accomplish this the trap computes a running sum address for those packets destined for the output ports and also for those to be directed to the recirculation buffers and appends this address as a further tag onto the front of each packet. When the packets then pass through the concentration network, packets destined to the output ports emerge on the left hand outputs, packets directed to the recirculation buffers on the right hand outputs, and empty packets in the centre. Packets directed to the output ports are routed to the requested ports by the banyan routing network while packets which lost arbitration are stored in a pool of shared buffers and re-enter the sorting network together with incoming packets at the next timeslot.

The same comment as regards difficulties with the size and partitioning of the Batcher sorting network applies to the Starlite design as to the three phase Batcher-banyan but here we also have the running adders of the trap network and an additional banyan concentration network to include. Thus more stages are required in the switch fabric and the number of interconnections is increased. In addition to this, analysis shows that the number of switch fabric ports devoted to re-entry must be two to three times the number of the switch input/output ports to ensure an acceptably low probability of packet loss, [65, 34]. The additional hardware required in the sorting network to handle the recirculated packets is reduced by the fact that the recirculated packets have already been sorted into order. However, a reduction in the buffering requirements is gained at the expense of a much larger switch fabric when compared to other switch designs. One further difficulty with the Starlite switch is that packets may be delivered out of sequence due to the recirculation mechanism. This may be overcome with a simple time stamping technique which gives older packets a higher priority at arbitration.

An extension to the Starlite switch to offer multicast capability is discussed in [70] which requires another two interconnection networks to be added to the switch fabric. Empty packets are injected into the switch fabric directed to every output in the multicast groups and are filled with a copy of the required packet data field in a copy fabric. Another modification of the basic Starlite design is discussed in [34] which uses multiple adjacent re-entry loops. An example of the layout of a 128×128 switch reveals a very large number of switch stages and interconnections.

Finally, it is a simple matter to extend any switch based upon the Batcher-banyan switch fabric to handle any number of levels of packet priority. If the priority field is appended to the least significant digit of the destination field in the tag at the front of the packet, then all packets contending for the same output port will emerge from the Batcher network sorted into order of priority. The arbitration logic may thus easily select the highest priority packet. This is likely to become a desirable feature of a fast packet switch in any network handling multi-service traffic but particularly in a public network.

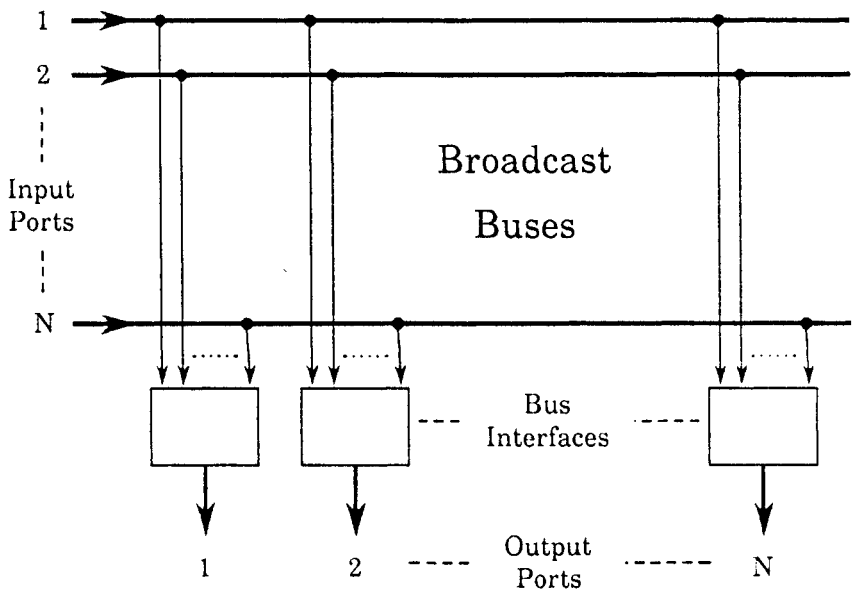

Figure 3.5: Structure of the Knockout Switch.

The Knockout Switch

The Knockout Switch is another example of an output buffered switch but differs from Starlite in that buffers are dedicated to specific output ports and not shared [163]. There is therefore no recirculation of packets. Packets which cannot be accommodated by the available buffers are discarded and the switch is dimensioned to keep the probability of packet loss sufficiently low.

The structure of the Knockout Switch is outlined in fig. 3.5. The switch is based on a fully connected switch fabric in which the traffic on each input port is broadcast to every output port. The switch operates synchronously at the packet level and at the start of each timeslot, incoming packets are broadcast across the buses of the switch fabric preceded by their destination tag. The bus interfaces of every output port inspect the destination tags of every packet on all of the broadcast buses to select packets that are addressed to them. These packets are then buffered in the switch output ports provided that no more than a limited number, typically eight, arrive at any output port during the same timeslot. The construction of the bus interface is shown in fig. 3.6. The packet filters select those packets addressed to the output port to which they belong. These are then submitted to an N input L output concentration network. Provided that no more than L packets arrive in any given timeslot they are passed on to the shared buffer otherwise excess packets are discarded. The shared buffer allows up to L packets to arrive simultaneously but maintains a single first in first out queue by means of the shifter which allocates packets to buffers in a cyclic manner. The buffers are emptied in a cyclic manner.

The most obvious difficulty with the Knockout Switch arises from the use of a fully

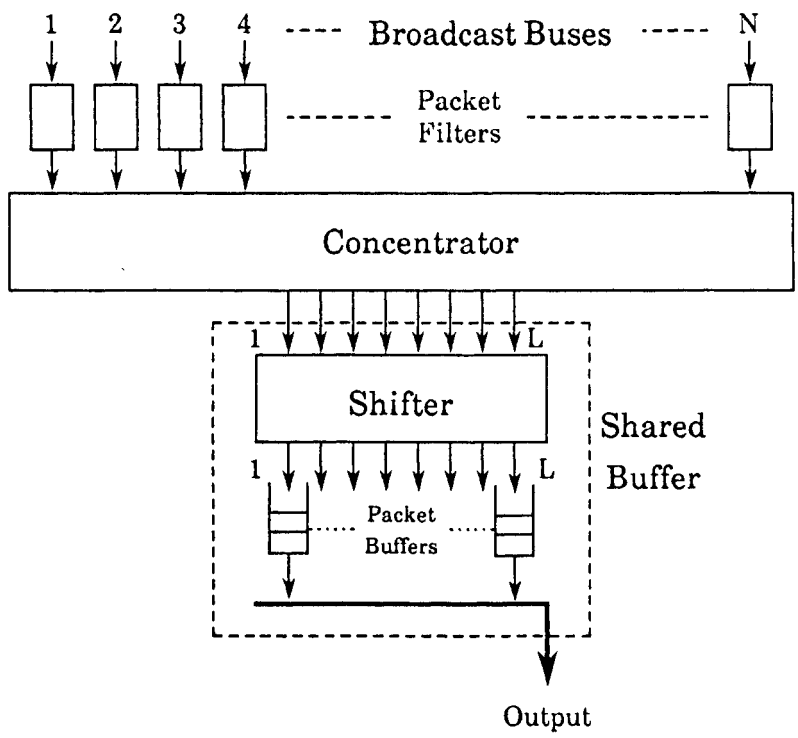

Figure 3.6: The bus interface of the Knockout Switch.

interconnected broadcast switch fabric. This results in an increase in the hardware and interconnections required of at least one order of magnitude when compared to input buffered switches of the same size. It does, however, offer the advantage of incremental growth allowing the switch to grow by one port at a time. The buffer requirement is approximately twice that of an input buffered switch for the same packet loss probability. Eight parallel buffers, each of five packets deep, in each switch output port will yield a packet loss probability of 10^{-6} at a load of 84% for random traffic. This is considered by the designers to be acceptable when compared to packet loss from other sources.

An extension of the Knockout Switch to handle variable length packets is introduced in [44] together with further thoughts on the possible implementation of the switch. Implementation of the Knockout switch using photonic components in the data paths of the switch is discussed in [43]. The broadcast nature of the switch fabric is likely to limit the maximum operating speed of an electrical implementation of the switch and will limit the maximum size of a photonic implementation. The $O(N^2)$ interconnection requirement will make switches larger than a few hundred ports difficult to construct. The broadcast nature of the switch fabric might suggest that the Knockout Switch could easily be adapted for multicast operation but this would require excessive complexity in the packet filters. Some suggestions have been made in [45] regarding the support of multicast capability but only at a fairly low

multicast capacity. The performance of an output buffered switch for periodic input traffic is considered in [75].

3.4 Internally Buffered Switches

An internally buffered fast packet switch is constructed from switching elements that contain one or more packet buffers per port. As with the construction of non-buffered switches we find that buffered switching elements may be divided into two classes: output buffered switching elements and input buffered switching elements. As the name suggests, an output buffered switching element is constructed by placing the buffering after the switching function on the outputs of the switching element. This gives the output buffered switching element a higher throughput than for an input buffered switching element due to head of the line blocking within the input queues of the input buffered switching element. It is, however, a more complex device to implement and requires a large number of packet buffers on each output port to reduce packet loss to reasonable proportions. Some proposed designs of output buffered switching element tend towards being complete fast packet switches in their own right, with a capacity of several Gbits/sec, of degree 8 or 16. In general, three-stage networks are suggested for the switch fabric structure but little work is available on the performance of such buffered switching elements in large switch structures.

In contrast, the input buffered switching elements suggested in the literature are small (2×2), with limited buffering of one or two packet buffers on each of the input ports of the switching element. They rely on backpressure between the stages of the switch fabric to prevent buffer overflow. They are based upon the banyan switch fabric, possibly with additional stages to distribute incident traffic evenly across the switch fabric, for which the performance is a little better characterised than for the class of output buffered switching elements.

Output Buffered Switching Elements

Prelude

Prelude, together with Starlite, is amongst the earlier designs of fast packet switch [141, 31]. As such it more closely resembles the structure of the conventional shared memory digital circuit switch than other fast packet switch designs. It forms an output buffered 16×16 switching element in which all of the functions of a complete fast packet switch are implemented. The switching element operates at a bit rate of 280 Mbits/sec on each switch port, with packets of length 128 bits, giving a capacity of about 3.6 Gbits/sec for a switching element operating at a load of 80%.

A serial to parallel converter transforms the incoming serial bit stream on each of the input ports into a stream of 8 bit octets in parallel, fig. 3.7. With 16 input ports and a total packet length of 16 octets, including a one octet header, a delay is introduced so as to align each of the incoming streams such that only a single packet

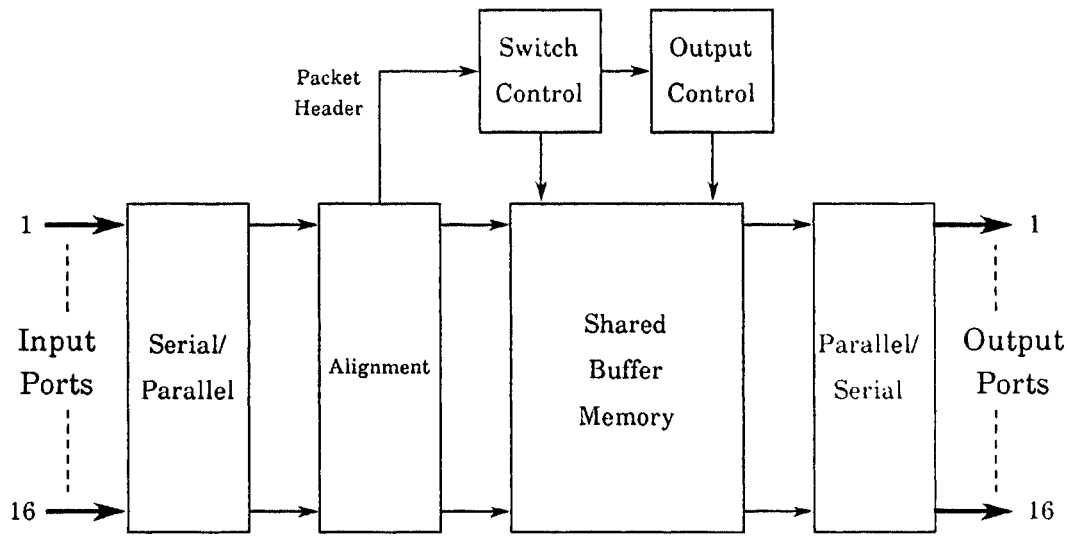

Figure 3.7: Structure of the Prelude switching element.

header emerges from the alignment unit during each octet clock cycle. A translation operation by table look-up is performed on the 8 bit packet header and the packet allocated a position in the shared memory in which it is to be stored. At the same time an entry is made in the relevant FIFO output queue of the output port to which it is routed. These output queues, one for each output port, consist of pointers to the location of the relevant packets in the shared memory. On the output side of the switch, packets are extracted from the shared memory and transmitted over the output ports according to the entries in the output queues.

The use of a shared memory, with a single shared processing unit for the packet headers, may limit the flexibility of the Prelude design. It does, however, allow a large amount of buffer space to be shared dynamically between all output ports. The switch could be extended to offer two levels of packet priority without great difficulty. Also the switch may offer multicast connections by writing pointers to the same packet into multiple output queues, although this introduces the possible problem of knowing when a packet may be deleted from the shared buffer space.

The Bus Matrix Switching Element

In common with the Prelude design, the Bus Matrix Switch implements all of the functions of a complete fast packet switch in every switching element [120]. The structure of the switching element is given in fig. 3.8. The primary packet distributor (PPD) inspects the incoming packets and transmits them across a bus to the appropriate cross point memory (XPM). The cross point memory is a FIFO queue which connects the incoming buses from the PPDs to the outgoing buses and these are arranged in a matrix structure. The secondary packet distributors (SPDs) read out packets from the cross point memories to the output ports.

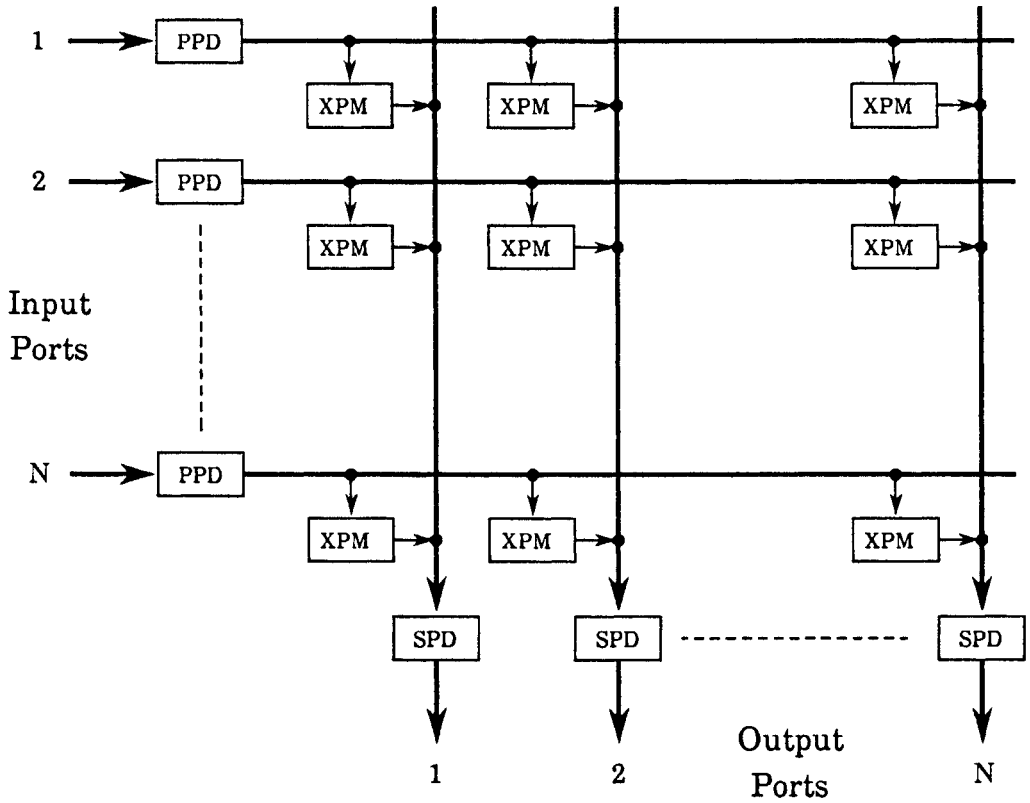

Figure 3.8: Structure of the Bus Matrix switching element.

The switch operates asynchronously at the packet level and may handle variable length packets. It is claimed that the broadcast function is easily implemented though multicast operation would probably be much more difficult. A size of 16 Kbytes is proposed for each cross point memory which suggests that using VLSI a 16×16 switching element could be implemented on a single circuit board. For CMOS implementation a maximum switching element size of 16×16 may be achieved with each input port operating at 160 Mbits/sec using 8 bit wide internal paths for the bus matrix. This would offer a total capacity of 2 Gbits/sec at a loading of 80%. For an ECL implementation the maximum switching element size is 8×8 with a line bit rate of 800 Mbits/sec offering 5 Gbits/sec switch capacity at an 80% load. A three stage switch fabric structure is suggested to achieve larger sizes of switch but little detailed information is given.

This design of buffered switching element is flexible in that various sizes and capacities of switching element may be constructed from standard parts. It does, however, require much more buffering than other designs as buffer space is partitioned not only between the outputs but also according to each input, thus there is no sharing of buffer space whatsoever.

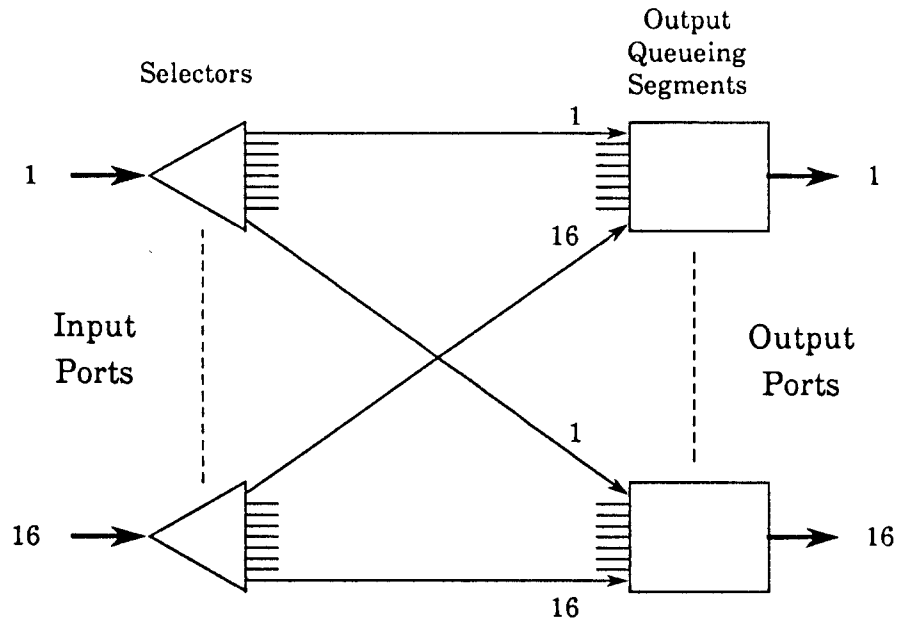

Figure 3.9: Structure of the IBM switching element.

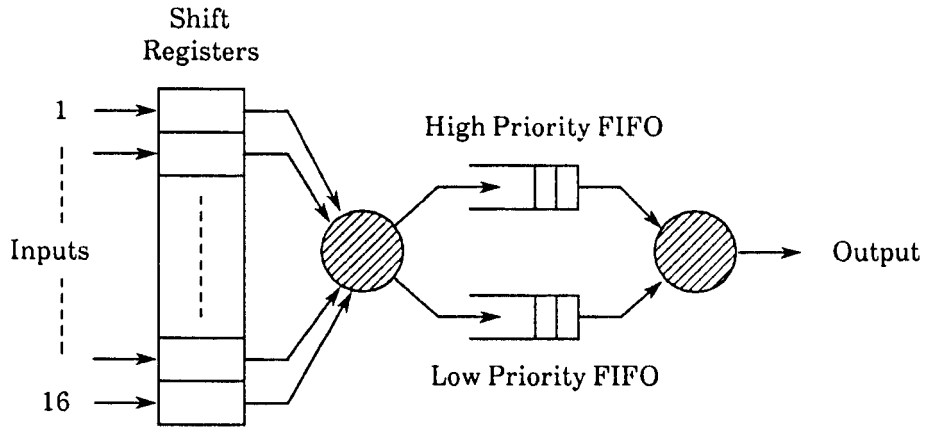

Figure 3.10: The output queueing segment of the IBM switching element.

The IBM Switching Element

This is a 16×16 output buffered switching element designed to offer a bit rate of 32 Mbits/sec on each switch port with two levels of packet priority to accommodate traffic with real-time constraints [4]. The structure of the switching element is illustrated in fig. 3.9. The selector examines the tag at the head of an incoming packet and forwards it to the corresponding output queueing segment. The output queueing segment, shown in fig. 3.10, is capable of accepting 16 packets arriving at once and queueing them in either of the two output FIFO queues according to the

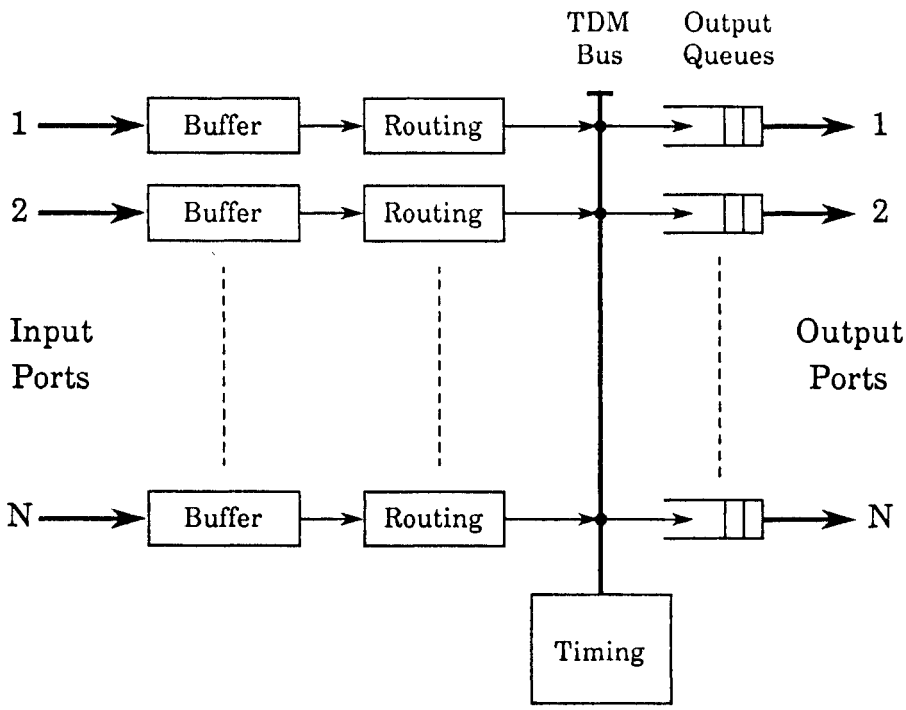

Figure 3.11: Structure of the TDM Bus switching element.

packet priority. It achieves this by operating at 16 times the line rate of the switch ports and uses a set of 16 shift registers to add a small amount of temporary storage. The design of the output queueing segment to operate at 16 times the line rate of the switch ports limits the bandwidth of the switch ports that may be attained. Also, large output FIFO queue sizes, in excess of a hundred packets per port, are proposed to maintain a low packet loss rate which might prove expensive to implement in VLSI. A backpressure mechanism is discussed to reduce the length of the output queues. Suggestions are given for the use of the switching element in both single stage and three stage switch fabrics.

The TDM Bus Switching Element

The last example of a high capacity switching element, [36, 35], bears some resemblance to the IBM switching element. An outline of the structure of the switching element is provided in fig. 3.11. Incoming packets are buffered in the single packet buffer at each of the input ports of the switching element. The TDM bus operates at the sum of the line rates of the input ports and each input port has a corresponding timeslot. The routing logic directs the packet to the required output queue, according to the tag at the head of the packet, during its timeslot on the bus. The output queues must also work at the same rate as the TDM bus in order to provide non-blocking operation.

The size of the switching element will be limited by the speed of the TDM bus and output queues, 8×8 at 560 Mbits/sec being suggested as possible within a few years. The output queues will also have to be large to minimise the packet loss due to buffer overflow, a length of 40 packets being proposed. The packet length is fixed, switch operation is synchronous and multicast operation within the switching element may be difficult to implement. As with the IBM switching element, packet priority could be introduced at the cost of increasing the number of output queues.

Input Buffered Switching Elements

Several fast packet switch designs have been proposed based upon the use of a 2×2 input buffered switching element of which two offer some details of the implementation of the switching element, Turner [145, 148, 19] and CSELT [128, 56]. In both cases the switching element is implemented in CMOS running at about 25 MHz with 8 bit wide data paths. Turner suggests the use of large buffers of about 10 Kbits each while CSELT, who have implemented their switching element in 3 μm CMOS, have used 512 bits per input buffer. Both designs employ backpressure between adjacent switch stages to avoid buffer overflow. A banyan interconnection network is suggested for the switch fabric with additional stages added to distribute the traffic evenly across the switch fabric. The CSELT switching element has been implemented using 6,000 gates which demonstrates the hardware simplicity of an input buffered switching element when compared to output buffered switching elements. Turner goes to some detail in considering the support of multicast connections across his switch design.

Although an input buffered switching element provides a much simpler design it also offers a lower performance than that of the output buffered switching element. Also the 2×2 elements require many more switching stages within the switch fabric and thus more interconnections than a switch fabric constructed from switching elements of higher degree. This tends to reduce the maximum size and capacity of switch which may be constructed.

3.5 Performance Comparison

Some general analytical and simulation results are available which allow a first order comparison of the performance of the basic switch structures. An analytical investigation of input buffered interconnection networks was presented in [126] which looks at the throughput at saturation of the non-blocking (or crossbar) network and also at the banyan (or delta) network. The result for the banyan network is expressed as a recurrence relation for which an asymptotic analysis is presented in [82] and upper and lower bounds on the solution in [85]. These analytical solutions, however, simplify the problem by assuming that blocked packets are discarded and that the switch is operated synchronously at the packet level. The first assumption causes the throughput to be overestimated while for the banyan network the second assumption leads to an underestimation because blocking within the network is maximised by

the assumption of synchronisation. Analysis of the throughput at saturation of input buffered, non-blocking switch fabrics in which blocked packets are not discarded but are queued and resubmitted is offered in [76] and [71]. A comparison of the delay performance of both input and output buffered switch fabrics is offered in [76]. The outcome of the above analyses is to demonstrate that for the non-buffered switch fabric an input buffered, non-blocking switch has a throughput at saturation of approximately 58% that of an output buffered switch. This is due to the effect of head of the line blocking within the input queue of a pure input buffered switch.

Some results are also available for an internally buffered switch fabric constructed from 2×2 switching elements. An analytical approach is taken in [40, 41, 74] while [100, 19] offer simulation results. Differences in the models of the switching elements mean that the results are not in exact agreement but in general they show that buffered switch fabrics of 2×2 switching elements with a single buffer on each input port offer a throughput at saturation performance similar to input buffered switch fabrics of non-buffered 2×2 switching elements. Further, internally buffered switch fabrics constructed from 2×2 switching elements with four buffers at each input port offer a throughput at saturation performance comparable to non-blocking, input buffered switch fabrics.

3.6 Summary

A fast packet switch requires two fundamental components: switching and buffering. Switching takes place in the switch fabric which is generally a multi-stage interconnection network constructed from the interconnection of a large number of fundamental switching elements in stages. If the buffering is external to the switch fabric a non-buffered switch fabric results, formed from non-buffered switching elements. In such a switch design, if the buffering precedes the switch fabric an input buffered switch results. Else, if the buffering follows the switch fabric an output buffered switch results. An internally buffered switch fabric is composed of buffered switching elements. The buffering within a switching element may either be located at its input ports or at its output ports.

An input buffered switch with a non-blocking switch fabric offers approximately 58% of the throughput at saturation performance of an output buffered switch. A blocking switch fabric will offer even lower performance but techniques such as input queue by-pass and multiple switch planes with output buffering across the switch planes exist and will increase the performance. An output buffered switch requires much more hardware than an input buffered switch of comparable size.

The performance of an internally buffered switch fabric composed of switching elements of high degree with buffering on the output ports will offer a performance which approaches that of an output buffered switch fabric. The switching element will, however, be complex, require much hardware for implementation and will require large buffer sizes. An internally buffered switch fabric composed of switching elements of low degree with small buffer sizes on the input ports and backpressure between

switch stages will exhibit a performance comparable to that of the input buffered switch fabric.

Chapter 4

Multi-Stage Interconnection Networks

The general concept of the multi-stage interconnection network, together with its routing properties, have been used in the preceding chapter to describe the operation of various designs of fast packet switch. In this chapter those networks that bear particular relevance to applications within the field of fast packet switching will be described in some detail within the context of interconnection networks in general.

4.1 An Introduction to Interconnection Networks

A number of useful general surveys of interconnection networks have been published, notably [99, 47, 95] with [132, 143] also being of some relevance. Much of the early work on interconnection networks was motivated by the needs of the communications industry, particularly in the context of telephone switching. With the growth of the computer industry, applications for interconnection networks within computing machines began to become apparent. Amongst the first of these was the sorting of sequences of numbers, but as interest in parallel processing grew, a large number of networks were proposed for processor to memory and processor to processor interconnection [131]. With the advent of the fast packet switch, interest in interconnection networks has turned full circle in that many of the networks originally proposed for parallel processing are now being considered for use in fast packet switch designs.

A simple classification of interconnection networks according to topology will first be offered followed by some comments on the control mechanisms employed with the various classes of multi-stage network. A discussion of the blocking characteristics of the various networks will then lead into a detailed discussion of some of the multi-stage interconnection networks useful in communications applications.

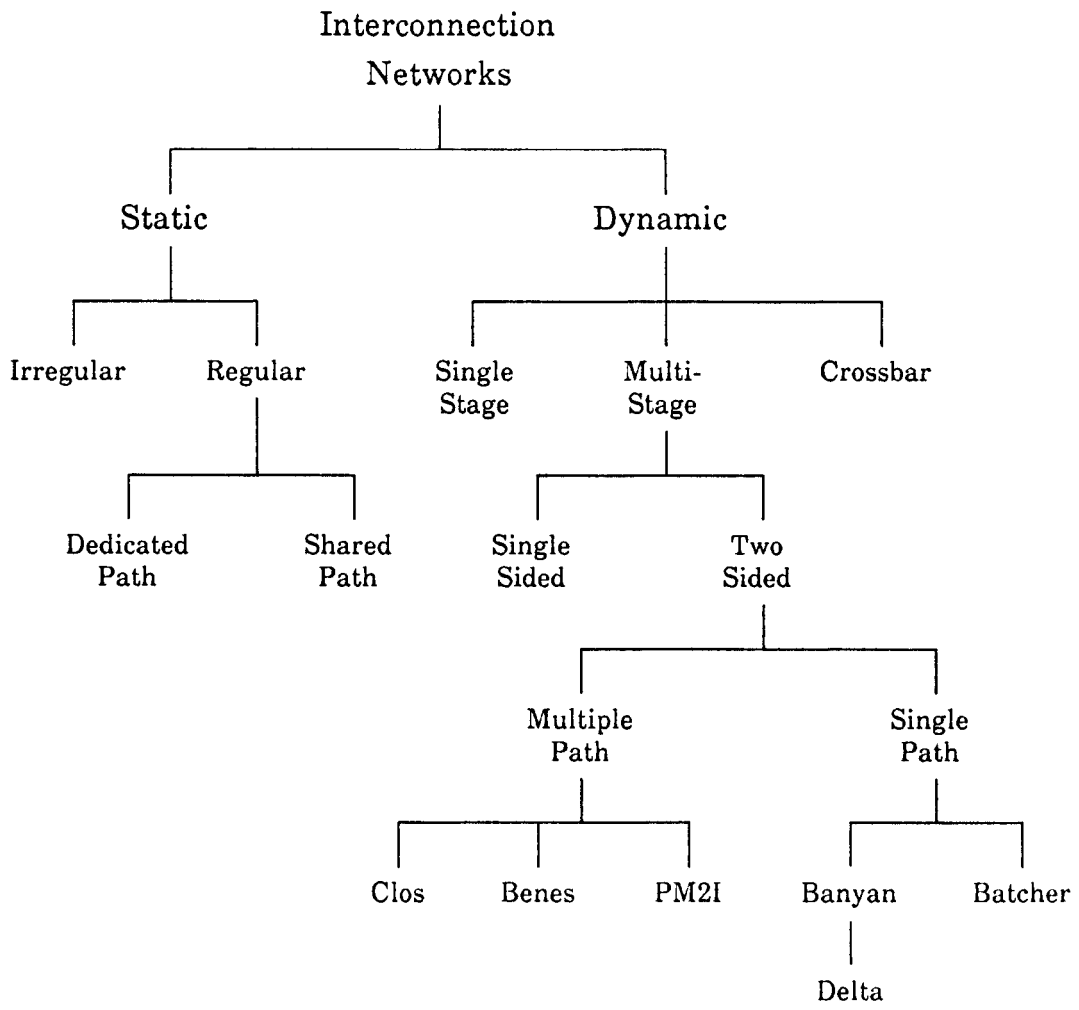

Figure 4.1: A simple classification of interconnection networks.

Topology

A simple, general classification of interconnection networks is presented in fig. 4.1. Regular, static networks, also called dedicated networks, are mostly used to interconnect loosely coupled processors to form parallel processing machines while any general packet switching network may be classed as an irregular static network. Two simple examples of dedicated path static networks are given in fig. 4.2 in which processing elements are connected by point-to-point links. Shared path static networks are formed by interconnecting processing elements with buses. In general the use of regular static networks has been restricted to the packet switched interconnection of loosely coupled processors as the delay across the network is dependent upon the distance between the communicating nodes. Also the processing delay required by the routing algorithm may render the use of short packets inefficient. Further, regular static structures often prove difficult to expand to large networks whilst maintain-

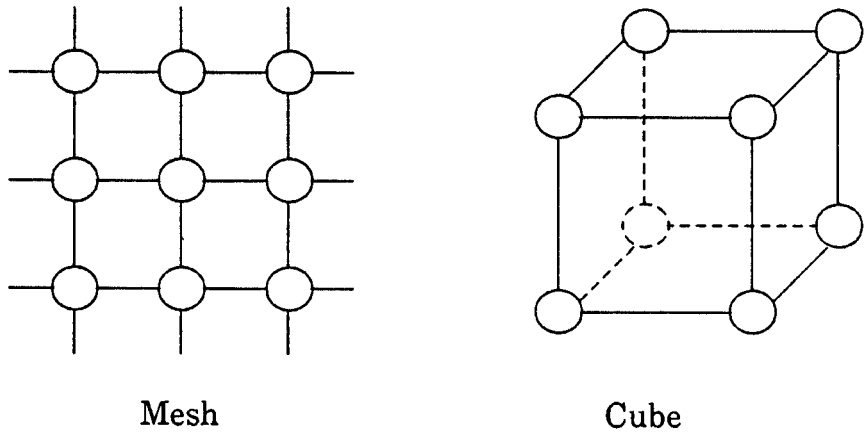

Figure 4.2: Examples of regular static network topologies.

ing the regularity of the structure. One notable exception is the use of a regular mesh topology in the Manhattan Street Network [98] which has been proposed as a metropolitan area network.

Dynamic interconnection networks are so called because the network clients are interconnected through an array of simple switching elements. Thus the pattern of interconnections between clients may be rapidly changed either by a centralised processor or by a distributed algorithm.

In [137] Stone introduced the perfect shuffle as a pattern of interconnection links of some interest in the solution of a number of classes of computational problem via a tightly coupled parallel processor. (The term 'perfect shuffle' derives from analogy with the process of shuffling a deck of cards in which the deck is divided into halves and re-assembled by alternately taking one card from each of the halves.) A single stage implementation is illustrated in fig. 4.3 comprising the perfect shuffle pattern of interconnection links followed by a single stage of switching elements. To complete the network every output is buffered and fed back to its corresponding input. Packets of data therefore circulate through the structure until they exit at the desired output [24].

If multiple copies of the single stage shuffle exchange are cascaded a multi-stage interconnection network results sometimes called a multi-stage shuffle exchange. Data is no longer required to circulate through the network but passes through the structure from the input side to the output side. Networks which have separate input and output sides are called two-sided, they are of great interest to communications applications and a number of examples will shortly be discussed. Single sided, multi-stage interconnection networks are also possible. Fig. 4.4 illustrates a single sided Clos structure. Both switches and links are bi-directional and all connections to the network may act as inputs and outputs. The TDM bus fast packet switch [36, 35] suggests the use of a single sided network but most fast packet switch designs use a two sided multi-stage interconnection network.

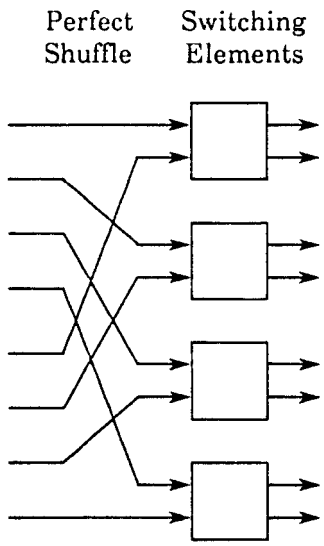

Figure 4.3: A single stage 8×8 shuffle exchange.

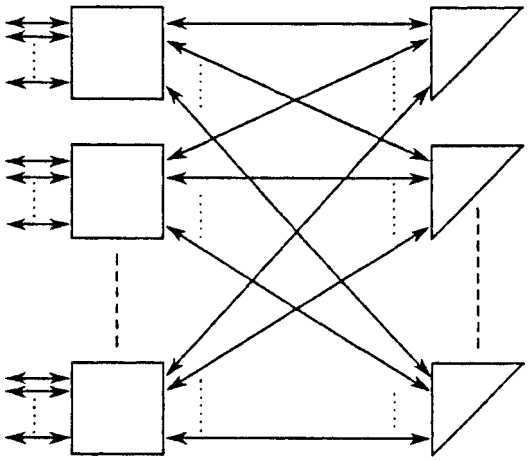

Figure 4.4: A single sided Clos network.

Concluding the discussion of the classification of networks introduced in fig. 4.1, the two sided multi-stage networks may be classed as either single path or multi-path. As might be expected, in a single path network only one unique path exists between any input/output pair but a choice of paths is available in a multi-path network. The Batcher sorting network has been included as a single path network because, for any given permutation of input to output requests, no choice of paths exists through the network. All of the examples of two sided multi-stage networks listed in fig. 4.1 will be discussed later with the exception of the PM2I class of networks [47, 99, 131]. This is also known as the data manipulator class of networks. It has a small number of multiple paths but nowhere near as many as the Clos or Beneš

networks. Its major use is in manipulating sets of data within a tightly coupled parallel processor although it has been suggested for use in the concentrator function of the Starlite switch [70]. It is not easily partitioned for VLSI implementation, it is less flexible, and its routing algorithm is more complex when compared to other multi-stage interconnection networks.

Control Mechanism

Interconnection networks may also be classified according to the control mechanism employed to effect connections between input ports and output ports. If the algorithm is centralised and implemented in a central processor then the state of all existing connections and all connection requests may be consulted in order to make the necessary routing decisions. The use of a centralised control mechanism implies circuit switching where the holding time of a connection is much greater than the time required to establish connection. The vast majority of modern telephone switch designs use centralised control.

In fast packet switching applications the control mechanism must be distributed across the switch fabric and must be capable of operating without access to information regarding the entire state of the switch. Three classes of distributed routing algorithm are relevant to a regular network: source routing, self-routing and regular routing. Source routing requires a tag to be prefixed to the packet which specifies all of the routing decisions to be taken within the network, one field of the tag for each switch in the path. It thus removes the burden of route computation from within the switch fabric to the periphery. The self-routing and regular routing control mechanisms are sometimes confused as both require a tag to be prefixed to the packet specifying the required destination output port number and both rely upon the regularity of the interconnection network. Self-routing applies to dynamic, multi-stage interconnection networks. It may be implemented such that each switching element within the path makes a simple routing decision based only upon the tag of the incoming packet independently of the position of the switching element within the interconnection network. The regular routing mechanism applies to regular static networks in which each network node makes a routing decision based upon the packet tag and the position of the node within the network. This decision requires a certain amount of computation and thus involves some delay. The regular routing algorithm is thus best suited to conventional packet switching applications. The routing decision in a self-routing algorithm, however, requires no computation, does not involve the maintenance of routing tables within the switch fabric, and may be executed by very simple hardware within a single bit time. It is therefore of considerable interest in fast packet switching applications.

Blocking Characteristics

Multi-stage interconnection networks may be further classified according to the blocking characteristics they present which is reflected in the throughput they offer to traffic

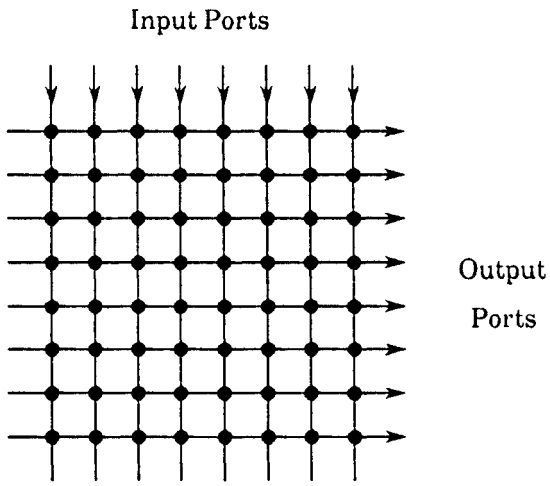

Figure 4.5: The general representation of a crossbar network.

with a random distribution of packet destinations. A network which is always capable of connecting a free input to a free output, regardless of the connections already established across the network, is said to be non-blocking. The crossbar and Clos networks are examples of non-blocking networks. A network that is always capable of connecting a free input to a free output, but which may require existing connections to be rearranged in order to do so, is called rearrangeable non-blocking. The Beneš network is an example of a rearrangeable non-blocking network. A network is said to be blocking if any free output may be unavailable to any free input because existing connections prevent a path from being established across the network. The banyan network is blocking to traffic with a random destination distribution.

As might be expected a non-blocking network requires more switching elements and interconnections than does a rearrangeable non-blocking network which in turn requires more than a blocking network. Also the throughput of a fast packet switch depends upon the blocking characteristics of its switch fabric. Finally a rearrangeable non-blocking network only provides non-blocking performance if a centralised control algorithm is available to perform the rearrangement of connections. For fast packet switching applications only distributed algorithms may be employed. It is therefore interesting to consider the improvement in performance that a rearrangeable structure might offer, for various distributed control algorithms, when compared to a blocking network. Three such algorithms will be examined in chapter 6.

4.2 The Crossbar Network

The crossbar network, or more often crossbar switch, is a non-blocking interconnection network that derives its name from a particular switch implementation developed for analogue telephone switching applications. The name is often taken to refer to non-blocking networks in general. Its structure is usually represented as shown in fig. 4.5.

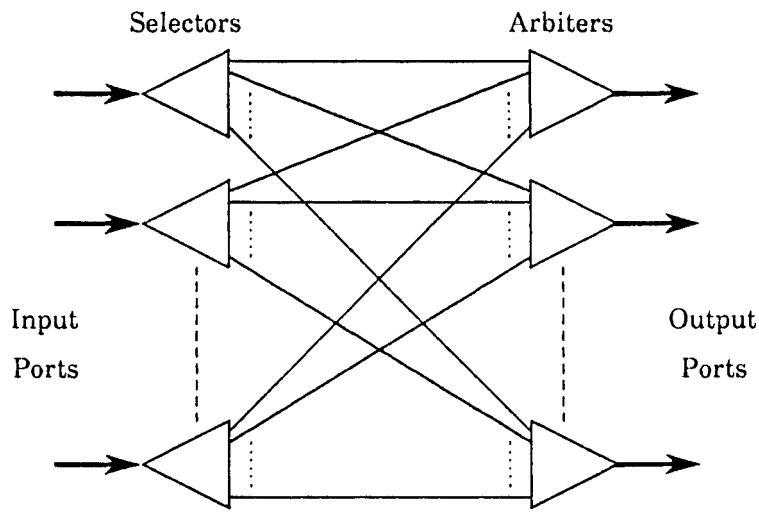

Figure 4.6: A self-routing crossbar switch.

Each node in the network is called a crosspoint and is a simple switch which has two states, open and closed. Using centralised control the network can satisfy all one-to-one (unicast) and one-to-many (multicast) connections. It requires $O(N^2)$ crosspoints and thus the hardware required to implement the network grows rapidly with the size of network.

Multi-stage interconnection networks are constructed from stages of interconnected crossbar switching elements of low degree. Fig. 4.6 illustrates an alternative design of crossbar switching element suitable for use with distributed control within the environment of a multi-stage interconnection network. An incident packet is prefaced by a tag indicating the required destination. The selector of the input port examines this tag and inspects the state of the arbiter of the required output port. If the selected arbiter indicates that the required output port is free the connection is established but if busy it is refused. All selectors may thus work concurrently and asynchronously. Multicast connections are not supported by this design of crossbar network. This crossbar network may be used with either a self-routing or with a source routing control algorithm and may be referred to as a self-routing crossbar switch.

4.3 Banyan Networks

Definition

The banyan network is a multi-stage network of interconnected crossbar switching elements originally defined in graph theoretic terms in [57] and named after the East Indian fig tree whose structure it is supposed to resemble. It is defined as having one and only one path from any input to any output and thus covers a very large class

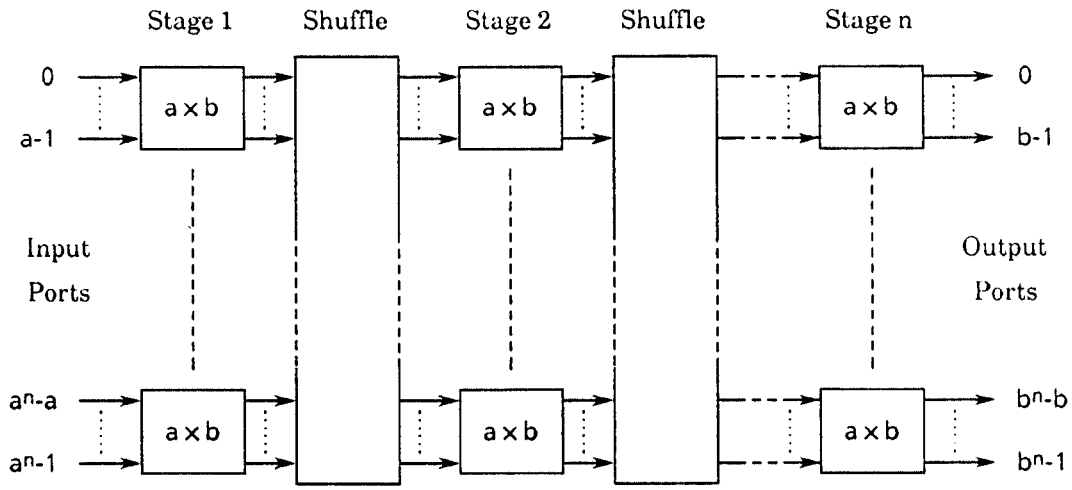

Figure 4.7: The general structure of a delta network.

of possible network structures. If the links in the banyan network are constrained to connecting switching elements in adjacent switching stages an L-level banyan results and if in addition all switching elements in the network are identical we get a regular banyan. A banyan network with square switching elements, i.e. those which have the same number of inputs as outputs, is called rectangular. Two classes of regular banyan are of specific interest, SW and CC banyans. The CC-banyan is rectangular by definition and its structure bears some resemblance to the PM2I class of networks. The SW-banyan can be shown to be self-routing and as such the rectangular SW-banyan constructed from 2×2 crossbar switching elements is almost invariably the network envisaged when the term 'banyan' is used in the literature of recent years. This is unfortunate as the term 'banyan', as originally defined, covers a much wider class of network.

Shortly after the definition of banyan networks the omega network was introduced [86] which formed a multi-stage interconnection network from the single stage shuffle exchange of [137]. The omega network was the first multi-stage interconnection network to demonstrate the self-routing property. A number of similar networks followed the omega network until in [126] the delta network was defined.

Delta Networks

The delta network is a multi-stage interconnection network which may be defined as a subset of the class of regular banyan networks that displays the self-routing property. Delta networks therefore form a class of interconnection networks which includes the SW-banyan, the omega network, the flip network, the indirect binary n-cube, the baseline and the reverse baseline networks which have all been proven topologically equivalent in [158]. The general structure of a delta network is given in fig. 4.7 in which stages of identical, but not necessarily square, switching elements are connected by

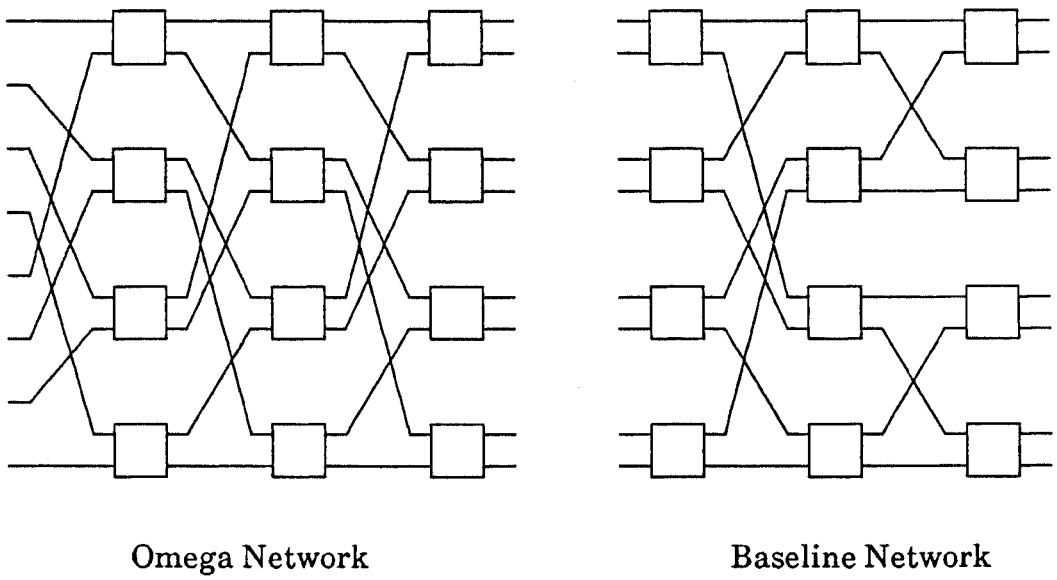

Figure 4.8: Examples of 8×8 delta networks constructed from 2×2 switching elements.

an interconnection pattern of links sometimes referred to as a permutation network but also called a shuffle. Hence the term 'multi-stage shuffle exchange' which also refers to this class of networks. In general the shuffle between each stage of the delta network will be different and it is this pattern of interconnection links that gives rise to the self-routing property. A delta network of some interest is the rectangular network in which the shuffle between each stage is identical. This in fact forms the omega network and if constructed from 2×2 switching elements the shuffle becomes the well known perfect shuffle. Two examples of an 8×8 delta network, constructed from 2×2 switching elements are given in fig. 4.8, the omega network and the baseline network [158] and an example of a 64×64 delta network constructed from 8×8 switching elements may be found in fig. 5.3.

A rectangular delta network of size $N \times N$ constructed from square switching elements of degree d requires $\log_d N$ switching stages with N/d switching elements per stage. It thus requires $O(N \log N)$ switching elements. An identical shuffle may be used for the interconnection pattern of links between each switching stage for networks built from any degree of switching element and may be constructed as follows. Label the N output ports of stage k from 1 to N. Label the switching elements of stage $k+1$ from 1 to N/d and the input ports of each of these switching elements from 1 to d. Connect the output ports 1 to N/d of stage k to input port 1 of each of the switching elements 1 to N/d in stage $k+1$. Connect the output ports $N/d+1$ to $2N/d$ of stage k to input port 2 of each of the switching elements 1 to N/d in stage $k+1$ and so on, a total of d times, until all N output ports of stage k are connected to all N input ports of stage $k+1$.

The general delta network is formally proven to be self-routing in [126] but for a rectangular delta network of crossbar switching elements of degree d it functions as

follows. The required destination port number is expressed numerically to the base d and will require $\log_d N$ digits, one digit for every stage of the delta network. The required output port number is prefixed to the packet as a tag and the packet inserted into the network. The most significant digit of the tag is used by the switching element in the first stage of the network to select the output port over which to transmit the packet. The second digit in the tag is used by the switching element in the second stage, and so on for each stage of the network until the least significant digit of the tag is reached at the last stage of the network. Each digit may be removed from the tag as it is used by a switching element or the whole tag may be rotated such that the digit at the front of the tag is always the one required by the next stage of switching elements. Each switching element merely selects the output port specified by the digit at the head of the tag. The pattern of interconnection links between stages in the network is so arranged that the packet will exit from the correct destination port provided that it is not blocked anywhere within the network. Each switching element within the network functions as a simple self-routing crossbar switch and the packet will exit from the correct port regardless of the input port of the delta network at which it originated.

Delta networks have been modified to introduce multiple paths through the network to increase the reliability or to enhance the throughput. The proposed methods include adding extra paths with switching elements of higher degree [87]; adding extra stages [124, 1, 100]; or by connecting multiple delta networks in parallel [85, 82].

4.4 The Clos Network

The Clos network was developed to satisfy the needs of the telephone switching industry for a non-blocking network that uses fewer crosspoints than the crossbar network of the equivalent size. It is a multi-stage interconnection network of crossbar switching elements and a square Clos network is illustrated in fig. 4.9. Clos has shown that the network is strictly non-blocking if the condition $m \geq 2n - 1$ holds [30]. The Clos network may be recursively decomposed into a five stage network, a seven stage network and so on by replacing each switch in the central stage by a three stage Clos network. The three stage non-blocking Clos network has fewer crosspoints than the equivalent crossbar network for all $N \geq 36$ and a growth of $O(N(\log N)^{2.27})$ crosspoints.

4.5 The Beneš Network

The Beneš network is a special case of the Clos network for which Beneš has shown that if $m \geq n$ the network is rearrangeable non-blocking [13]. If $n = m$ then for networks of size $N = n^j$, where j is an integer, the $r \times r$ switches of the central stage of the Beneš network may be substituted by three stage Beneš networks recursively until a structure results in which all switching elements are of degree n. An 8×8 Beneš network of 2×2 switching elements is illustrated in fig. 4.10 and by comparison

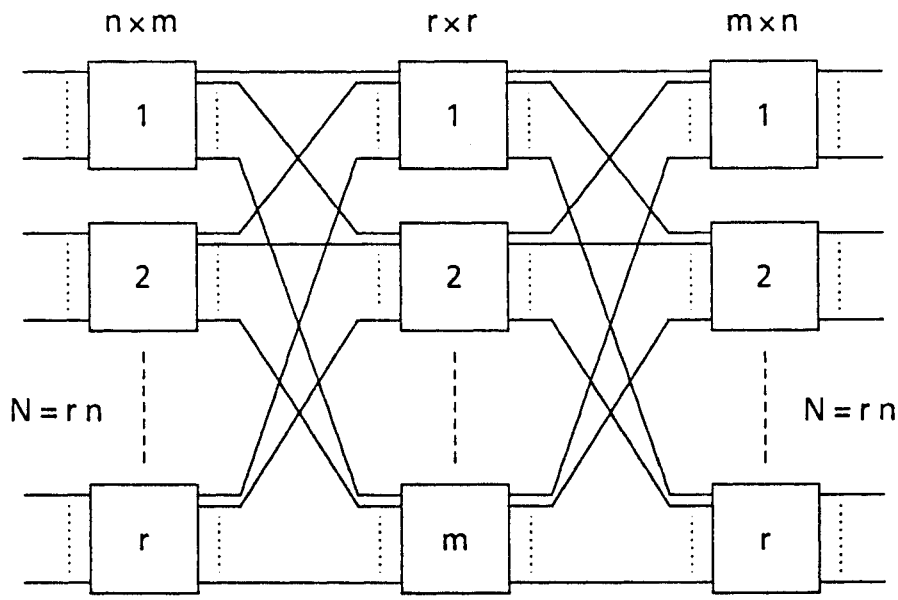

Figure 4.9: A square three stage Clos network.

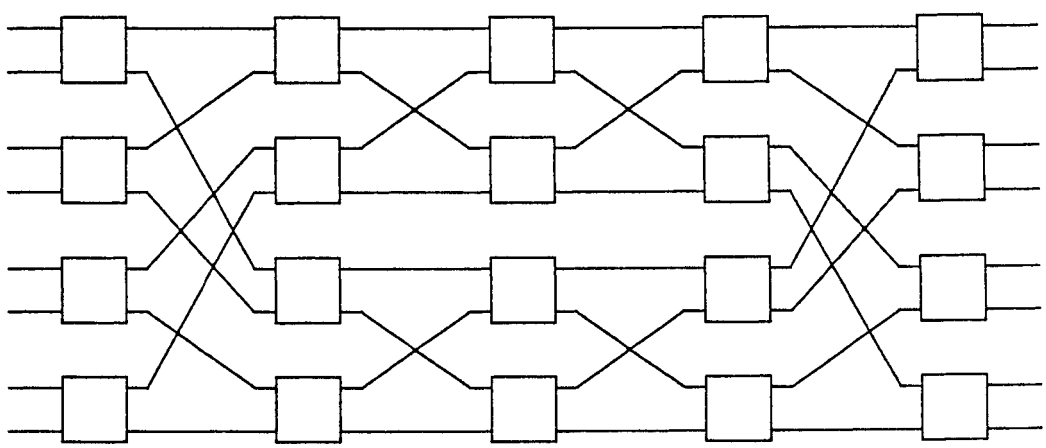

Figure 4.10: An 8×8 Beneš network constructed from 2×2 switching elements.

with fig. 4.8 it may be seen that the Beneš network may be formed by reflecting the equivalent baseline network about the central stage. An $N \times N$ Beneš network requires $2\log_d N - 1$ stages of switching elements of degree d, (i.e. $n = m = d$), with N/d switching elements per stage and also has a growth of $O(N \log N)$.

Only the final $\log_d N$ switching stages are required to provide the routing function in the Beneš network thus the additional $\log_d N - 1$ stages offer multiple paths through the network. Indeed the Beneš network may be considered as a delta network preceded by switching stages which distribute the incident traffic across the switch fabric making use of the multiple paths to offer fault tolerance and to reduce

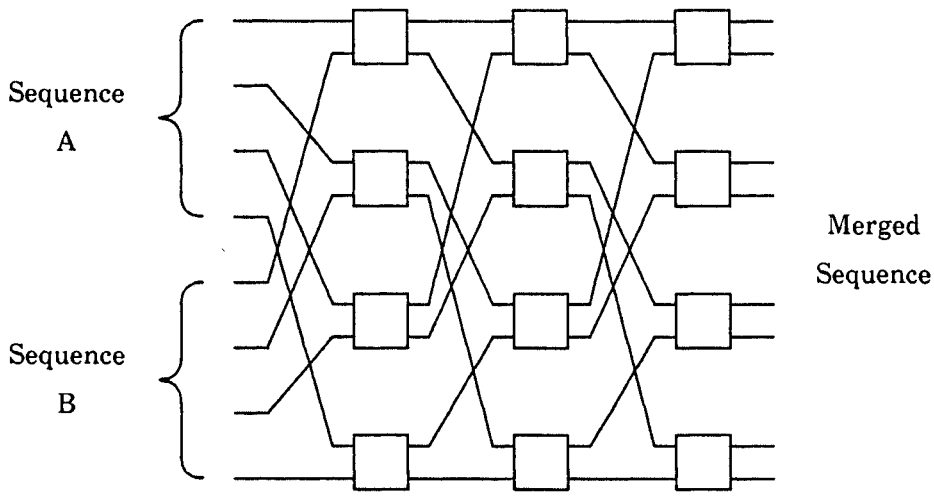

Figure 4.11: An 8×8 bitonic sorter.

blocking. If a centralised control algorithm is employed the network may operate in a rearrangeable non-blocking mode but the algorithm required is both time consuming and centralised.

Various options are possible for the use of a distributed algorithm across a Beneš network. The routing stages of the network may use the same self-routing algorithm as for the delta network, based upon the use of a destination routing tag. The distribution stages of the network may be switched according to three possible algorithms: source routing, random routing or flooding. In source routing the tag is extended to explicitly direct the switching of the distribution stages in exactly the same manner as the routing stages. If one path proves busy then another is selected and attempted from the periphery of the switch fabric. The random routing algorithm allows the distribution stages of the switch fabric to select any free path to the subsequent switch stages at random. The flooding algorithm sends copies of the incident packet concurrently across all free paths that lead to the required destination in the knowledge that only one path to the destination will be accepted. All other copies will fail and will quickly be removed from the network. All three algorithms have been investigated and results are presented in chapter 6.

4.6 The Batcher Sorting Network

A Batcher network will sort any arbitrary sequence of numbers into ascending (or descending) order [12]. It is constructed from a network of bitonic sorters each of which is a multi-stage interconnection network constructed from 2×2 comparison elements. A comparison element receives two numbers synchronously and in bit serial form at its two input ports and outputs the greater number on one output port and the lesser on the other. An 8×8 bitonic sorter is shown in fig. 4.11 which takes

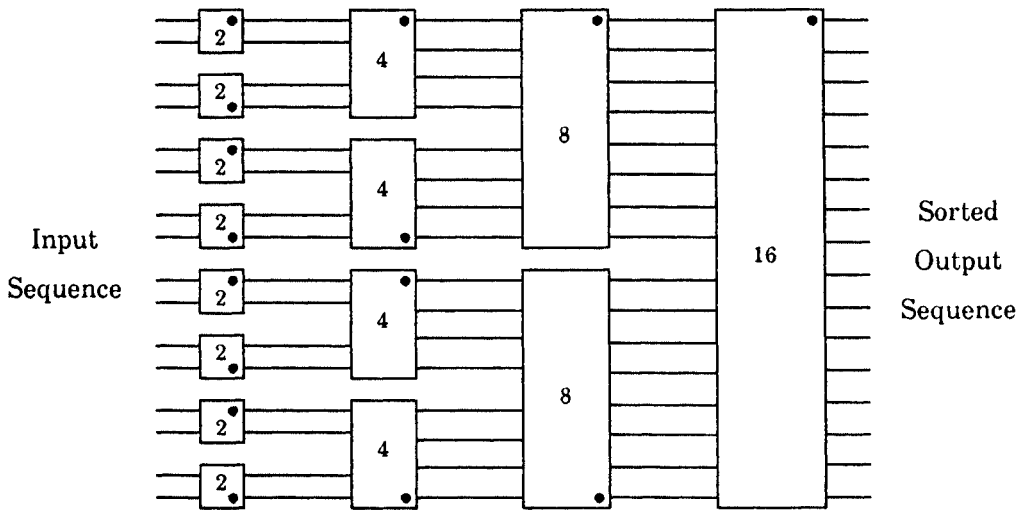

Figure 4.12: A 16×16 full sorter.

two monotonic sequences of numbers, one ascending and the other descending, and merges them into a single sorted sequence. All comparison elements must be aligned to sort the higher number into the upper (or lower) port to produce an ascending (or descending) sequence. The topology of the bitonic sorter is exactly that of the omega network for 2×2 switching elements and the interconnection pattern of links between each stage is the perfect shuffle.

To form a full sorting network a multi-stage network of bitonic sorters must be arranged as shown in fig. 4.12. Each element is a bitonic sorter of the size indicated, with the dot signifying the port at which the highest number of the output sequence will exit. The full sorting network requires $\frac{1}{2}\log_2 n(\log_2 n+1)$ stages of $n/2$ comparison elements per stage and has a growth of $O(N(\log N)^2)$. The comparison elements are, however, relatively easy to construct.

In switching networks the full sorter is used to sort according to the tag at the head of each packet and the state of the network is latched while the packets are transmitted across the network. The Batcher network is of interest in fast packet switching applications because a non-blocking network may be formed by a Batcher sorting network followed by a banyan routing network. Further details on the construction of sorting networks, with reference to their use in a Batcher-banyan switch fabric, may be found in [70, 34, 107].

4.7 Summary

The most appropriate interconnection network for applications within the switch fabric of a fast packet switch is the two sided multi-stage interconnection network. It supports a simple distributed self-routing control algorithm and the distance between

all inputs and outputs is constant. Such networks may be classified according to their performance into non-blocking, rearrangeable non-blocking and blocking networks. The non-blocking network offers ideal performance and may be constructed from a crossbar network, a Clos network or a Batcher sorting network followed by a banyan routing network (the Batcher-banyan switch fabric). The Beneš network is rearrangeable non-blocking but its non-blocking properties may only be attained by the use of a centralised control algorithm. It offers multiple paths and its performance under various distributed control algorithms is of interest for fast packet switching applications. The delta network forms a subset of the more general class of banyan networks that offers the self-routing property. It is the simplest of the self-routing multi-stage interconnection networks discussed but offers blocking performance.

Chapter 5

Design of the Cambridge Fast Packet Switch

In this chapter the design of the Cambridge Fast Packet Switch is considered in detail. A review is first presented of some early work on binary routing networks. Although this work was developed in the context of the local area network it introduces some ideas on high-speed switching mechanisms which were to influence the design of the fast packet switch. The discussion is then widened to introduce the general principles which guided the switch design. Finally, the switch itself is described with some of the design options whose influence on the performance of the switch will be presented in the following chapter.

5.1 Binary Routing Networks

The Buffered Binary Routing Node

Binary routing networks were first proposed about ten years ago by Hopper and Wheeler [69] as a new class of local area network with simpler hardware and better performance than the existing bus or ring designs. The networks were based upon the buffered binary routing node as shown in fig. 5.1. The value of the first bit in the route field of an arriving packet was used to direct the node to switch the packet either to the upper port if zero or to the lower output port if one. The whole route field of the packet was rotated by one bit so as to present the next bit of the route field to the succeeding node. If the selected output was free the packet was transmitted across the node with little delay but if it was busy the packet was buffered and a backward busy signal asserted to prevent the transmission of further packets, on the relevant input port, until the buffered packet was transmitted. This formed one of the first proposals to employ a buffered 2×2 self-routing crossbar packet switch in a communications network. However, as the application envisaged was that of local area networks, most of the topologies considered for constructing binary routing networks were buses, rings and trees which distributed the switching nodes across the local

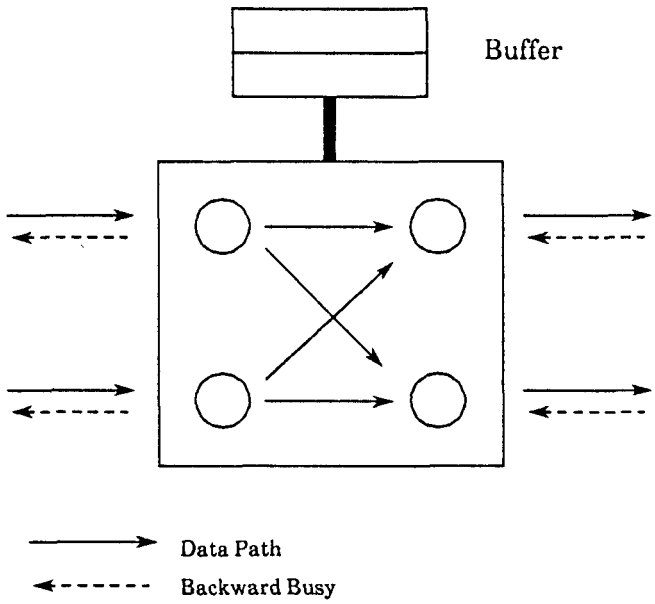

Figure 5.1: The structure of a buffered binary routing node.

area. The switching mechanism thus became source routing rather than self-routing and the networks were suitable for operation only at relatively low loads. A binary routing network based upon a shuffle exchange network, to form a fast packet switch, was foreseen but the idea was not developed.

Non-Buffered Networks

Between 1981 and 1983 the author was involved in a project to investigate possible switching mechanisms for the interconnection of high-speed optical fibre transmission links. The work of Hopper and Wheeler was taken as a starting point but it soon became clear that the implementation of a buffered switching node would prove expensive and difficult, at the speed required, in the technology available at that time. If possible, a solution capable of implementation in gate array devices was sought. A switching mechanism based upon a non-buffered binary routing node was therefore proposed in [110, 112]. The buffer was removed from the binary routing node and the backward busy signal modified to become a reverse path through the network, back to the originating node, established in parallel with the forward path. If a packet incident at a node should find that the desired output was busy the reverse path was used to inform the originating node of the contention with very little delay. On receiving a contention signal the originating node removed this attempt to transmit the packet and tried again a short while later.

In [111] a self-routing crossbar switch was proposed according to the design presented in fig. 4.6 and the use of this switching element in both delta and Beneš networks was suggested but no attempt was made to follow this up with any perfor-

mance investigations. A detailed investigation into the design and construction of a non-buffered binary routing node was reported in [113] in which a hardware model was constructed in TTL operating at 10 MHz. From this model it was clear that a non-buffered binary crossbar switching element could be fabricated with less than 400 gates and that crossbar switching elements of degree 8 or possibly 16 could be achieved.

Binary Shuffle Exchange Networks

The suggestion in [69] that a fast packet switch might be constructed from a binary routing network in a shuffle exchange topology was investigated by Milway [100] for use in local area network applications. He considered the performance of both buffered and non-buffered designs and concluded that within the context of the local area network the performance of the non-buffered design was adequate. He also considered the improvement in performance that could be obtained by a form of input queue by-pass algorithm and also by adding additional stages of switching to the front of networks which selected at random from the available paths to the destination. As the work was directed to the use of binary routing networks in the context of the local area network, much of the investigation was directed towards operation at low loads and only switching elements of degree 2 were considered. The hardware implementation of a non-buffered binary routing node was investigated using programmable logic array devices operating at 10 MHz and the performance of a 4×4 network of binary routing nodes was shown to agree well with performance results derived from a simulation model.

5.2 Design Issues

Output Buffered Switching Elements

A fast packet switch must be constructed from either buffered or non-buffered switching elements. Considering the existing switch designs about half of them propose the use of a buffered switch fabric using output buffered switching elements [141, 120, 4, 36]. This results in a very complex design of switching element requiring a large number of buffers on each output of every switching element in the switch fabric if packet loss is to be kept to acceptable levels. Such a design is not unreasonable considering the current state of the art in VLSI technology yet a simpler design of switching element may be worth investigating for a number of reasons. First, a simple design of switching element is likely to be suitable for fabrication in a wide range of implementation technologies. Thus a simple but high-speed switch may well be capable of a greater capacity. Looking toward the long term, with a simple design of switching element it may be possible to implement the data paths of the switch fabric using integrated optics [14, 134, 64] leading to very high-speed operation. If it is possible to implement the switching element using gate array technology a more flexible switch design will result. Thus the resulting switch design may be much more

easily customised to fit each environment that it is commissioned to serve. Hence, parameters such as line code, priority levels, packet length, etc. may be modified much more easily than in a fixed VLSI design. Finally, the fact that most other switch designs employ a VLSI implementation makes a design capable of implementation in small gate arrays an interesting subject for investigation.

Input Buffered Switching Elements

A design based upon an internally buffered switch fabric may still be practicable with the requirement for a simple design of switching element if the switching element is input buffered. Two designs [148, 128] based upon input buffered switching elements have already been investigated. Also, Milway has shown [100] that even with infinite buffers at both inputs of a 2×2 buffered switching element, the performance of the resulting switch is only slightly in excess of that of an input buffered non-blocking switch fabric. If a simple buffered design is selected, the size of the switching element is likely to be limited to 2×2. Thus even if the buffering were implemented in memory external to the logic of the switching element, performance is unlikely to greatly exceed that of the input buffered non-blocking switch fabric.

A switch fabric constructed from switching elements of degree two presents another difficulty. The number of switch stages is maximised, hence the number of interconnections within the switch fabric is also maximised. It is likely that in any implementation, the number of interconnections required in the switch fabric is going to limit the maximum size of switch that can be realised. Thus, not only would a simple switching element design be desirable but also the size of that switching element would preferably be of a degree greater than two. If not, the switch fabric should be capable of partitioning such that a standard array of switching elements may be implemented in a single device from which the switch fabric may be constructed. This is the approach taken in the Batcher-banyan switch fabric [34] but it has resulted in the design of several different VLSI devices to implement the switch fabric.

Non-Buffered Switching Elements

Considering the possibility of a non-buffered switch fabric, it is clear that if simplicity of implementation is a major requirement an input buffered switch is a much more suitable candidate than an output buffered switch. The only major design of input buffered fast packet switch so far proposed for communications applications, the three phase Batcher-banyan [71], uses a non-blocking switch fabric. This requires a much larger switch fabric than is needed for a simple blocking switch fabric and thus necessitates implementation in VLSI. Also the algorithm required to ensure non-blocking operation renders the use of short packet lengths extremely inefficient. Hence the question of the performance of the various possible blocking switch fabrics in comparison with that of the non-blocking switch fabric is of considerable interest. Further, a number of techniques are available to enhance the performance of an input buffered switch fabric, for example input queue by-pass and a multi-plane switch

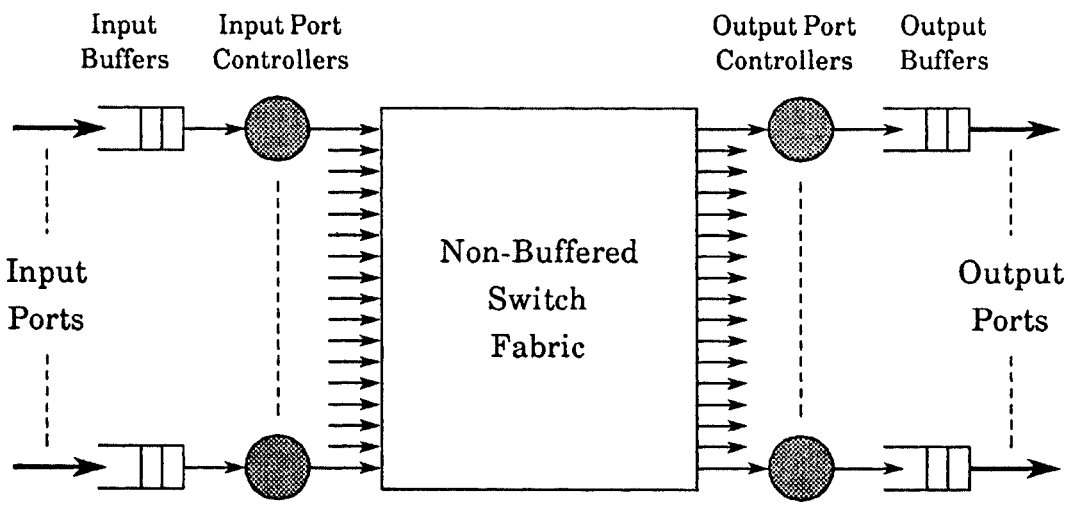

Figure 5.2: The basic structure of the Cambridge Fast Packet Switch.

fabric with output buffering across the multiple switch planes. It may be interesting to investigate how close the use of these techniques may improve the performance of a blocking switch fabric towards that of an output buffered switch. (The output buffered switch with infinite output queues represents the best possible theoretical switch performance.)

5.3 The Switching Mechanism

At the lowest level, a fast packet switch will generally use a connection-oriented switching mechanism in which a virtual circuit is established between source and destination before communications traffic is exchanged. Thus at each input port of the switch a table is maintained which lists the parameters of each active connection and assigns a label to each connection. The label appears in the header of every incoming packet and is used to identify the connection to which each packet belongs. In general a connection will traverse several switches. Hence, to avoid problems of global label allocation, each switch allocates its own labels to refer to the connections that it supports. The label of every packet must therefore be translated as it traverses each fast packet switch in the path. This operation is generally performed at the input port with the replacement label stored as a field in the connection table. Other fields in the table will include the output port of the switch through which the connection is routed and possibly other parameters such as priority and traffic type.

Connections are set up across a network of fast packet switches using a message based signalling system similar to that of a modern telephone network. However, as connections are virtual, no system bandwidth is exclusively allocated to any connection, so connections may lay idle for substantial periods of time at little cost to the network.

The basic structure of the Cambridge Fast Packet Switch is given in fig. 5.2. The connection tables are housed in the input port controllers while in a simple switch the output port controllers are required for little more than line conditioning and similar tasks. In the basic switch, output buffers would only be required in order to interface to output lines running at a different rate from that of the switch fabric. An incoming packet arrives in the input buffer which is a first in first out (FIFO) queue. When free, the respective input port controller extracts the label from the packet at the head of the queue and uses it to reference the connection table. Each input port controller operates asynchronously, at the packet level, and independently of all other controllers. From the table it receives two components, an outgoing label and a tag. The outgoing label is used to replace the incoming label within the packet. The tag specifies the required destination output port of the switch and is attached to the front of the packet. The input port controller then initiates a set-up attempt by launching the packet into the switch fabric, tag first and in bit serial form. There are two possible outcomes, either the packet will be successful and reach the desired output buffer, or it will fail. A set-up attempt may fail either because it is blocked by other traffic within the switch fabric or because the requested output port is busy serving another packet. If the set-up attempt fails, the switch fabric will assert a collision signal which is returned to the input port controller, along a reverse path, typically within a few bit times of emission of the packet tag. On receiving the collision signal the input port controller removes the set-up attempt from the switch fabric and waits for a delay typically equivalent to 10% of the length of a packet. This is the retry delay and at the end of this period the input port controller begins a fresh attempt to transmit the packet. It continues to do so until it is successful or until it exceeds a limit designed to detect fault conditions.

A slightly more complex algorithm that offers an improvement in performance at high loads does not repeatedly attempt to transmit the same packet but on the failure of a set-up attempt searches through the input queue and attempts to transmit the second packet. If that attempt fails the third packet on the queue is attempted and so on cyclically through the queue until a successful transmission is achieved. This overcomes the so called 'head of the line' blocking problem [71, 76] but care has to be taken not to get packets on the same virtual circuit out of sequence. This algorithm will be referred to as input queue by-pass [19].

A simple model of the operation of the fast packet switch may be drawn by analogy with the operation of a well known local area network: Ethernet. Ethernet may be considered as a fast packet switch which distributes the switching function across the local area using a single shared medium switch fabric. The fast packet switch described above merely confines the switching function within a box so that a multi-path medium of much higher bandwidth may be implemented. The input port controller of the fast packet switch corresponds to the media access controller of Ethernet and in both cases the controller launches a packet into the switch fabric and if it is unsuccessful the switch fabric informs it immediately. The difference between the two lies in the fact that in Ethernet a collision destroys both colliding packets therefore an exponential random back-off algorithm is required. In the fast packet

switch, however, collisions are non-destructive in the sense that one of the colliding packets always survives, so a simple retransmission algorithm is sufficient.

5.4 The Switch Fabric

The switching mechanism described above requires a self-routing switch fabric. The switch fabrics proposed for investigation are therefore constructed from multi-stage interconnection networks of crossbar switching elements. The switching mechanism relies upon the ability of the switch fabric to inform the input port controllers of a collision between a packet attempting set-up and one already established. This is achieved by setting up a reverse path through the switch fabric in parallel with the forward path. Every link in the interconnection network consists of two paths, a forward path to carry the data and a reverse path for the collision signal. Every switching element sets up both paths in parallel provided that the required output on the forward path is free. If blocked, it will return a collision signal on the reverse path and all switching elements from the partially established path will return to the idle state as soon as the input port controller removes the failed packet from the switch fabric.

Three classes of self-routing switch fabric present themselves as deserving of investigation under the above switching mechanism: non-blocking, rearrangeable non-blocking and blocking. Although several methods of constructing a non-blocking switch have been demonstrated they tend to be expensive in terms of their hardware requirements. This class of switch fabric will thus be used as a standard against which other switch fabrics may be compared.

The Beneš network belongs to the class of rearrangeable non-blocking networks when controlled by a centralised switching algorithm. Its performance under the various possible distributed algorithms is of interest. The delta network seems the most appropriate class of blocking, self-routing switch fabric to bear closer scrutiny.

Whilst the majority of research interest has been expended upon multi-stage interconnection networks constructed from 2×2 switching elements, previous investigations suggested that it might be possible to implement crossbar switching elements of up to degree 16 in gate array technology [113]. It is therefore of considerable interest to investigate what effect the degree of the switching element has upon the performance of both delta and Beneš switch fabrics.

Delta Networks

An example of a 64×64 delta network constructed from switching elements of degree 8 is given in fig. 5.3. The self-routing property is easily demonstrated. The output ports of each switching element are numbered from 0 to 7 with the uppermost port 0. The output ports of the switch fabric are numbered from 0 to 63 with the uppermost port 0. If the number of the required output port is expressed to the base 8, two digits will be required. If the most significant digit is used to select the output of the

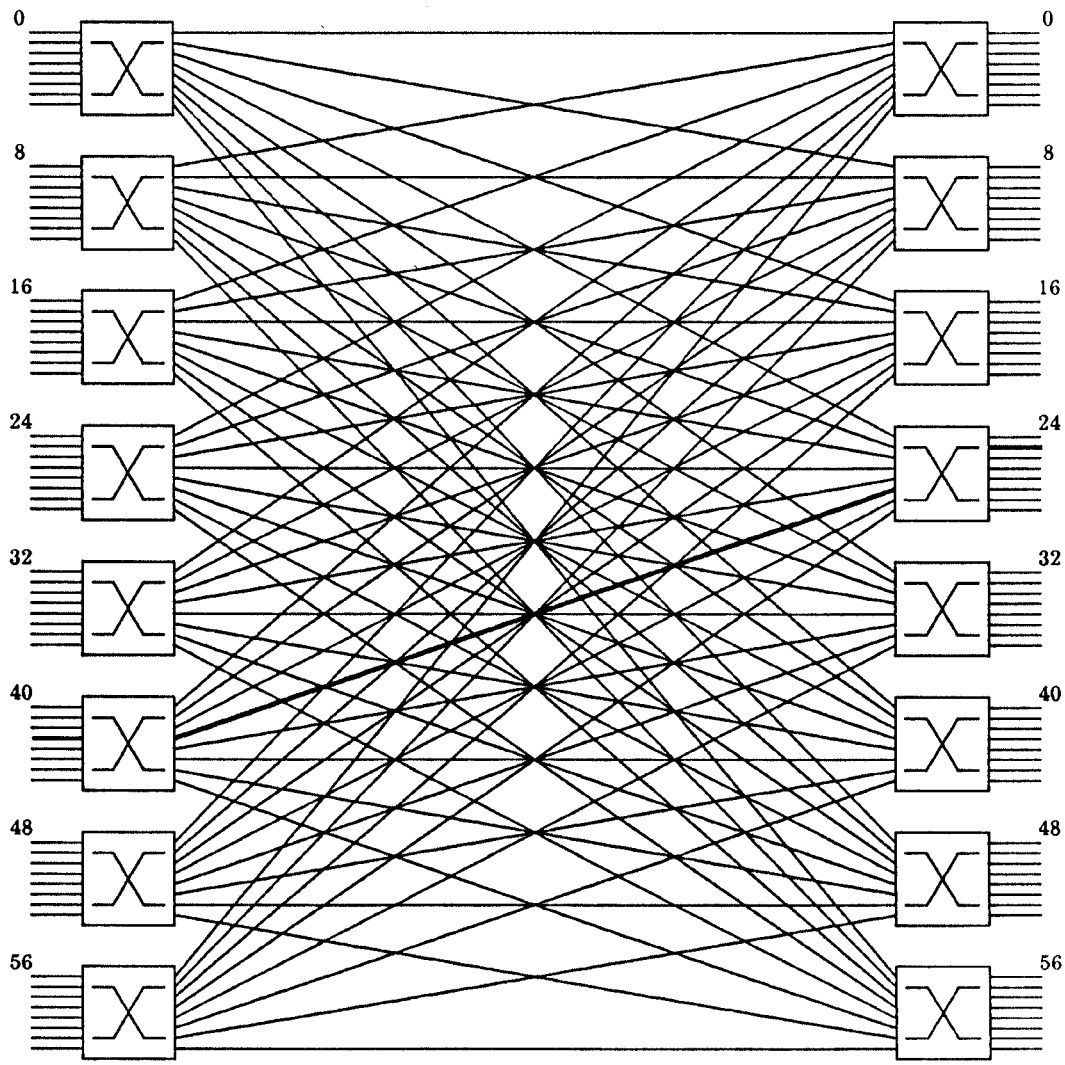

Figure 5.3: A 64×64 delta network of 8×8 switching elements.

switching element of the first stage and the least significant digit that of the second stage, a path to the required output port is established. This is true regardless of the input port from which the path originates. A path from input port 43 to output port 25, (i.e. output port number 31 to base 8,) is illustrated.

The use of switching elements of degree greater than 2 raises the problem that delta networks are only defined in sizes that are an integer power of the degree of the switching element. This would result in large increments between valid sizes of network. The proposed solution is to replicate the interconnection links between stages which permits networks to be built to any size that is an integer power of 2, from switching elements of any degree that is also an integer power of 2, [1, 85]. Thus a modified delta network of size N requires s stages, where $s = \lceil \log_d N \rceil$, of N/d switching elements per stage and each link of the pure delta network is replicated

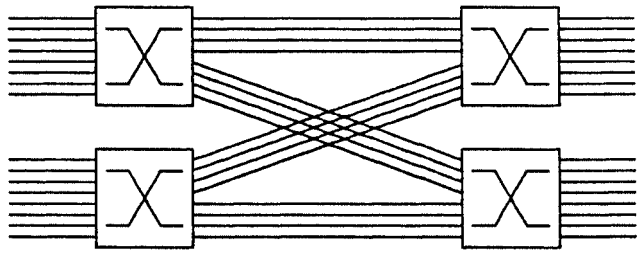

Figure 5.4: A 16×16 modified delta network of 8×8 switching elements.

d^s/N times.[1] Strictly speaking the modified delta network is no longer a member of the class of banyan networks and fig. 5.4 illustrates a 16×16 modified delta network of switching elements of degree 8. Clearly this network offers four equivalent paths between every input/output pair.

We now have the possibility of multiple paths existing between the same pair of input and output ports. This increases the performance and fault tolerance of the switch but requires an algorithm to select between equivalent paths. Fortunately, as there is no buffering within the switch fabric, each incident packet may be routed independently across the switch fabric without the risk of out-of-sequence errors between packets travelling on the same virtual circuit. Two algorithms have been investigated: searching and flooding. In the searching mechanism the input port controller attempts to transmit across each of the equivalent paths in sequence until it meets with success. It does this simply by incrementing the most significant bits of the packet tag. In the flooding method, the incoming packet is broadcast simultaneously over all free paths that lead to the destination such that the destination selects one of the incident copies. All other copies of the packet collide with each other and are removed from the switch fabric immediately, after the transmission of only a few bits. To implement this algorithm the switching elements in the first stage of the switch fabric would have to be modified to broadcast each incoming packet over all of the relevant paths. For any size of delta network modified to offer multiple paths in this way, all of the equivalent paths may always be selected from any switching element within the first stage of the switch fabric. Also there can never be more than $d/2$ equivalent paths to any single output port.

The Beneš Network

The delta network offers an acceptable performance for traffic which has a random destination distribution but its performance can be markedly impaired for incident traffic with a worst case distribution of destinations [159]. This is easily seen from fig. 5.3 as all connections between input ports 0 to 7 and output ports 0 to 7 share a single common link and likewise for ports 8 to 15 and so on. For this network the identity connection, in which every input port requests connection to every output

[1] $\lceil x \rceil$ signifies the smallest integer equal to or greater than x.

port of the same port number, represents the worst case traffic pattern. For this traffic pattern a total of only N/d connections may be established even though there is no contention for output ports. For some applications this sensitivity of the switch fabric to the destination distribution of the incident traffic may not be significant. For high performance switches, however, and in order to handle traffic sources which have an average bandwidth in excess of about 10% of the switch port bandwidth, extra stages of switching must be introduced to distribute the incident traffic across the switch fabric. These additional stages of switching are often termed the distribution fabric and the self-routing stages of the switch fabric are called the routing fabric. In order to fully distribute the incident traffic, with any arbitrary destination distribution, across an entire s stage delta network requires $s-1$ distribution stages and results in a Beneš topology. Thus the Beneš network is of interest because it offers increased throughput and also removes the sensitivity of the switch fabric to the destination distribution of the incident traffic. Due to the number of multiple paths through the Beneš network it may also increase the reliability of the switch fabric through fault tolerance.

A 64×64 Beneš network of switching elements of degree 8 is illustrated in fig. 5.5. For a network whose total size is an integer power of the degree of the switching elements, all switching elements will be of the same degree d and the number of equivalent paths through the network will be d^{s-1}, (where $s = \log_d N$). Beneš networks can be constructed to any size that is an integer power of 2 but if the size is not an integer power of the degree of the switching element the network will require switching elements of two different degrees. Refer back to fig. 4.9 which describes the construction of a general square three-stage network. The Beneš condition requires that $m = n$. Hence, for example, to construct a 16×16 Beneš network $n = m = 2$ and $r = 8$. If $n = m < r$ a Beneš network which requires the least hardware is constructed. Consider, however, the network in which $n = m = 8$ and $r = 2$ which also gives a 16×16 network that satisfies the Beneš condition. It may require more hardware but it is of interest because it can be formed by sub-equipping a Beneš network that is an integer power of the degree of the switching element. Thus all switching elements are of the same degree, idle switching element ports in the central stage of the network are disabled, and the first and final stages of the network are equipped with only N/d switching elements. This structure will be referred to as the sub-equipped Beneš network. A 32×32 sub-equipped Beneš network of 8×8 switching elements is illustrated in fig. 5.6.

To reduce the number of devices required in the sub-equipped Beneš network of fig. 5.6, the central stage might be formed from 8×8 switching elements, each configured as two 4×4 devices as a selectable option within the standard device. A 16×16 sub-equipped Beneš structure would require switching elements in the central stage configured as four 2×2 devices. A network that is very similar to the sub-equipped Beneš may be formed by reflecting the modified delta network, illustrated in fig. 5.4, about its central stage. The central switching stage of this network, however, cannot be formed easily from a single standard part for all sizes of network and it cannot offer a significantly greater performance than that of the sub-equipped

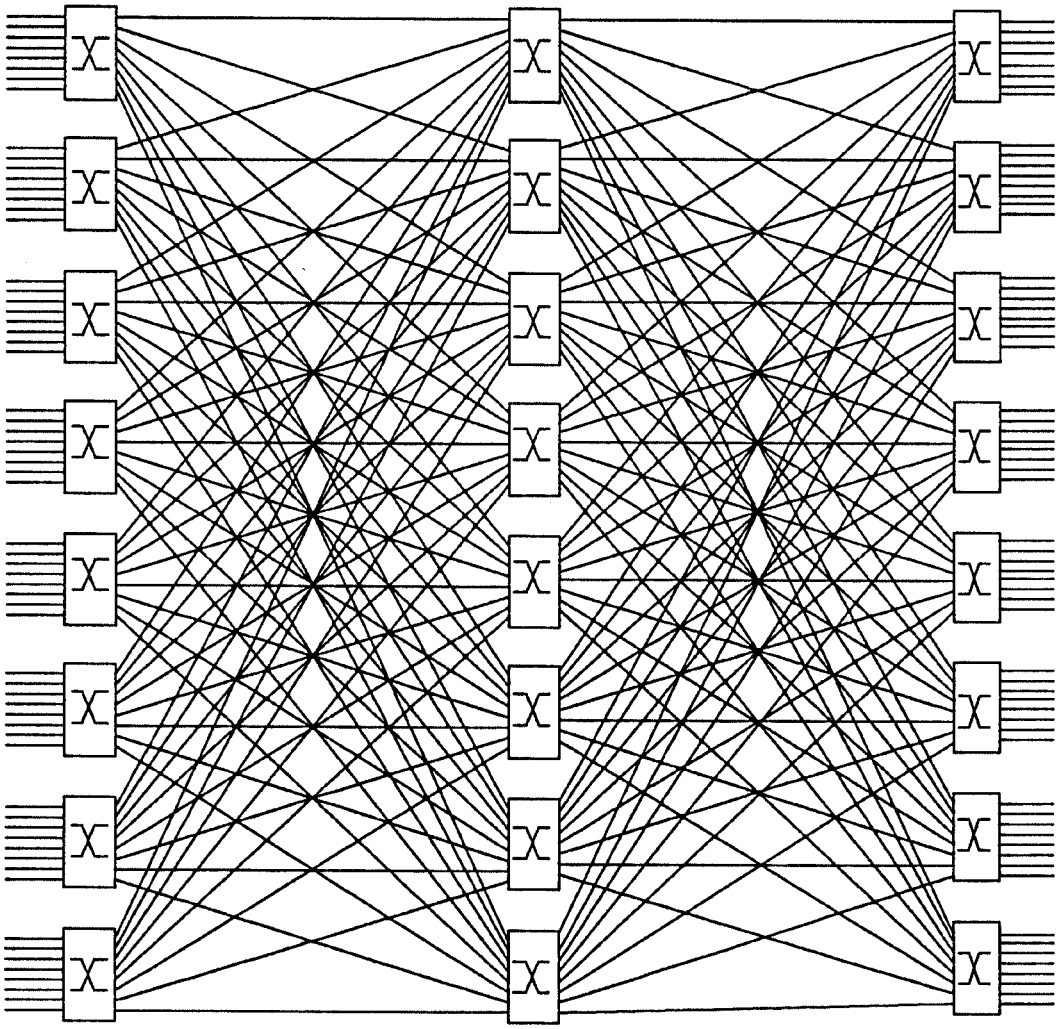

Figure 5.5: A 64×64 Beneš network of 8×8 switching elements.

Beneš structure.

The Beneš network introduces a large number of equivalent paths into the switch fabric. Once again, as there is no buffering within the switch fabric the path for each incident packet may be selected independently without fear of inducing out-of-sequence errors between packets belonging to the same virtual circuit. Three possible algorithms for selecting a path through the network have been investigated: searching, flooding and random. The searching and flooding algorithms operate in the same manner as for the delta network save that there are many more paths to search or flood. In the random algorithm the distribution stages of the switch are implemented with switching elements that select any free path to the succeeding stage at random.

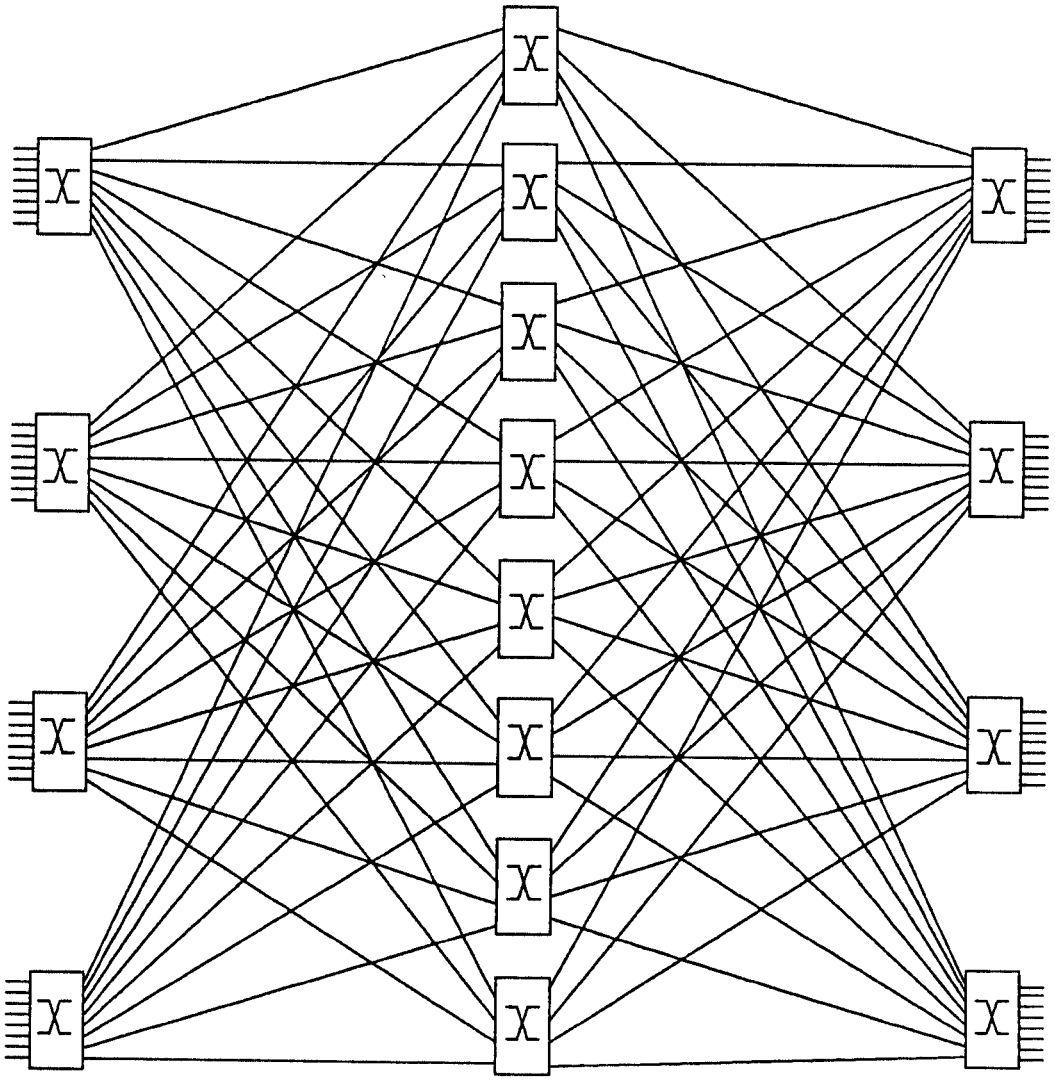

Figure 5.6: A 32×32 sub-equipped Beneš network of 8×8 switching elements.

5.5 The Multi-Plane Switch Structure

It is common practice in the design of a telecommunications switch to duplicate or even replicate the switch fabric and control hardware for reliability and ease of maintenance. If this is achieved in a load sharing manner the performance of the switch is also enhanced. The general structure of a two-plane switch is shown in fig. 5.7 and may be extended to form a multi-plane switch of any arbitrary number of planes. It consists of two identical switch planes, each switch plane being a complete delta network with or without a distribution fabric. The two switch planes are connected in parallel to form a load sharing arrangement [85, 82].

Once again multiple paths are being introduced into the switch fabric and either a

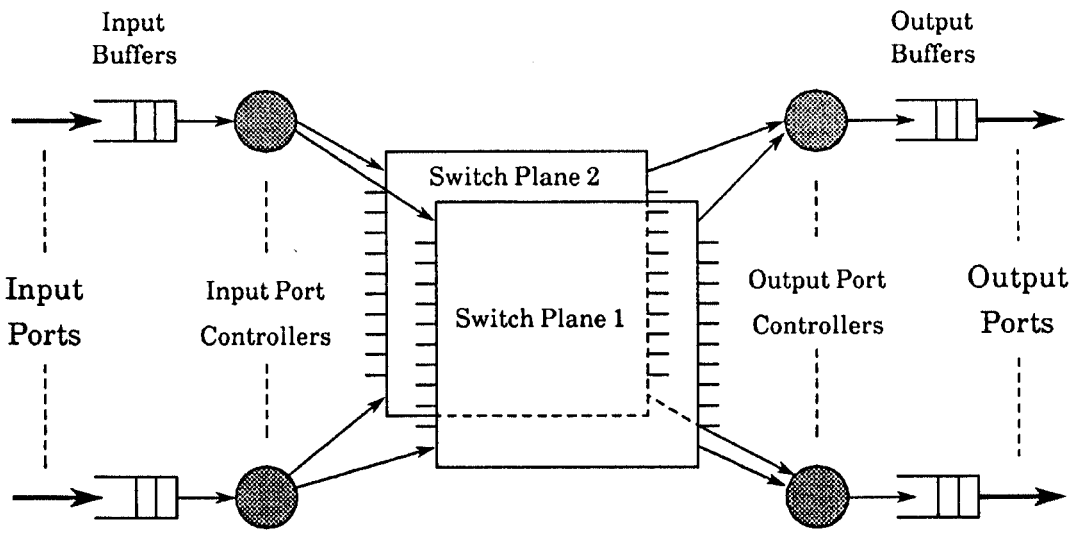

Figure 5.7: A two-plane switch structure.

searching or a flooding algorithm may be used to select a path across one of the switch planes. In the searching algorithm each plane is attempted in sequence until a path is established. For the flooding algorithm, the input port controller transmits a copy of the packet over each switch plane simultaneously and the output port controller selects one of the successful copies. All other copies will fail and will be removed from the switch fabric after the transmission of only a few bits.

A simple implementation of input port controller will only be capable of handling a single packet at once. Improved performance may be obtained if the input port controller is capable of transmitting multiple packets across separate switch planes simultaneously, at the expense of increased hardware complexity. The same is true for the output port controller. A simple implementation will only be able to handle a single packet at a time and thus must reject set-up attempts arriving across the free plane while it is busy serving an existing packet. A more complex output port controller will be capable of handling multiple packets arriving at the same time and buffering them in a first in first out manner in the output buffer. In this manner a measure of output buffering may be provided to increase the performance of the switch at the expense of a more complex output port controller.

5.6 Summary

For reasons of hardware complexity and flexibility of implementation, an input buffered switch design with a non-buffered switch fabric has been selected. For the same reasons the investigation of a blocking switch fabric, the delta network, has been proposed. To improve performance and to remove the sensitivity of the switch to the destination distribution of the incident traffic a switch fabric based upon a Beneš

topology has been suggested. The majority of the existing work has concentrated on the use of 2×2 switching elements. Switching elements of higher degree will offer enhanced performance and reduce the number of interconnections required within the switch fabric which is a major factor limiting the maximum size of switch that may be implemented. Previous work suggests that non-buffered switching elements of up to degree 16 may be suitable for implementation in gate array technology.

To enhance the performance of the basic design, input queue by-pass and the use of multiple switch planes in parallel have been proposed. Output buffering across the multiple switch planes will further increase the performance as will the ability to transmit packets from the same input port across multiple switch planes simultaneously.

Chapter 6

Switch Fabric Performance

Three classes of switch fabric have been proposed, each with a range of possible implementation parameters, and these require investigation in order to select a preferred switch design. This switch design will then be investigated in greater detail in the following chapter.

The simplest way to quantify the performance of a particular switch implementation is to measure the normalised average throughput of the switch when saturated with traffic with a uniform random distribution of packet destinations. This is called the throughput at saturation, or sometimes the maximum throughput, of the switch and it gives a useful measure of the capacity of the switch by which different switch designs may be compared. Another useful measure is the mean packet delay, from entry to exit of the switch, for identical traffic sources on each of the switch ports. A simulation model has been developed to compare the throughput at saturation and mean delay performance of the different switch fabrics according to the various implementation parameters summarised in table 6.1.

6.1 Traffic Models

Two traffic models were developed for the comparison of the various switch designs: a saturation model for the evaluation of throughput at saturation and a slotted traffic source for the delay comparison. In the saturation model every input port of the switch is saturated with incoming traffic so that a new packet is always available at every input port on completion of transmission of the packet in service. For switches that do not employ input queue by-pass it was not necessary to model the input queues but merely to supply each input port with a fresh packet whenever it completed a packet transmission. For switches with input queue by-pass, an input queue was modelled on every switch port which was always full. For switches up to size 512×512 the queue was 100 packets long while for larger switches a queue size of 10 packets was considered sufficient. All packets were of the same length and followed a uniform random destination distribution while all output ports acted as a perfect sink.

Parameter	*Range*
Switch Fabric Size	2×2 to 4096×4096
Interconnection Networks	Crossbar Delta Beneš
Degree of Switching Element	2, 4, 8, 16
Multiple Path Algorithms	Searching Flooding Random
Multiple Switch Planes	1 to 4
Port Controllers	Regular Input Queue By-Pass Double Buffered Output De-Luxe

Table 6.1: Switch fabric design parameters.

While the throughput at saturation gives information about the maximum capacity of the switch design, no input buffered switch can effectively be operated at this capacity as the input queues would become permanently full leading to very high delay and high packet loss. To compare the mean delay through different switch structures, traffic sources which produce a load below that of the throughput at saturation must be used on each input port. One such traffic source that has been analysed in the literature is the slotted traffic source, or more correctly referred to as the Bernouilli arrival process [76]. In this model each input port receives a contiguous stream of timeslots, each timeslot being the length of a single packet. Each timeslot may be either empty or filled with a single packet and the load offered by the traffic source is the uniform random probability of any timeslot containing a packet. (A slotted source with a load in excess of the throughput at saturation of the switch fabric becomes equivalent to a saturated source.) All packets are given a random destination distribution and all sources are set to the same value of applied load for each measurement of mean delay.

6.2 The Simulation Model

In order to reduce the amount of computer time required by the simulation model to acceptable proportions one major simplification was made. Each packet set-up attempt was modelled as an instantaneous event. Thus in the model each packet set-up attempt is taken in turn and a complete path traced through the interconnection network. If all nodes in the path are originally free then the set-up attempt is deemed successful and all nodes in the path marked busy for the duration of the transmission of the packet. For flooding algorithms all of the multiple paths to the destination are investigated until a free path is located. If no free path is available the set-up

attempt is considered blocked and a new set-up attempt scheduled to occur after the retry delay.

In reality a packet will set up on a stage by stage basis, thus a packet which fails set-up could itself cause blocking during its unsuccessful set-up attempt. The effect of this simplification is to over-estimate the throughput at saturation. A more detailed simulation model which does consider the set-up of packets on a stage by stage basis has also been investigated to evaluate the effect of this simplification. It will be shown that for two-plane delta networks constructed from switching elements of degree 8 or 16 the error introduced by this simplification in the evaluation of the throughput at saturation is in general less than about 2%.

Other simplifications include the modelling of the release of the path on completion of packet transmission as instantaneous. In a real implementation there may be a fixed delay or a stage by stage release mechanism. Also it is assumed in the saturation model that a new packet set-up attempt immediately follows transmission of the previous packet whereas a small recovery delay might actually be required. To assume otherwise, however, would imply modelling the characteristics of one particular implementation which is not the intention of a general comparison of switch fabrics.

The simulation model is a discrete time, event driven simulator. For the measurements of throughput at saturation and delay for slotted traffic a resolution on the time axis of 1/100 of a packet length was found to be sufficient. (Time within the simulation model is generally normalised to the packet length, the packet length being the emission delay of a packet at the speed of the switch fabric.) In effect, no limit was placed upon the number of set-up attempts per packet. The simulation was in general initialised with random time relationships between all packets and run for a time of 200 packet lengths to acquire stability before measurements commenced. The majority of simulations were run for a total of 200,000 packets which, for the measurements of throughput at saturation, yielded a standard deviation of about 0.4% of the mean for the smaller network sizes to about 0.2% for the larger networks (64×64 and greater). The throughput per port at saturation is normalised to represent the proportion of time on average that an output port carries valid traffic. Due to the simplification of the model this figure will include the routing bits of the tag, the label, and any other line code, framing or other overhead bits that a packet must carry. The useful port bandwidth is thus slightly less than that indicated by the measurements of throughput at saturation but the figure proves useful for the purpose of the comparison of switch fabrics.

6.3 The Crossbar Switch Fabric

Throughput at Saturation

The crossbar switch fabric is non-blocking thus it offers the ideal performance against which other interconnection networks may be compared. In the input buffered cross-

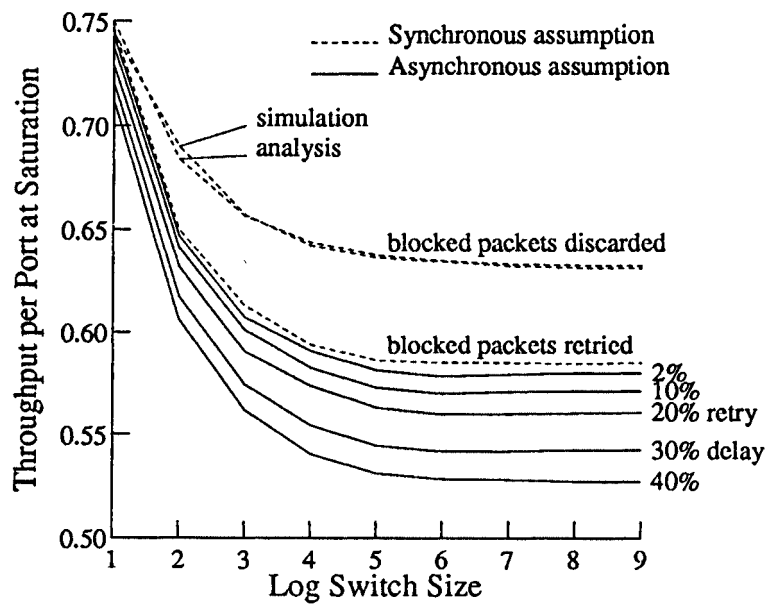

Figure 6.1: Throughput at saturation for the crossbar switch fabric.

bar switch, blocking, (or to be more precise contention,) proceeds solely from the probability of multiple sources attempting to transmit to the same destination at the same time. The throughput at saturation results of crossbar switch fabrics up to size 512×512 are presented in fig. 6.1 under various assumptions. The switch size ($N \times N$) is expressed as $\log_2 N$ and the curves are discrete, points being connected purely for visual convenience. In the synchronous assumption all packets are presented to the switch synchronously and in phase such that contention for all output ports occurs instantaneously and those packets that are successful are then transmitted across the switch fabric. Those packets that are blocked may either be discarded or wait exactly one packet length to be resubmitted in the next timeslot. In the asynchronous assumption there are random time relationships between all packets. If a packet is blocked it waits for the duration of one retry delay and is then resubmitted independently of all other packets. The retry delay is expressed as a percentage of the packet length.

The analysis of [126] gives an expression for the throughput at saturation of a crossbar switch fabric under the assumptions of synchronous packet arrival and blocked packets discarded. The expression is $1-(1-1/N)^N$ and is compared with the simulator output in the upper curves of fig. 6.1. It has an asymptote of $(1-e^{-1}) = 0.632$ for large N and the simulation may be seen to agree very closely with the analytical result. This is hardly surprising as the crossbar switch fabric is a very simple network to simulate. If packets are resubmitted rather than discarded, the synchronous crossbar switch fabric becomes much harder to analyse but both [76] and [71] give the result of $(2-\sqrt{2}) = 0.586$ for the asymptote of the throughput at saturation for large N. Again this agrees closely with the simulation results.

The throughput at saturation for asynchronous operation of a crossbar switch fabric at various values of retry delay is also given in fig. 6.1. Asynchronous operation with a retry delay of zero is equivalent to synchronous operation as the contention for the output ports occurs immediately they become free from the previous packet. In asynchronous operation the throughput at saturation is reduced as the retry delay increases. This is because the greater the retry delay, the greater the probability that an output port spends some idle time after serving one packet before a contending input port resubmits a packet set-up attempt.

For the synchronous crossbar switch fabric with blocked packets retried the throughput at saturation for traffic with a uniform random destination distribution represents the probability (p_a) that any packet will be successful on any set-up attempt. Hence the operation of any input port of the switch may be modelled as a geometric server in which the probability that a packet will require j set-up attempts is $p_a(1-p_a)^{j-1}$. The mean delay across the switch fabric is thus:

$$\sum_{j=1}^{\infty} j p_a (1-p_a)^{j-1} = 1/p_a$$

The delay across the switch fabric was measured with the simulation model and was shown to equal the inverse of the throughput at saturation for both synchronous and asynchronous models.

Mean Delay for Slotted Traffic

Analytical results are also available of the mean delay for slotted traffic for both the input buffered crossbar switch fabric and the output buffered switch. In [76] the mean delay of the input buffered crossbar switch fabric is derived numerically while that of the output buffered switch is given as:

$$\frac{(N-1)}{N} . \frac{p}{2(1-p)} + 1$$

where N is the size of the switch and p is the traffic load (which is the probability that any input timeslot contains a packet). An approximate expression is derived in [71] for the mean delay of the input buffered crossbar switch of large N:

$$\frac{(2-p)(1-p)}{(2-\sqrt{2}-p)(2+\sqrt{2}-p)}$$

These analytical results are plotted in fig. 6.2, for switches of large N, together with the results of the simulation model. The curves from the simulation model and the analysis of [76] are virtually coincident while the analysis of [71] gives an approximation of reasonable accuracy.

The effect of introducing input queue by-pass, and output buffering across a two-plane crossbar switch fabric, is shown in fig. 6.3. (A two-plane design with output buffering is termed 'double buffered' and a double buffered structure with input queue

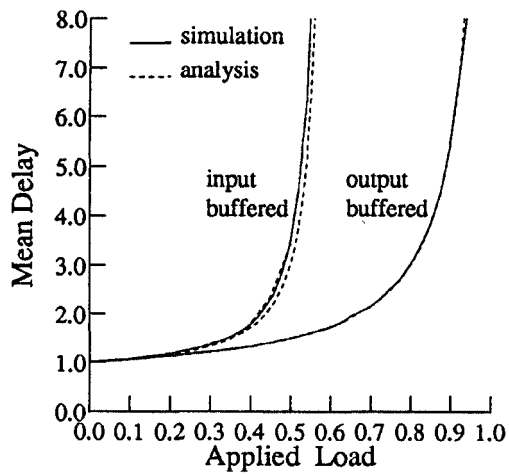

Figure 6.2: Analysis and simulation of mean delay performance for slotted traffic.

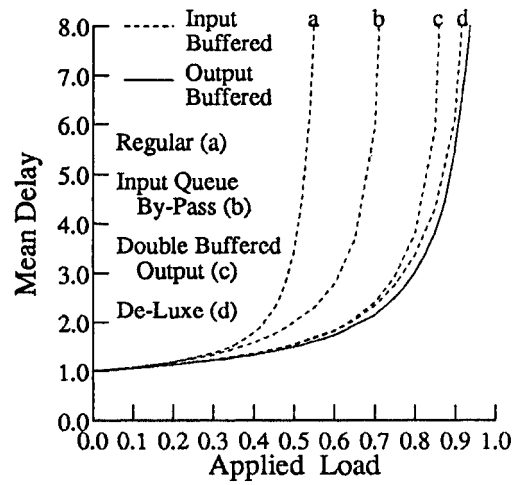

Figure 6.3: Mean delay performance of crossbar switch structures for slotted traffic.

by-pass is referred to as 'the de-luxe model' for convenience. A switch structure with neither queue by-pass nor output buffering is termed 'regular' or occasionally 'pure input buffered'.) These curves apply to the input buffered crossbar switch fabric of large N and are compared to the delay performance of the output buffered switch. For the regular model, each output port was capable of handling only a single packet arriving at any time whereas the double buffered model could handle two. The length of both input and output queues was effectively unlimited. A retry delay of 10% of the packet length between unsuccessful set-up attempts was assumed in the input queue by-pass model. The double buffered curve gives the performance that a two-plane input buffered Batcher-banyan switch fabric might achieve with output buffering across the two planes. It assumes that each input port is only capable of handling at most one packet at a time. (A synchronous Batcher-banyan switch fabric is effectively unable to make use of input queue by-pass.) It is clear that the performance of the de-luxe model approaches very closely that of the output buffered switch but these results assume the use of a non-blocking switch fabric. The performance of the blocking and rearrangeable non-blocking structures will now be presented which are more easily capable of implementation. Finally, a table of the throughput at saturation results of the crossbar switch fabric is given in the appendix for the four classes of switch design with both synchronous and asynchronous operation.

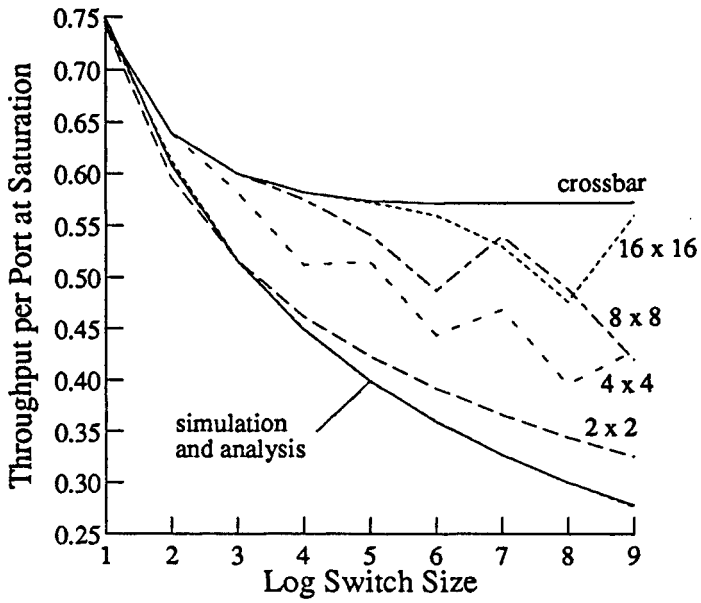

Figure 6.4: Throughput at saturation for single plane input buffered flooding delta networks.

6.4 The Delta Network

Single Plane Delta Networks

The throughput at saturation performance of single plane input buffered delta networks is presented in fig. 6.4 constructed from switching elements of degree 2,4,8 and 16. A flooding algorithm has been used with asynchronous operation, blocked packets resubmitted and a retry delay of 10% of the packet length. The corresponding curve for the equivalent crossbar switch is included for comparison. The perturbations in the curves of degree greater than 2 are due to the presence of multiple paths across the network between the same input and output ports. The minima indicate the pure delta network in which the size of the network is an integer power of the degree of the switching element and only a single unique path connects any input to any output.

For the pure delta network under synchronous operation with blocked packets dropped, analytical results for the throughput at saturation are available from [126] in the form of a recurrence relation:

$$m_i = 1 - \left(1 - \frac{m_{i-1}}{d}\right)^d$$

The throughput at saturation at the output of stage i is given by m_i where $m_0 = 1$ and the delta network is constructed from switching elements of degree d. The analytical results from the above expression agree very closely with the results from the simulation model for synchronous operation with blocked packets dropped, to within 0.4%. This is well within the 95% confidence interval of the simulation results.

The analytical and simulation results for delta networks of degree 2 are compared in fig. 6.4 where it may be seen that the two curves are virtually co-incident.

The throughput at saturation performance of the delta network under synchronous operation with blocked packets retried is approximately 8% lower than with blocked packets dropped. This reduction in performance is of the same magnitude as for the crossbar network, fig. 6.1. For the case of blocked packets retried, the throughput at saturation of the delta network under synchronous operation is less than that obtained under asynchronous operation whereas for the crossbar switch fabric a slightly greater result was obtained, fig. 6.1. This is due to the additional blocking effect of unsuccessful packet set-up attempts which themselves cause blocking within a multi-stage network. This effect is at a maximum in the synchronous model where all set-up attempts occur together. It is approximately proportional to the number of stages of interconnection within the network. For asynchronous operation the magnitude of the effect depends upon the packet length, the retry delay, the size of the switch fabric and the speed with which a packet set-up attempt is detected and removed from the switch fabric. The selection of the retry delay in an asynchronous switch is thus a compromise between the loss of throughput from an increasing retry delay, illustrated in fig. 6.1, and the increase in blocking resulting from an increased number of unsuccessful packet set-up attempts, for decreasing retry delay, in the delta network. A retry delay of 10% of the packet length has been selected as a reasonable compromise for fast packet switching applications. An investigation of the case in which the retry delay was exponentially distributed about a mean value yielded no difference in performance to that of a constant retry delay.

Multi-Plane Delta Networks

The throughput at saturation of a switch fabric constructed from multiple delta networks in parallel is illustrated in fig. 6.5 in which multiple paths are investigated by (a) a searching algorithm and (b) a flooding algorithm. The networks are constructed from switching elements of degree 8 and 1 to 4 parallel switch planes are shown with a retry delay of 10% of the packet length and are compared to the crossbar switch fabric. Both the input and output port controllers are capable of handling only a single packet at a time (i.e. the regular switch structure). It is evident that the use of two switch planes in parallel yields a useful increase in throughput performance beyond that of a single plane but that the use of more than two switch planes offers little incremental improvement in throughput performance.

A maximum of two switch planes in parallel has thus been selected for the regular switch structure. No more than two switch planes have been investigated for the structure in which output buffering is provided across the switch planes as the complexity of such a structure begins to approach that of an output buffered switch. The performance of the switch may be enhanced with much less hardware than a three plane output buffered structure would require. Two possible techniques are the use of input queue by-pass and input port controllers capable of handling two packets simultaneously, one across each switch plane.

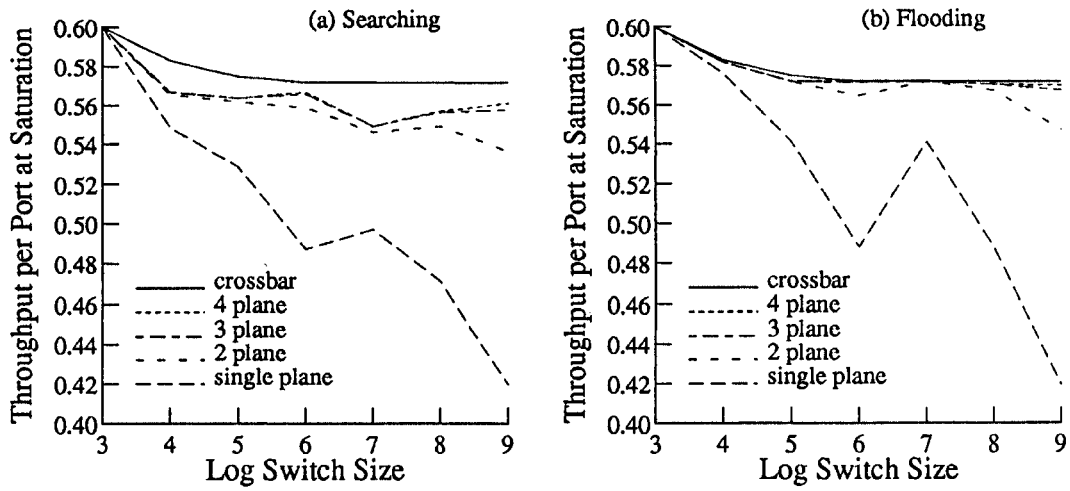

Figure 6.5: Throughput at saturation for multiple delta networks in parallel.

Two basic algorithms exist to select a path through a switch structure that offers multiple paths: flooding and searching. In the flooding algorithm the number of equivalent paths that require investigation changes with the size of the switch fabric thus the switch fabric would require construction with a number of different basic switching elements. A slightly modified flooding algorithm has been investigated that permits the switch fabric to be constructed from identical switching elements. It floods across the parallel planes but searches within each plane and is referred to as the flood-planes algorithm. For both the flood-planes and searching algorithms, improved performance is obtained if a random path is selected for the first set-up attempt, (flood-planes random and search random.) After this, paths are searched in sequence until a free path is located across the switch fabric. The throughput at saturation of the various algorithms is compared in fig. 6.6 for a two-plane regular delta network with switching elements of degree 8 and a retry delay of 10% of the packet length. The flood-planes random algorithm has been selected for further study as it differs from the pure flooding algorithm only marginally and results in a simpler hardware implementation.[1] Also the simulation does not model the interference between failed set-up attempts which is likely to be larger in the pure flooding algorithm than for the flood-planes algorithm.

The throughput at saturation for pure input buffered two-plane delta networks constructed from switching elements of degree 2,4,8 and 16 is shown in fig. 6.7 for a flood-planes algorithm and a retry delay of 10%. Curves for the crossbar switch fabric and the single plane flooding delta network of switching elements of degree 8 are included for comparison.

The effect of input queue by-pass and output buffering on the mean delay performance with slotted traffic is shown in fig. 6.8. It presents the results for a 64×64

[1] Further references to the flood-planes and searching algorithms in the context of delta networks imply the flood-planes random and search random variants.

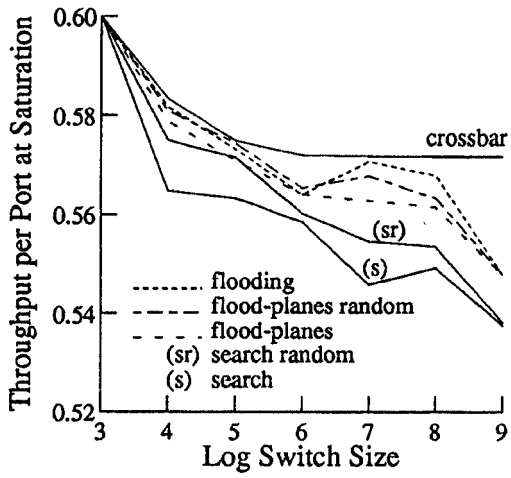

Figure 6.6: Comparison of algorithms to select a free path across the network.

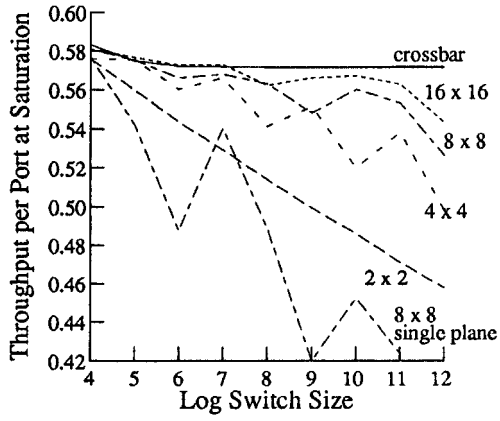

Figure 6.7: Throughput at saturation for two-plane pure input buffered delta networks.

two-plane delta network of switching elements of degree 8 with a flood-planes algorithm. The performance of the single plane crossbar, the two-plane output buffered crossbar, and the output buffered switch are also given for comparison. The curves behave as expected and the de-luxe two-plane delta network, (output buffered with input queue by-pass,) offers a similar performance to the two-plane output buffered crossbar switch fabric. The detailed results of the throughput at saturation performance of both single and two-plane delta networks with input queue by-pass and output buffering are given in the appendix. Delta networks constructed from switching elements of degree 2, 4, 8 and 16 are considered with a flood-planes algorithm and a retry delay of 10% of the packet length.

A More Accurate Model

The simulation results presented so far, for the delta network under asynchronous operation, have ignored the element of blocking caused by unsuccessful packet set-up attempts. A more accurate simulation model was developed to measure the error introduced into the simulation results by this simplification. In the accurate model, packets are set up on a stage by stage basis such that all set-up attempts, partial, successful and unsuccessful, contribute to the blocking within the network. A packet length of 256 bits was selected with an additional routing tag of length $\log_2 N$ bits. It was assumed that the partial path of a blocked set-up attempt would be cleared after a delay of two bit times. The performance of a searching algorithm was modelled as the multiple simultaneous set-up attempts of a flooding algorithm would have consumed an excessive amount of computer time in the simulation.

The percentage error of the simple model expressed with respect to the results of the more accurate model are given in table 6.2 for switching elements of degree 8 and

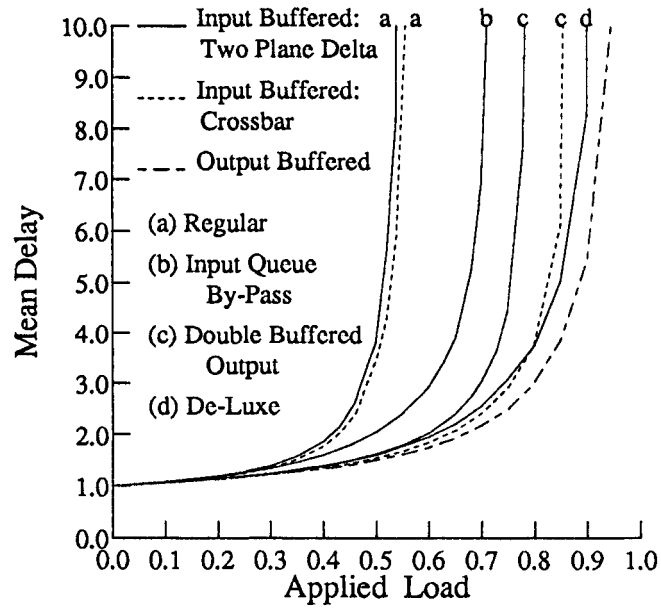

Figure 6.8: Comparison of mean delay performance for slotted traffic of various switch structures of size 64×64.

Size	Single Plane	Two Plane	Double Buffered
8	−0.83	−0.85	−0.34
16	+2.44	+1.38	+0.68
32	+2.32	+0.88	+0.29
64	+2.45	+0.27	−0.20
128	+3.21	+2.61	+0.84
256	+3.21	+1.94	+0.34
512	+4.24	+1.15	−0.01

Table 6.2: Percentage error in throughput at saturation of simple model for delta networks with switching elements of degree 8.

a retry delay of 10% of the packet length. The results for the single plane, two-plane (regular) and two-plane output buffered (double buffered) models are given and the results for the same structures with input queue by-pass are similar. The percentage error in the simple model for single plane delta networks with switching elements of degree 2, 4, 8 and 16 is given in table 6.3.

The simple model slightly underestimates the throughput at saturation of a single stage switch fabric, i.e. a switch fabric consisting of a single crossbar switching element. This is because the simple model assumes that an output port is busy for the entire duration of the packet plus the route bits whereas in the accurate model an output port is not marked busy until the route bits have been consumed. The simple model of the single plane switch is the least accurate as in this structure the density

Size	Degree 2	Degree 4	Degree 8	Degree 16
2	−0.12	–	–	–
4	−0.05	−0.22	–	–
8	+1.34	+1.37	−0.83	–
16	+2.79	+1.26	+2.44	−0.97
32	+8.41	+3.29	+2.32	+3.13
64	+6.75	+4.21	+2.45	+2.79
128	+7.28	+4.03	+3.21	+2.25
256	+7.46	+5.25	+3.21	+2.37
512	+8.63	+4.92	+4.24	+3.91

Table 6.3: Percentage error in throughput at saturation of simple model for single plane delta networks.

of unsuccessful packet set-up attempts is greatest. Also the error introduced by the simplified model is greater for delta networks constructed from switching elements of lower degree. In general, for two-plane delta networks constructed from switching elements of degree 8 or more, the error in the simple model for the estimation of throughput at saturation is no greater than 2%.

The comparison between the simple and accurate models has been measured for the searching algorithm. Thus the measurement of the throughput at saturation under a searching algorithm provides an accurate lower bound on the performance that may be expected from delta networks. The measurements taken with a flood-planes algorithm give an upper bound to the expected throughput performance at saturation. The throughput at saturation measurements from both flood-planes and searching algorithms are presented in the appendix for all of the classes of switch structure discussed, with switching elements of degree 2, 4, 8 and 16.

6.5 The Beneš Network

An investigation of the throughput at saturation of the Beneš and the sub-equipped Beneš structures reveals that the searching algorithm yields a much poorer performance than the flooding and random algorithms and is even slightly inferior to the performance of the equivalent single plane regular delta network. This result is perhaps to be expected as the Beneš structures offer many more paths to be searched sequentially than does the delta network. Also the increased number of switch stages increases the possibility of blocking within the switch fabric at high loads. The random algorithm offers a performance which is better than that of the equivalent single plane regular delta network because it always selects a path through the distribution stages of the switch fabric which is known to be free. As might be expected the flooding algorithm offers the best throughput performance which is very close to that of the equivalent crossbar network. The simulation model, however, does not consider the effect of the interference between packet set-up attempts caused by the multiple

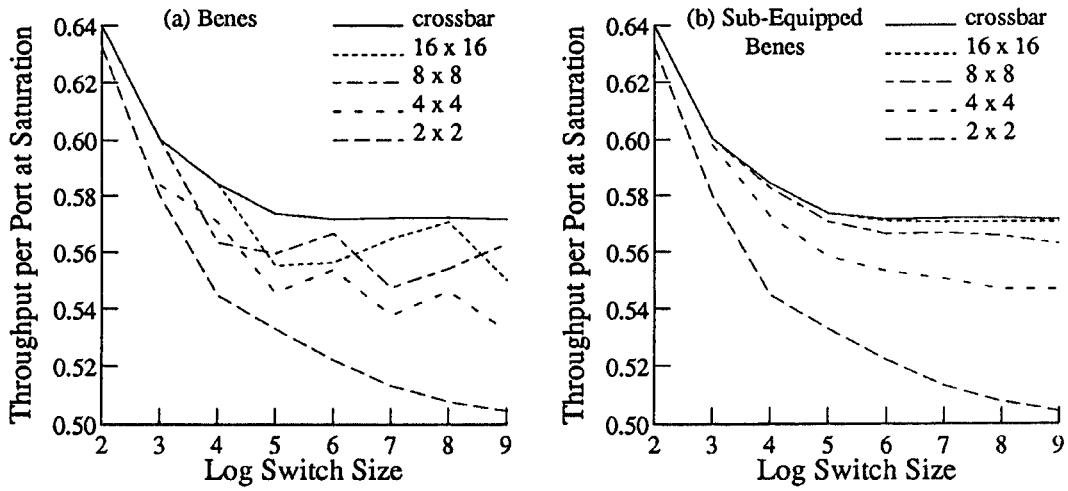

Figure 6.9: Throughput at saturation for flooding Beneš structures.

copies of each packet generated by the flooding algorithm. The results for the flooding algorithm must therefore be considered as an upper bound on the performance of Beneš structures while those of the random algorithm may be considered as a lower bound. The accuracy of the results for the random algorithm is likely to be similar to, if not better than that of the single plane delta network. For applications that require a short packet length the random path selection algorithm is therefore preferred. For longer packet lengths, however, the flooding algorithm may offer improved performance as the relative interference between multiple packet set-up attempts is reduced by the increase in packet length.

Two versions of the flooding algorithm were compared. One version selected at random between all of the free paths to the destination whereas the other was biased to select a path through the uppermost central stages of the switch fabric thus attempting to pack the successful set-up attempts as closely as possible. No difference in throughput performance at saturation was observed between these two variations.

The throughput at saturation for both (a) Beneš and (b) sub-equipped Beneš structures with a flooding algorithm and a retry delay of 10% of the packet length are given in fig. 6.9. The perturbations in the Beneš curve are due to the degree of the switching elements in the first and final stages which may be lower than that of the switching elements in the central stages. The throughput at saturation is a maximum when all switching elements are of the same degree d which occurs when the size of the switch fabric is an integer power of d.

The detailed results of the throughput at saturation of the sub-equipped Beneš structure for both random and flooding algorithms, with and without input queue by-pass are presented in the appendix. For networks constructed from switching elements of degree 8 and 16 the performance of the flooding sub-equipped Beneš structure is very close to that of the equivalent crossbar network for both regular and input queue by-pass switch designs. Under the random algorithm a Beneš network offers a

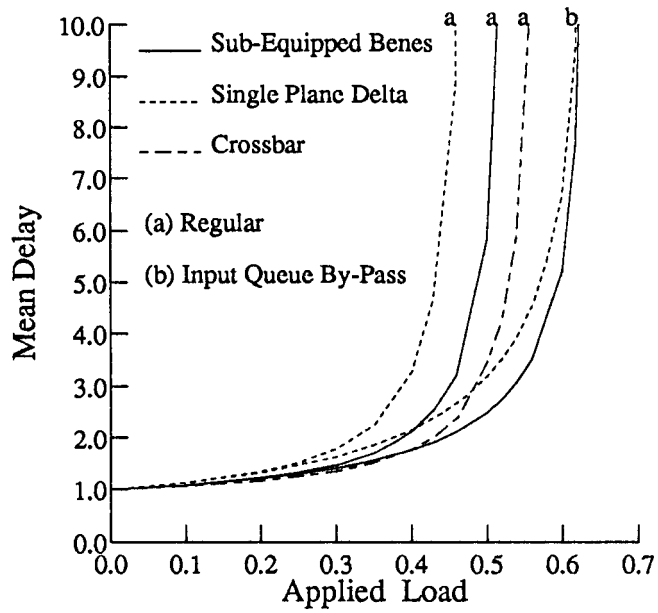

Figure 6.10: Comparison of mean delay performance for slotted traffic of 64×64 sub-equipped Beneš networks against other structures.

performance between that of the single and two-plane regular delta networks whereas with input queue by-pass the Beneš structure offers a performance very similar to that of the single plane delta network with input queue by-pass. For switch structures that feature output buffering across two Beneš switch planes the performance of the crossbar network may be considered as an upper bound and that of the equivalent delta network as a lower bound.

The mean delay performance for slotted traffic of a 64×64 sub-equipped Beneš network of switching elements of degree 8 with a random algorithm is given in fig. 6.10. Both regular and input queue by-pass switch structures are compared to the equivalent single plane delta and crossbar networks. The delay performance of the single plane delta network with input queue by-pass may clearly be seen to approach that of the Beneš structure as the load on the switch increases. This suggests that the improvement in the throughput performance of the Beneš structure which is gained by spatially distributing the incident traffic across the switch fabric is matched by the distribution in the time domain of the input queue by-pass algorithm as the average queue length in the input buffers grows with the load.

Finally, it is unlikely that a Beneš switch fabric will be selected as opposed to a delta network purely on the grounds of its throughput performance at saturation. The performance difference between the two structures is not particularly significant. The Beneš structure is of interest because of its greatly reduced sensitivity to the destination distribution of the incident traffic when compared to the delta network.

6.6 Summary

A simulation model has been developed to compare the throughput at saturation performance of various switch structures according to a number of implementation parameters. Where analytical results have been available, comparison with the results of the simulation model has been shown to yield very close agreement to well within the 95% confidence interval of the simulation results. The performance of a pure input buffered switch with a crossbar switch fabric is shown to be about 58% of that of an output buffered switch. However, the use of input queue by-pass and a two-plane switch fabric with output buffering enhances the performance to become very close to that of the output buffered switch.

For delta networks, the use of a two-plane switch fabric is recommended on the grounds of increased throughput, increased reliability and ease of maintenance. The use of more than two switch planes in parallel is not justified from the point of view of increased performance. The use of switching elements of degree 8 or 16 is preferred against those of degree 2 or 4 because of improved throughput performance and reduced interconnections within the switch fabric. For delta networks the searching algorithm offers a performance only slightly lower than that of a flooding algorithm and a hybrid algorithm which searches within each switch plane but floods across the planes is recommended for its ease of implementation. A two-plane regular delta network offers a throughput performance only slightly inferior to that of a crossbar switch fabric. The introduction of input queue by-pass together with output buffering yields a performance comparable to the two plane crossbar switch fabric with output buffering, and only slightly lower than that of the output buffered switch.

In general, a Beneš switch fabric is unlikely to be favoured above a delta network purely on the grounds of its throughput performance. It is of interest because it reduces the sensitivity of the switch to the destination distribution of the incident traffic. The performance of the Beneš switch fabric lies between that of the equivalent delta and crossbar networks. For applications that require a short packet length the random path selection algorithm is recommended whereas a flooding algorithm may yield improved performance for longer packet lengths.

Chapter 7

Performance for Multi-Service Traffic

The measurements presented in the previous chapter concentrated on comparing the performance of the various possible switch fabrics. We now consider how to integrate multiple services (voice, video, image, text, data, etc.) onto the switch structure. The models of the source traffic selected, and also to some extent the performance measurements taken, are necessarily simple yet they offer a first order guide to the performance of the switch under a mixed traffic load. The voice service multiplexed with data traffic is selected for a more detailed investigation as it presents a well characterised traffic source of practical interest. The results of the detailed investigation of the voice traffic model are shown to agree well with the simple model of mixed traffic but further study is required to characterise the performance of a fast packet switch for applications within the general purpose broadband ISDN.

7.1 Multi-Service Traffic Requirements

It may be argued that all communications services may be classified into two fundamental categories according to the delay requirement that they present to the network, and for lack of better terminology they will be referred to as reserved and unreserved services. A reserved service exacts an inflexible, low delay and low variance of delay requirement, whereas unreserved services are much more flexible in the range of delay that can be tolerated. Due to the delay requirement, an incoming reserved service call will only be accepted if sufficient switch bandwidth is available. A measure of the switch bandwidth allocated to reserved service calls is kept and once it reaches a maximum, determined by the delay requirement, further reserved service calls are refused. The allocation of switch bandwidth to unreserved service calls may be much less stringent.

The majority of reserved services derive from information based upon a physical property that changes rapidly with time, e.g. voice and video, and often contain a

high degree of redundancy, thus permitting an appreciable packet loss before any noticeable deterioration in quality is perceived. There are some reserved services, however, that are highly sensitive to error, e.g. process control, in which the delay constraint proceeds from the requirement for a high priority service, yet such services are generally of low bandwidth. Unreserved services include the bulk of data transfer, interactive and transaction services at various priorities.

The delay constraint is not the only difference between these two basic service classifications. A reserved service requires a guaranteed bandwidth and delay performance throughout the entire duration of the connection, else the connection request must be refused. An unreserved service expects the bandwidth and delay associated with a connection to vary according to the traffic load on the network.

Three approaches to the support of these two fundamental classes of service across a fast packet switch have been proposed. The first makes no differentiation between the classes of traffic at the lowest level within the switch fabric. To guarantee the delay performance for reserved service traffic it assumes that at the access point to the network all connections are constrained to conform to the measure of network resources that each was allocated when the connection was set up. Measures such as the average and peak bandwidth and the burstiness of the traffic have been suggested [5]. This method does not share switch capacity between the various classes of traffic very efficiently and tends to limit the peak bandwidth available to bursty services.

A second approach allocates levels of priority to the different classes of traffic at the lowest level within the switch fabric. Thus high priority traffic achieves the lowest delay and variance of delay whilst the lower priority traffic shares what switch capacity remains after the higher priority traffic has been serviced. This ensures that the delay performance of the higher priority traffic is not greatly influenced by the amount of lower priority traffic within the switch. High priority (reserved service) traffic must still be allocated according to some measure of the switch capacity required but low priority traffic is not constrained to the same degree and may share the remaining available bandwidth. The priority of packets within the switch need not necessarily be allocated purely according to the class of traffic they carry. Traffic originating from some switch ports may require a higher priority than traffic from other ports. Also packets within the switch might be time stamped to give packets that have been delayed the longest in the input queues a higher priority. This would tend to reduce the variance of delay across the switch.

A third approach to the support of multi-service traffic across a fast packet switch argues that all services must be guaranteed a minimum performance at all times. The free capacity remaining within the switch fabric is measured before any connection request is granted to ensure that sufficient capacity remains to support the minimum requirements of the connection. A protocol is also implemented to ensure that the available bandwidth is distributed fairly between all switch ports and also between all classes of traffic within each switch port [54]. This approach may be the most efficient but it is difficult to apply to a large multi-path switch. For the performance measures in this chapter two classes of traffic have been introduced, reserved and unreserved,

in which reserved traffic has a higher priority than unreserved traffic.

7.2 Extensions to the Switch

In order to support the two fundamental services, reserved service traffic must be given priority at all input and output ports. At the input ports, the single input queue at every port of fig. 5.2 is replaced by two queues, one for reserved service packets and one for unreserved service packets. A priority field is also added to the tag to distinguish the two classes of packet. The input port controller is modified so as to transmit unreserved service packets only when the reserved service packet queue is empty, and to postpone repeated set-up attempts of an unsuccessful unreserved service packet on the arrival of a reserved service packet. The transmission of a successful unreserved service packet is not interrupted by the arrival of a reserved service packet.

Reserved service priority must also be ensured at the output ports of the switch fabric. Two mechanisms have been investigated to implement reserved service traffic priority at the output ports and both of them are capable of a simple hardware implementation within the output port controller of the switch. The first mechanism applies to a two-plane switch structure with a simple output port controller that is only capable of handling a single packet at any one time. If a second packet arrives across the free plane while the output port controller is already busy with a packet then the set-up attempt must be rejected but the priority of the packet may be read before it is rejected. If the rejected packet is of high priority then following the completion of the packet in service the output port will only accept a high priority packet. In a two-plane switch fabric with output buffering across the switch planes an alternative algorithm is adopted. In this situation the output port controller will not accept more than one low priority packet at any one time.

7.3 Traffic Models

Two models of unreserved service traffic were used, saturation and Poisson. In the saturation model, unreserved service traffic was generated to keep each input port continuously busy while in the Poisson model, unreserved service packets were generated according to a Poisson arrival process. Both models generated traffic with a uniform random destination distribution. Three models of reserved service traffic were investigated: Poisson, talkspurt voice and TDM voice. In the Poisson model, reserved service packets were generated according to a Poisson arrival process with a uniform random distribution of packet destinations. In the talkspurt voice case, a superposition of individual voice sources was modelled, on every input port of the switch, in which the on-off characteristics of speech were used for bandwidth compression (i.e. packet voice with silence detection). Each voice source was assumed to exhibit two states, active and silent, representing the talkspurts and pauses present in conversational speech [17]. In the active state each voice source generated packets at

a regular rate representing 32 kbits/sec voice coding, 256 bit packets with a further 32 bits overhead, and a 20 MHz system clock. No packets were generated in the silent state. The two states were modelled by an exponential distribution with means of 1.2 and 1.8 seconds respectively [33], and each voice source transmitted packets to a single destination which was selected at random during initialisation. The TDM voice model was simply a talkspurt model with silent periods of zero duration to represent packet voice without silence detection. A random phase relationship was assumed between all voice sources.

The measurement of delay selected for the performance of the reserved service was that of the 99th percentile of the delay distribution [7]. It was assumed that packet voice traffic may withstand a 1% random packet loss, for small packet sizes [58, 60], without perceptible loss of quality. Hence, the measure of maximum delay was the delay within which 99% of all reserved service packets arrived at their destination. One consequence was that the accuracy of the maximum delay measurements was much lower than that of throughput as the tail of the delay distribution was being examined.

Delay was normalised to the packet length and all measurements were taken with a retry delay of 10% of the packet length. Where appropriate, delay results were smoothed with a least squares polynomial regression followed by interpolation with a cubic spline. Applied load and throughput per port were also normalised and reflect the average utilisation of input and output ports.

7.4 Poisson Traffic

For a 64×64 two-plane pure input buffered delta network with switching elements of degree 8, fig. 7.1 gives the throughput result with a Poisson reserved service traffic source and a saturated unreserved service traffic source on each of the switch input ports. As the reserved service traffic load is increased, so the maximum unreserved service traffic load that the switch is able to sustain falls, so as to maintain the load on the switch reasonably constant at saturation. The reserved service throughput response in the absence of any unreserved service traffic is identical to that in the presence of unreserved service sources. Fig. 7.2 gives the corresponding maximum delay curves for reserved service traffic with and without the presence of saturated unreserved service traffic. The curves are plotted as the mean of eight separate simulation runs, each run for a total of 100,000 packets at each level of reserved service traffic load, with the 95% confidence interval plotted for each point. The maximum delay for reserved service traffic in the presence of saturated unreserved service traffic is approximately 40% greater than in the absence of unreserved service traffic. This difference is due to the probability of an incident reserved service packet finding the input node already busy serving an unreserved service packet that has achieved set-up.

The same measurements were taken of a number of other switch structures but using only a single simulation run on each of the delay curves. The switch structures

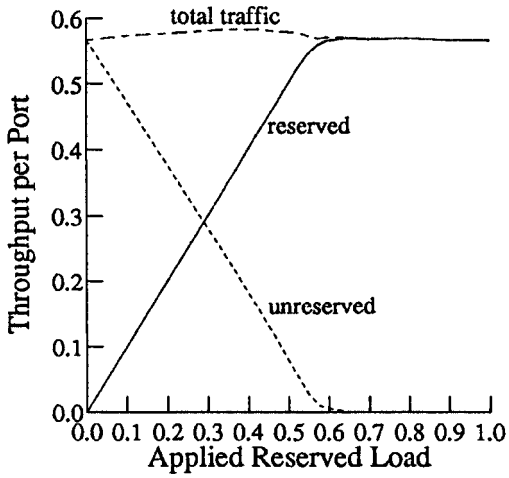

Figure 7.1: Throughput performance for the Poisson reserved service + saturated unreserved service traffic model.

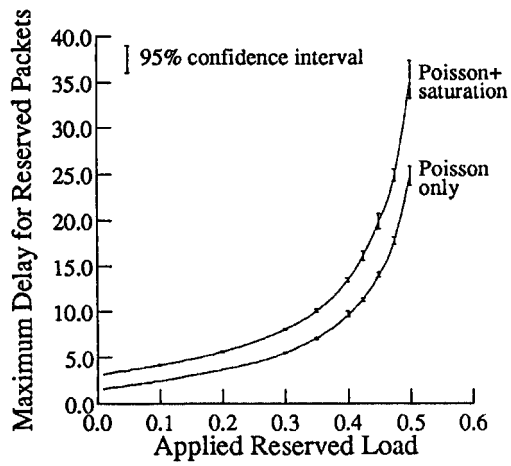

Figure 7.2: Maximum reserved service packet delay for the Poisson reserved service traffic model with and without saturated unreserved service traffic.

investigated were: the 64×64 regular two-plane delta with switching elements of degree 2 and 16; the 512×512 regular two-plane delta with switching elements of degree 8; the 64×64 sub-equipped Beneš with switching elements of degree 8 and the 64×64 crossbar switch. All switch structures yielded similar results scaled in proportion to the throughput at saturation of the respective switch fabric. Unless otherwise stated further measurements apply to the 64×64 pure input buffered two-plane delta network with switching elements of degree 8.

The same measurements were repeated with a non-uniform distribution of unreserved service traffic. Of the unreserved service packets, 80% were directed to 8 of the switch ports while the remaining 20% were directed randomly across all output ports. The throughput performance of the reserved service traffic was not affected but the maximum delay performance was reduced slightly. This reduction was due to the fact that the majority of the output ports of the switch were more lightly loaded with unreserved service traffic in the non-uniform distribution. It would appear that the load and distribution of the unreserved service traffic has only a slight effect on the maximum delay performance of reserved service traffic. Investigations also suggest that it is possible to operate a fast packet switch with input and output ports running at widely different mean traffic loads, as might be the case, for example, between ports connected to inter-switch trunks and those connected to local area networks. With highly non-uniform traffic distributions the total capacity of the switch is used less efficiently but the heavily utilised ports may offer a much higher throughput than for a uniform distribution.

In figs. 7.3 and 7.4 a Poisson reserved service traffic source is multiplexed with a Poisson unreserved service source at every input port of the switch. Fig. 7.3 shows the throughput performance of unreserved service traffic for several reserved service

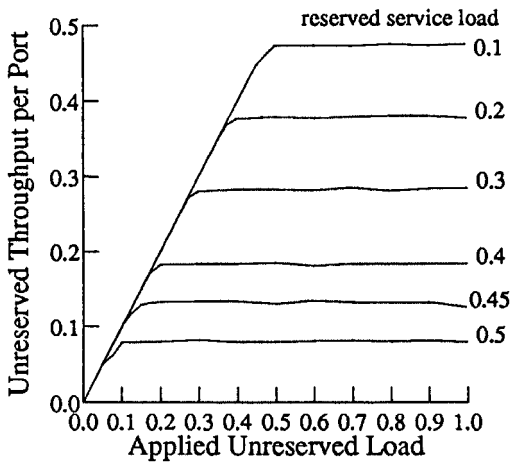

Figure 7.3: Unreserved service throughput performance for the Poisson reserved service + Poisson unreserved service traffic model.

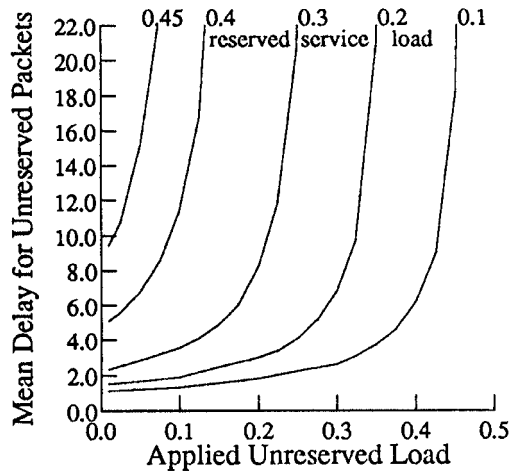

Figure 7.4: Mean unreserved service packet delay for the Poisson reserved service + Poisson unreserved service traffic model.

traffic loads. Fig. 7.4 shows the corresponding mean delay for unreserved service traffic. Both curves saturate at a level that reflects the remaining switch bandwidth available after serving the requirements of reserved service traffic. The reserved service throughput characteristic in this case is identical to that observed with a saturated unreserved service traffic source while the maximum reserved service delay is reduced in proportion to the amount that the total load on the switch falls below saturation.

To give a comparative impression of switch performance fig. 7.5(i) shows the maximum delay performance of various designs of fast packet switch of size 64×64 for Poisson traffic. Once again it may be seen that the performance of the pure output buffered switch is only slightly greater than that of the highest performance two-plane delta design. This in turn is of slightly greater performance than a two-plane Batcher-banyan (i.e. crossbar switch) as the latter is synchronous at the packet level and therefore cannot take advantage of input queue by-pass.

The performance of the two-plane delta networks of size 64×64 for Poisson reserved service traffic in the presence of saturated unreserved service traffic is presented in fig. 7.5(ii). The curves for the crossbar switch under Poisson traffic are reproduced for comparison. The maximum delay performance for reserved service traffic of the two-plane delta networks featuring output buffering is slightly impaired in the presence of saturated unreserved service traffic, particularly at low loads, due to the change in priority mechanism at the output ports. With output buffering the priority mechanism in the output port controllers will only accept a single unreserved service packet at any one time. Without output buffering an output port controller will reject unreserved service packets after detecting a failed reserved service packet set-up attempt until a reserved service packet has been serviced. The difference in performance between these two output port priority mechanisms is insufficient to be

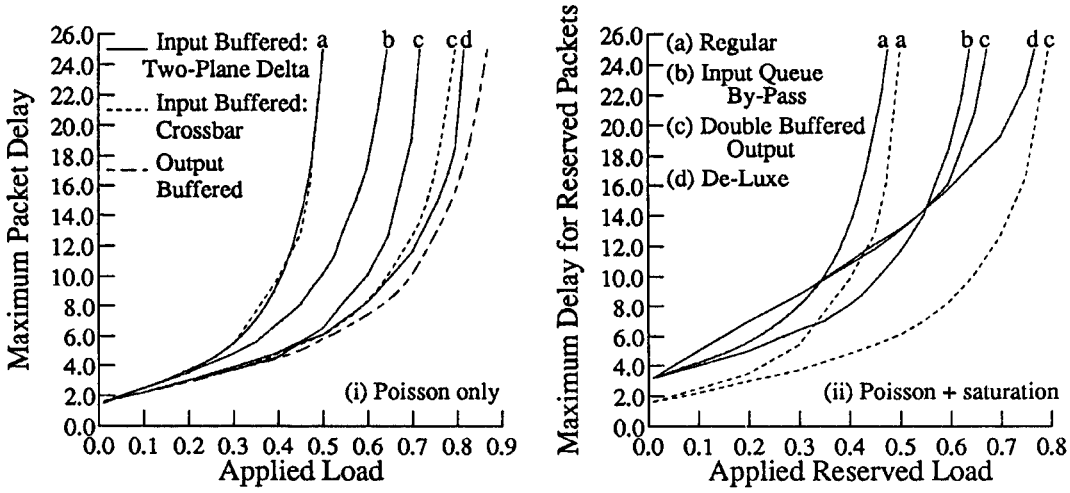

Figure 7.5: Comparison of maximum delay performance of various switch designs of size 64×64 for Poisson traffic with and without saturated unreserved service traffic.

of practical significance.

For the 64×64 regular two-plane delta network with Poisson traffic sources the queue lengths were observed to be short and to stabilise rapidly up to a traffic load of about 0.45. Beyond this traffic load the delay across the switch becomes increasingly sensitive to small changes in the applied load. This figure represents a load of 80% of the throughput at saturation of the switch fabric and provides a reasonable estimate for the upper bound of the applied reserved service traffic load for stable operation of the switch. Table 7.1 provides a comparison of the maximum delay performance of the various switch structures of size 64×64 for Poisson reserved service traffic. It presents results at the maximum load of 80% of their respective measures of throughput at saturation both in the presence and absence of saturated unreserved service traffic. For all switch structures the maximum delay in the absence of unreserved service traffic is in the region of 15 packet lengths while for structures based on the delta network in the presence of saturated unreserved service traffic it increases to about 20 packet lengths. This rule of thumb holds for all sizes and structures of switch investigated. For an implementation in CMOS operating at 50 MHz with 256 bit packets this result implies that 99% of all reserved service packets will traverse the switch within about 100 μsecs regardless of the load or distribution of the unreserved service traffic.

7.5 Talkspurt Voice

The Poisson arrival process is one of the simplest and most widely used traffic models but it is necessary to show that the performance results derived from the Poisson model bear some relation to the performance that might be expected from a more realistic model of multi-service traffic. Telephony voice traffic was chosen for a closer

Switch Design	*Throughput at Maximum Load*	*Delay at Maximum Load*	
		Poisson Only	*Poisson+ Saturation*
Two Plane Delta:			
Regular	0.453	14.3	20.3
Queue By-Pass	0.585	16.0	17.4
Double Buffer	0.639	12.2	20.0
De-Luxe	0.738	14.6	21.9
Crossbar:			
Regular	0.469	15.7	–
Double Buffer	0.718	14.2	–
Output Buffered	0.8	16.0	–

Table 7.1: Comparison of maximum delay performance of various 64×64 switch designs at maximum reserved service traffic load.

investigation as it has been widely studied, its characteristics are well known, it is fairly easily modelled to a reasonable degree of accuracy and it is of practical significance. The talkspurt voice model, described in section 7.3, was used to investigate the maximum delay performance for packet voice both with silence removed (talkspurt voice) and without the removal of silence periods (TDM voice). One problem with the modelling of talkspurt voice is the extremely long simulation runs required to simulate a reasonable number of talkspurts from each voice source. It was found that if each voice source was initialised to a random state within the talkspurt/pause cycle then the simulation would rapidly converge to a stable estimate of the maximum delay performance. Simulations at the higher loads took the longest to converge. At the highest simulated load of 80% of the throughput at saturation of the switch fabric, a simulation which ran for 50 seconds of simulated time demonstrated that a stable estimate of the maximum delay was attained after 2 seconds of simulated time. The talkspurt simulations were therefore run for 4 seconds of simulated time which was sufficient for the majority of sources to complete at least one talkspurt/pause cycle. The arrival of voice calls and the holding time of each call were not modelled. The same number of voice sources were modelled on each switch port and remained constant during each simulation run. The load on the switch was thus proportional to the number of voice sources on each switch port. With the parameters selected for the talkspurt voice model an applied load of 0.45 corresponded to 625 voice sources per switch port, and to 250 voice sources per switch port for the TDM voice model.

The maximum delay performance of the talkspurt and TDM voice models is compared to the Poisson reserved service traffic model in the absence of unreserved service traffic in fig. 7.6(a) and in the presence of saturated unreserved service traffic in fig. 7.6(b). These measurements apply to the 64×64 pure input buffered two-plane delta network with switching elements of degree 8 but measurements from other structures yield similar results. It is evident that within the region of stable operation there is no significant difference in the maximum delay across the switch

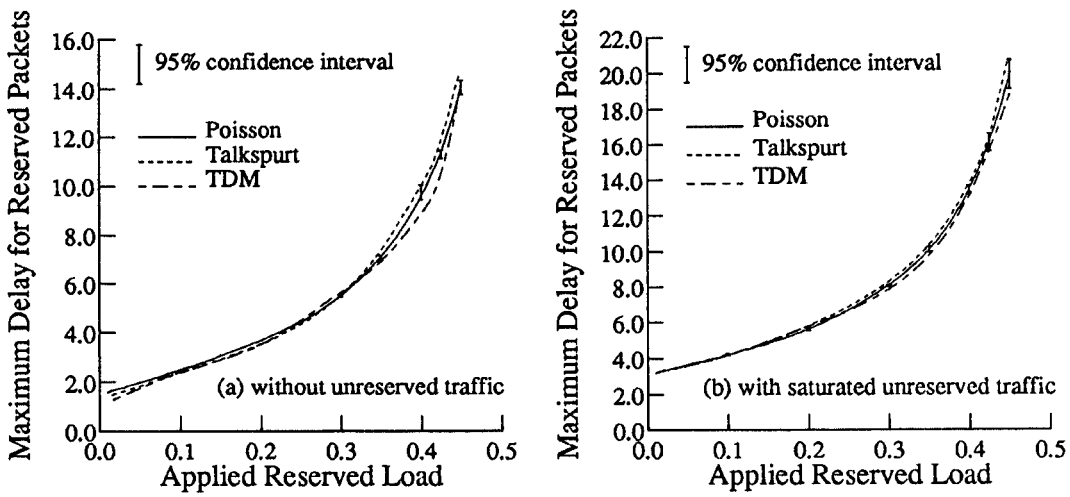

Figure 7.6: A comparison of maximum reserved service packet delay for Poisson, talkspurt and TDM voice models both with and without saturated unreserved service traffic.

for Poisson, talkspurt and TDM voice sources, either in the presence or absence of saturated unreserved service traffic. Furthermore an observation of the inter-arrival times of packets generated by the talkspurt model on a single input port reveals a very close approximation to the exponential distribution in agreement with the analysis presented in [78]. Thus the superposition of a large number of talkspurt voice sources may be modelled by a Poisson arrival process, with reasonable accuracy, for applied loads below about 80% of saturation. At loads in excess of about 80% of saturation the simulation model takes a long time to converge and analytical results suggest that its performance departs from that of the Poisson model [136, 63]. The departure from the Poisson model is to some extent dependent upon the number of sources multiplexed onto each switch port but operation at such high loads for reserved service traffic is unlikely to be required of a fast packet switch.

7.6 Packet Length

The effect of variable length unreserved service packets upon the performance of the reserved service traffic will now be examined. The results presented so far have assumed a constant packet length and all results have been normalised to become independent of the absolute packet length. In the following investigation all packets are assumed to consist of a header and an information component and the results are normalised to the value of the information component. First we consider the case in which reserved service packets and unreserved service packets are of different but constant length. The length of the unreserved service packet is expressed in terms of the reserved service packet information field, and all packets have a header of one eighth of the length of the reserved service packet information field. The throughput results

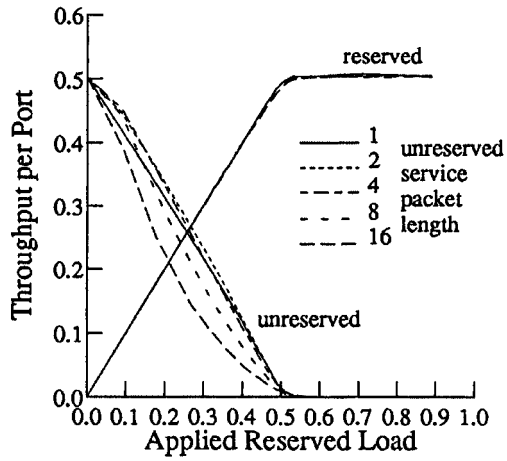

Figure 7.7: Effect of unreserved service packet length on throughput performance for the Poisson reserved service + saturated unreserved service traffic model.

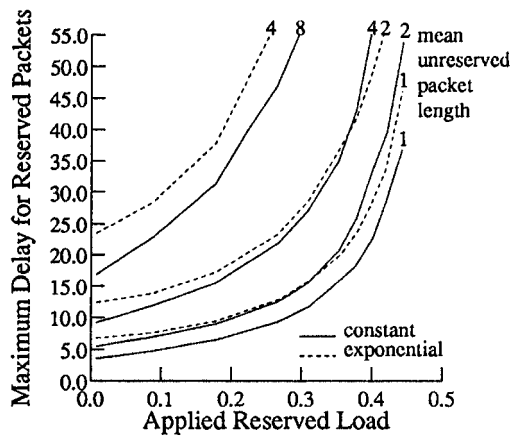

Figure 7.8: Effect of unreserved service packet length, constant and exponentially distributed, on maximum reserved service packet delay.

are presented in fig. 7.7 for the Poisson reserved service traffic model in the presence of unreserved service traffic. It may be seen that the reserved service throughput performance is not unduly affected by the unreserved service packet length. However, the unreserved service throughput at saturation, for large unreserved service packet lengths, is lower than that for small packets showing that the advantage of low packet overhead is rapidly outweighed by the superior multiplexing capability of small packet sizes.

If this result applied to multiplexed flows in general it would indeed be very interesting, however, it is likely to be the result of head of the line blocking in an input buffered fast packet switch with non-preemptive reserved service traffic priority. As the packet length of unreserved service traffic increases, so the probability of head of the line blocking in the reserved service input queues is increased. This in turn reduces the switch capacity available to unreserved service traffic. An examination of the case in which the length of the information component of all unreserved service packets was given by an exponential distribution revealed similar results with a further reduction in the unreserved service throughput performance of between 10% to 20%, due to the variability in packet length.

The effect of the unreserved service packet length upon the maximum reserved service packet delay performance is given in fig. 7.8. As expected, a variable length unreserved service packet exerts a greater impairment of performance than one of constant length, and the shorter the mean packet length the less the reserved service packet delay performance is affected. Hence, conventional sizes of data packet must clearly be broken down into short packets for multiplexing with delay sensitive traffic but this may not be necessary for a 'data-only' environment.

The throughput and maximum delay performance was also considered of Poisson traffic in the absence of unreserved service traffic in which all packet lengths followed a uniform random distribution of ±10% about the mean value. No drop in performance was detected with respect to that of constant length packets. For the case in which all packet lengths followed an exponential distribution about the mean, a drop in the throughput performance of about 12% was measured with a corresponding impairment of the delay performance. Thus the switch is insensitive to the variation in packet length that might be introduced by a line code employing 'bit-stuffing' but larger variations will cause a reduction in performance.

7.7 Buffer Overflow

In the performance measurements presented so far in this chapter no constraint has been imposed upon the length of the input queues. In practice an upper limit of 100 packets was imposed upon all queues but under normal operating conditions this limit was never approached. It is interesting to consider the maximum length of input queues required to maintain a given probability of buffer overflow.

An approximate analysis for slotted traffic is presented in [71] in which the buffer overflow probability is given by the expression:

$$\frac{p(2-p)}{2(1-p)}\left[\frac{p^2}{2(1-p)^2}\right]^B$$

The applied load is given by p which represents the probability of an input timeslot containing a packet while B gives the number of packet buffers in each input queue. Using a simulation model to measure the packet loss probability a minimum buffer overflow of 100 packets or so is required in order to yield a measurement of reasonable accuracy. This makes the measurement of packet loss probabilities better than 10^{-4} difficult using a simulation model without excessively long simulation runs [5]. A linear regression of the approximate simulation results for the $\log_{10}$ of the buffer overflow probability is given in fig. 7.9 for buffer lengths of 8, 12, 16, and 20. These results derive from an asynchronous crossbar switch fabric under slotted traffic with a retry delay of 10% of the packet length. The curves are extrapolated to a packet loss probability of 10^{-6} and compared to the analytical result for a synchronous crossbar switch fabric. The results differ due to the difference in throughput performance between the synchronous and asynchronous crossbar switch models and also because both results are only approximations. Further investigation suggests that at a load of 80% of the throughput at saturation of the switch fabric the single plane regular delta structure has a slightly poorer packet loss performance than the crossbar switch and the two-plane delta network with output buffering a slightly better performance.

To gain a more accurate insight into the packet loss probability of the various switch designs a simulation model would be required designed specifically for that task. These results, however, indicate that at a load of 80% of saturation, with input buffers of length 20 packets, all switches offer a packet loss probability better than

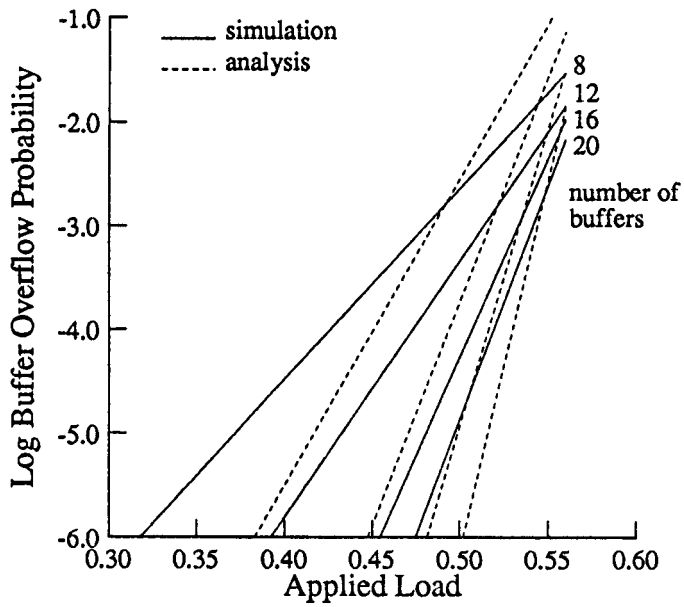

Figure 7.9: Buffer overflow probability for the input buffered crossbar switch.

10^{-6} for slotted traffic. In the case of two-plane switches with output buffering the performance appears to be significantly better than 10^{-6}.

7.8 Discussion

The major result of the performance investigation of the fast packet switch for multi-service traffic is that if a minimum of two classes of traffic are defined a statistical guarantee can be made concerning the delay performance of the higher priority class of traffic. For all switch designs investigated 99% of all reserved service packets will traverse the switch within a delay of less than 20 packet lengths provided that the reserved service traffic load does not exceed 80% of the throughput at saturation of the switch fabric. This guarantee holds regardless of the traffic load or distribution of the unreserved service.

The reserved service is so named because before an incoming call is accepted the switch checks to see that the required bandwidth is available before allocating it to the new call. If insufficient bandwidth is available the call is refused. To accomplish this the switch may be able to make peak and average load measurements on the input and output ports of the switch required by the new call. Alternatively it might maintain a sum of the bandwidth already allocated on all input and output ports. The new call supplies an estimate of the bandwidth resources required and provided that both input and output ports may accommodate this load the new call is accepted. For this bandwidth reservation mechanism to work each call must be monitored by a policing mechanism to ensure that it does not exceed the bandwidth resources

that it was allocated when accepted. One possible method of implementing this control mechanism at the entry point to a network of fast packet switches is discussed in [5] which uses three parameters to characterise the bandwidth requirements of a call: peak bandwidth, average bandwidth and burstiness. From these parameters a method of estimating the effective bandwidth required by a call is discussed and a simple hardware mechanism is given to ensure that the call does not exceed the agreed parameters.

Some sources of reserved service traffic are easily characterised in terms of their traffic requirements. Voice telephony is an obvious example in which the traffic characteristics depend upon the coding scheme employed, but for a large enough multiplex of sources, reliable statistical assumptions may be made. Video sources are not so easily characterised and their traffic requirements depend heavily upon the coding employed and the picture content. Video represents a class of traffic that may be both very bursty and also exhibit a low delay requirement. One solution for handling such traffic is to extend the priority scheme to more than two levels and to allocate video traffic a priority below voice traffic but above delay insensitive traffic. An estimate of the average bandwidth requirement of a video call must be made by the bandwith reservation mechanism and the entire class of video traffic must be subjected to some form of policing mechanism to ensure that it does not exceed its bandwidth allocation [2]. The parameters of the policing mechanism might be varied according to the load of lower priority traffic in the switch. This would permit greater flexibility when the switch was more lightly loaded. Another possibility for handling video sources is to use variable bit rate coding with interaction between each source and the network to vary the source traffic characteristics during a call according to the bandwidth available across the network.

It must also be recognised that the switch is a statistical switch and that the measure of delay performance for reserved service traffic is an average measure taken over all ports of the switch for the duration of the measurement. There will be some shorter periods of time in which the delay performance is worse than the average. Also the delay performance is averaged over all calls on each switch port thus some calls may receive a poorer performance than others during the lifetime of the connection. In order to reduce this effect it may be necessary to select a maximum load for reserved service traffic below 80% of the throughput at saturation depending upon the delay sensitivity of the source traffic and the sensitivity of the switch to short duration overloads.

When bandwidth is allocated to the reserved service it is not removed from the pool of bandwidth available to both reserved and unreserved service calls as would be the case in a circuit switch. All of the switch bandwidth is shared between all traffic, with reserved service packets receiving the higher priority. If the unreserved service traffic is also to be assured of a worst case delay performance, the unreserved service calls will also need to be monitored and bandwidth allocated but probably in a less stringent manner than for the reserved service.

Although only two classes of traffic have been investigated it is clear that the technique may be extended to a larger number of classes of traffic. Internal network

signalling traffic, for example, may require the highest priority while the unreserved service may be divided into interactive services which have some delay requirement and bulk transfer services that are much less sensitive to delay.

7.9 Summary

An extension to the design of the switch has been proposed to support two fundamental classes of traffic. Reserved service traffic receives a higher priority in the switch fabric and handles classes of traffic that are sensitive to delay whilst the unreserved service caters for traffic that is less delay sensitive. Simulation results indicate that for a Poisson reserved service traffic loading of up to 80% of the throughput at saturation of the switch fabric, the upper bound on delay for 99% of all incident reserved service packets is in the region of 20 packet lengths. Further, unreserved service traffic may be multiplexed with reserved service traffic, at every input port of the switch, so as to operate the switch continuously at saturation, without affecting the bounded delay performance of the reserved service. This result has been shown to hold for a wide range of switch structures and switch fabric design parameters. Also the reserved service throughput and delay performance appears insensitive to the arrival distribution and to the destination distribution of unreserved service traffic.

A closer investigation of the delay performance of the switch for voice traffic modelled as a superposition of individual packet voice sources on each switch port, both with and without silence detection, reveals no significant departure from the delay performance of the Poisson model. This traffic model was observed to give a packet arrival distribution closely approximating that of a Poisson source.

For delay sensitive, reserved service traffic performance, the packet length for both reserved and unreserved service traffic should be kept short and constant. No performance impairment is introduced by a $\pm 10\%$ variation in packet length but an exponential distribution of packet lengths causes a loss in throughput performance of the order of 12% for a 64×64 regular two-plane delta network. For a single service implementation, moderately insensitive to delay, variable length packets of any reasonable maximum length may be supported.

A cursory inspection of the buffer overflow probability suggests that an input buffer length of 20 packets is sufficient to offer a packet loss probability of less than 10^{-6} for slotted traffic at a traffic load of 80% of the throughput at saturation for all switch designs.

Chapter 8

Implementation of the Fast Packet Switch

The performance of the fast packet switch has been investigated by means of a simulation model. Confidence in the accuracy of the simulation results has been gained by comparison with a number of analytical models with which close agreement has been demonstrated. The error introduced into the simulation model has been investigated to yield upper and lower bounds upon the performance results. All of these results, however, assume the availability and performance of the crossbar switching element. This chapter presents an investigation of the design and implementation of an experimental crossbar switching element [114] and also of a simple input port controller from which a fast packet switch may be constructed. Measurements of the experimental implementation demonstrate that its performance is very close to that assumed by the simulation model. A discussion then follows of the various enhancements required and options available for the full-scale implementation of a fast packet switch.

8.1 An Experimental Implementation

To investigate the hardware realisation of a crossbar switching element and to demonstrate the simplicity of the design an implementation in gate array technology was undertaken. A low volume prototype gate array fabrication service was available in which an electron beam was used to write the final layer metal interconnection pattern directly onto a wafer of gate array devices. The target gate array family was the Texas Instruments TAHC series in 3 μm HCMOS of which the largest device, the TAHC10, was selected. This device contains the equivalent of 1120 two input NAND gates with 40 input/output buffers. Of the raw gates available a maximum of 896 could be accessed using the computer aided design software and of these, few designs could use much more than 50% due to the layout and routing restrictions. (With only a single layer of metal interconnection available it was in general necessary to sacrifice a large number of transistor cells in order to provide interconnections across

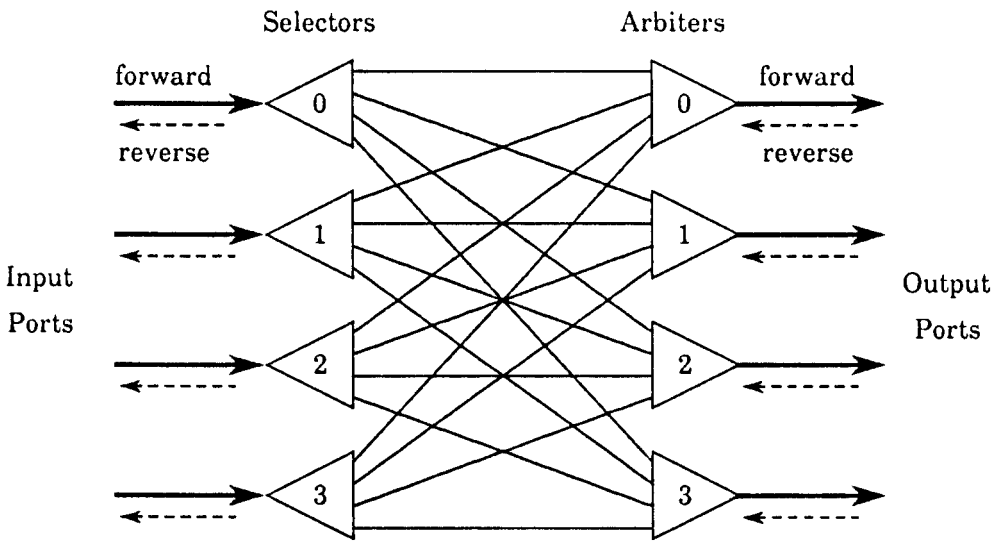

Figure 8.1: Structure of the 4×4 crossbar switching element.

the device.) The computer aided design software included a complete suite of design, layout, testing and fault location programs with an adequate library of basic logic elements, e.g. gates, flip-flops and latches. Two separate devices were constructed on separate TAHC10 gate arrays: a 4×4 crossbar switching element, and a single port input port controller with a general purpose 8 bit microprocessor bus interface.

8.2 The Switching Element

Although the simulation results indicated that 8×8 would be the preferred size of switching element the limited size of the gate array available restricted the implementation of the switching element to a 4×4 design. The complete structure of the 4×4 crossbar switching element, implemented on a single TAHC10 gate array, is illustrated in fig. 8.1. The structure consists of a set of identical selectors and a set of identical arbiters in a fully interconnected topology. Each selector and arbiter operates independently and asynchronously at the packet level but synchronously at the bit level with the use of a common system clock. Increasing the size of the switching element to 8×8 and beyond is merely a matter of increasing the size of the selectors and arbiters and interconnecting one selector per input port and one arbiter per output port in a completely interconnected topology. Crossbar switching elements of any integer dimension may be constructed in this manner and non-square, concentrating or expanding switches, are also possible.

Operation of the Switching Element

Each link of fig. 8.1 consists of two paths, a forward path which carries the traffic and a reverse path used to pass collision information back through the switch fabric. Each selector monitors an input port and when an idle to active transition is observed on the forward path it removes the first two digits from the tag at the head of the incident packet. These digits are used to select the required output port and the state of the relevant arbiter is inspected. If the arbiter is free, the remainder of the packet is passed over the forward channel to the output port. The reverse channel is also connected transparently from output port back to input port so that collision information from later stages in the switch fabric may be quickly transmitted back through the switch fabric to the input port controllers. If the arbiter indicates that the output port is busy, the selector asserts the collision (busy) signal on the reverse channel of the input port. The selector returns to the idle state when an active to idle transition is detected on the forward path.

The arbiter monitors the forward channels from each of the selectors. As soon as a channel goes active it is switched through to the output port and the reverse channel is connected transparently. The reverse channels to all other selectors are then set to the busy state. The arbiter returns to the idle state as soon as the forward path of the selected channel goes idle.

Implementation of the Selector

The selector is required to distinguish between an active and an idle forward channel. Normally this would be achieved using some form of line code but with a total budget for the switch of less than 500 gates, the selector must be implemented with less than 60 gates which leaves little spare logic to handle a line code. The device does however have an excess of input and output pins thus a simple solution was adopted for the forward path. The forward path was formed from two signals on separate I/O pins: the data signal which carried the information component and the active signal which defined when the data signal carried valid information. Each input (and output) port thus carried three signals: the data and active signals forming the forward path and the busy signal which was the reverse path.

The implementation of the selector is shown in detail in fig 8.2. When the incoming active signal is asserted the route selector latches the first two bits of the data signal. These are applied to the selector switches which connect the 'busy' and 'active' signals of the appropriate arbiter to the collision controller. The route selector then issues an active signal which causes the collision controller to examine the state of the incoming busy signal from the selected arbiter. If this signal indicates 'not-busy' it is connected transparently through to the outgoing busy signal of the input port and the collision controller asserts the active signal which propagates through the selector switches to the selected arbiter. If the incoming busy signal indicates 'busy' then the collision controller asserts the outgoing busy signal of the input port and keeps the outgoing active signal in the 'inactive' state. The selector is reset to the idle state when the

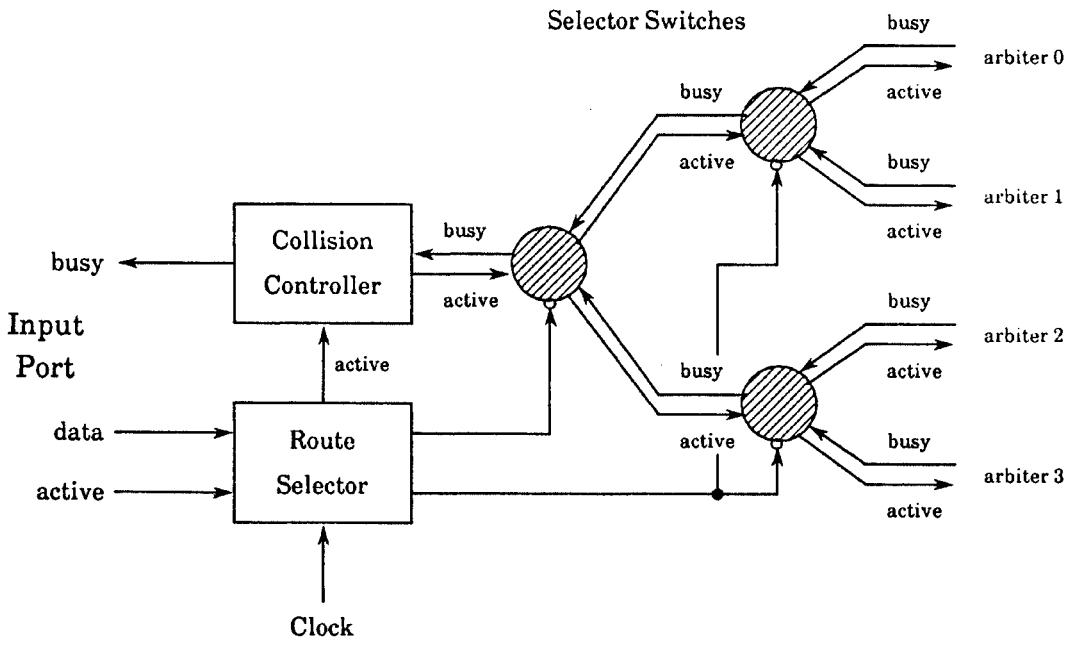

Figure 8.2: Implementation of a 1 to 4 selector.

incoming active signal of the input port indicates the idle condition. The complete 1 to 4 selector required a total of 42 gates.

Implementation of the Arbiter

The implementation of the arbiter is shown in detail in fig. 8.3. The active signals from each of the selectors are fed into a priority encoder. If multiple active signals are asserted during the same clock period the priority encoder selects one of them according to a simple priority scheme. The event is rare so a simple priority scheme should be sufficient otherwise a round robin or random scheme could be employed. When an incoming active signal is asserted and selected, the priority encoder sets the arbiter switches to connect the incoming data signal, corresponding to the selected port, to the retiming flip-flop on the data line of the output port. The priority encoder then issues the outgoing active signal which enables the connection of the incoming busy signal of the output port through the arbiter switches to the appropriate selector. The arbiter switches also set all other busy lines to the busy state which prohibits other input ports from attempting to access a busy arbiter. When the active signal of the selected input port drops to inactive, the outgoing active signal returns to the inactive state after a delay of one bit time. Thus the tail of the active signal propagates across the switch fabric together with the tail of the data signal. The arbiter is ready for access by another input port after a further delay of only one bit time. The complete 4 to 1 arbiter required a total of 50 gates.

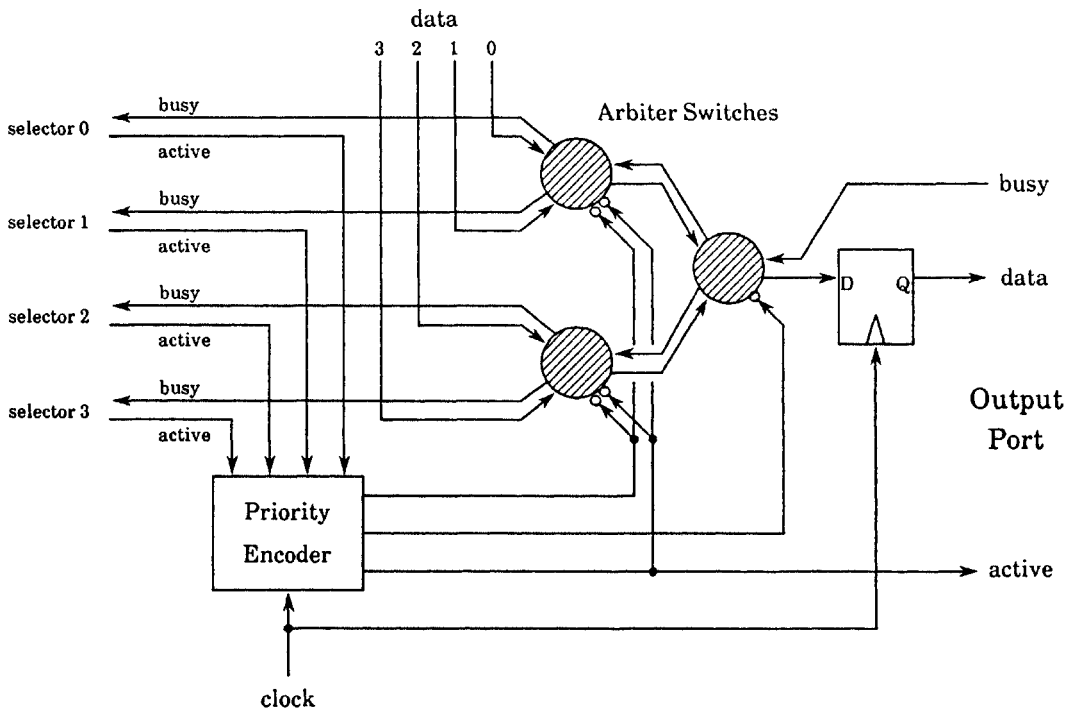

Figure 8.3: Implementation of a 4 to 1 arbiter.

Implementation of the Switching Element

The critical path in the set-up of a packet across the switching element is from the assertion of the active signal at the route selector to the selection of the appropriate data signal by the arbiter switches in time to be clocked by the output flip-flop. In the existing design this must occur within one bit time which limits the clock frequency of the experimental design to 8 MHz. The inclusion of one or two bits delay at the input ports of the switch would ease this restriction. In the existing implementation, each 4×4 switching element inserts a delay of only one bit time into the data path and the data path passes through no more than two gates and a flip-flop.

The complete 4×4 crossbar switching element required a total of 378 gates. This represents a 42% utilisation of the nominal size of the gate array which ought not to have caused any layout difficulties. Several difficulties were however encountered, most of them resulting from the fact that only a single layer of metal interconnection was available. This required interconnections in the vertical direction to proceed via polysilicon tunnels of which there were generally insufficient and the use of which increased the delay across the interconnections and required the sacrifice of transistor cells. The use of a gate array with two layers of metallisation would overcome these difficulties.

8.3 The Input Port Controller

The function of the input port controller of the experimental implementation was to demonstrate the functional operation of the switching element and to measure its performance. To achieve this it was not considered necessary to send real data packets across the switch but merely to set up a connection to the requested output port and to hold the connection for the length of a packet while sending a preset bit pattern across the connection. This greatly simplified the design of the experimental input port controller whilst allowing the functional operation of the crossbar switching element to be investigated using an oscilloscope and a logic analyser. To have attempted to transmit real data packets across the switching element would have required a much more complex experimental set-up and would have yielded no further results on the performance of the switching element than the simple arrangement to be described.

The experimental input port controller was implemented in a single TAHC10 gate array and controlled a single input port of the switching element. Four of these devices were therefore required to investigate the operation and performance of a single 4×4 switching element, one on each of the four input ports. A functional diagram of the experimental input port controller is given in fig. 8.4. The device is connected to the outside world via a standard 8 bit microprocessor bus interface and to the switching element via the three signals data, active and busy of the output port. Three status outputs indicate the state of the input port controller and their condition may also be read across the bus interface on the lowest three lines of the data bus. To initiate operation of the input port controller an 8 bit word is written to the bus interface, passed through the latch and into the recirculating shift register. The most significant bits define the required destination while the remaining bits form the bit pattern to be transferred across the data line. The active signal is asserted and the data transferred across the data switch to the output port. If the incoming busy line is asserted within 16 bit times the active signal is dropped and the sequencer waits for a retry delay of 32 bit times from the beginning of the set-up attempt before commencing the next set-up attempt. If no busy signal is received within 16 bit times of the beginning of a set-up attempt the path set status bit is established and the path held for a total packet length of 256 bits including the routing tag. On completion of packet transmission the input port remains idle for 16 bit times to prevent an input port with a large number of packets to the same destination from excluding other input ports from obtaining access to that output port.

A second mode of operation was added in which a path once set remains held until released by a specific instruction received across the bus interface. In this mode the data path, once established, is switched by the data switch to connect an external data signal to the output port. This permits packet data to be transmitted across the switching element using an external framing and line coding device such as an HDLC chip. It also permits the operation of the switching element to be slowed down for observation. The experimental input port controller required a total of 292 gates and presented no implementation or layout difficulties on the TAHC10 gate array.

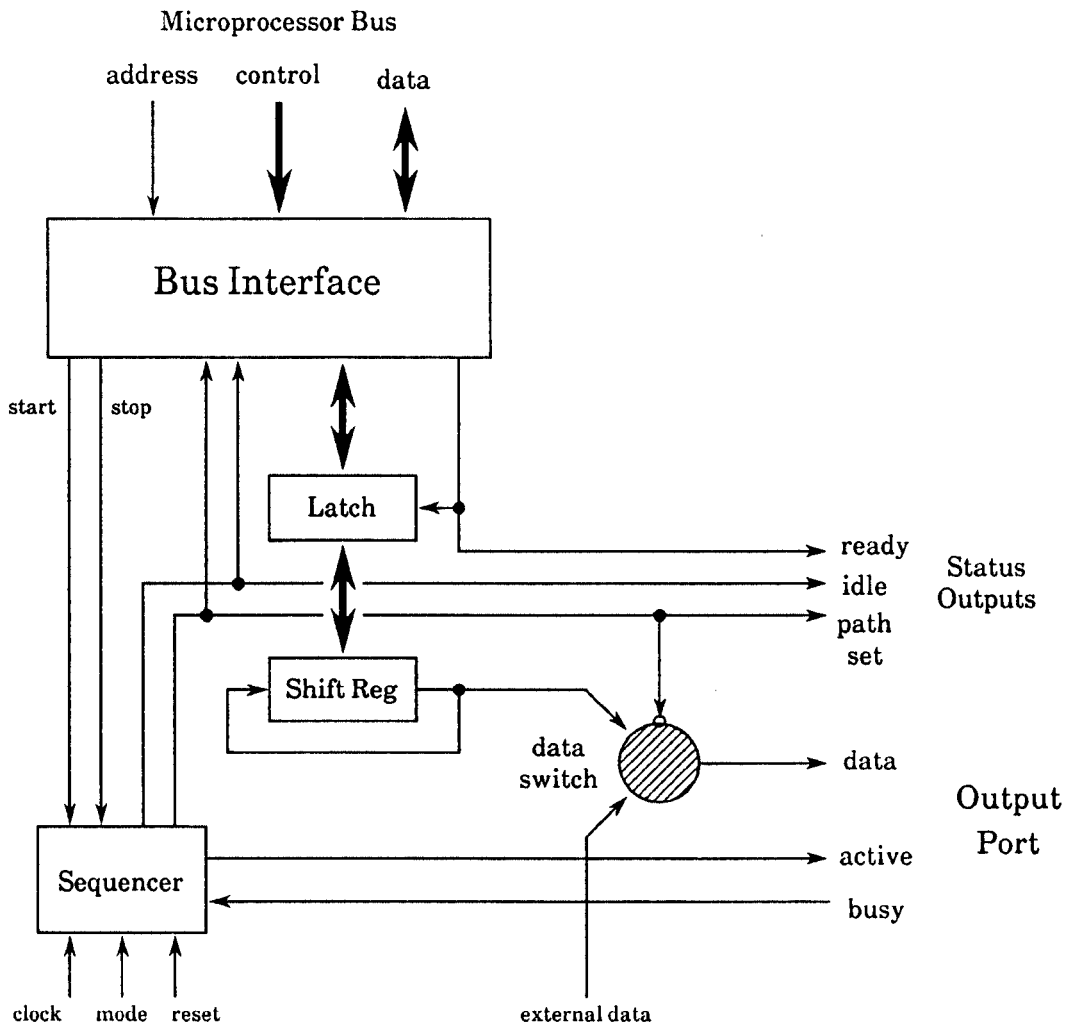

Figure 8.4: The experimental input port controller.

8.4 Performance Measurements

Each of the four input ports of a 4×4 crossbar switching element was connected to an experimental input port controller which were in turn connected to the system bus of a microcomputer. The microcomputer could thus request the transmission of test packets from any input port to any output port and inspect the status of each of the input port controllers. The busy signals of the output ports of the switching element were also interfaced to the system bus so that each output port could be enabled or disabled. The functional operation of the switching element was investigated using an oscilloscope and a logic analyser in the test packet mode and by transferring external data patterns across the switching element. The measured operation of the switching element was in agreement with that predicted by the logic level simulator of the CAD software at a clock rate of up to 8 MHz. The input port controller from input port

number 3 of the switching element was disconnected and that input port connected to output port number 3 of the switching element. This allowed test packets to be routed through two switch stages in cascade demonstrating that a four bit routing tag was correctly interpreted and indicating that the routing mechanism would operate successfully with an arbitrary number of stages of switching elements in cascade.

To measure the throughput at saturation performance of the switching element the system clock was decreased until the microcomputer was capable of keeping all input ports saturated with test packet requests with a uniform random distribution of destination tags. The measurement was repeated eight times for a total of 200,000 packets each to yield a throughput at saturation result of 0.6204 ± 0.0004. The same measurement was taken using the simulation model set up to simulate the characteristics of the 4×4 crossbar switching element. The simulation model gave a throughput at saturation of 0.6160 ± 0.0001 which differs from the measured value by 0.7%. This difference is of the same order of magnitude as that between the analytical [76] and simulation results for the 4×4 crossbar switch. It originates from the slight non-uniformity of the random number generators used.

The active line of one of the input ports was connected to a counter so that the average number of retry attempts per packet could be measured for a switching element at saturation. This figure could also be calculated from the number of packets generated within the period of measurement. The measurement of the average number of retries per packet yielded a value of 4.30 ± 0.02 which agreed with the calculated value to within 0.25% demonstrating that the switching element was indeed operating at saturation and behaving as predicted. The measurement also agrees closely with the result of the simulation model for the average number of retries per packet at saturation, to within 0.5%. This, however, is hardly surprising as the average number of retries per packet at saturation is closely related to the throughput at saturation.

8.5 Towards a Full-Scale Switch Implementation

The experimental switch implementation has demonstrated that a crossbar switching element may be constructed in current gate array technology using very few gates and also that its performance agrees closely with that predicted by the simulation model. The developments required to construct a full-scale implementation of the fast packet switch, based upon the results of the experimental model, will now be discussed.

The basic structure of the fast packet switch configured for general purpose communications applications is shown in fig. 8.5. The separate input and output port controllers of every k^{th} input and output port of the switch fabric have been combined to form I/O port controllers which handle full-duplex ports. The overall structure closely resembles a PABX or a current telecommunications circuit switch consisting of a control plane, a switch plane and some line cards.

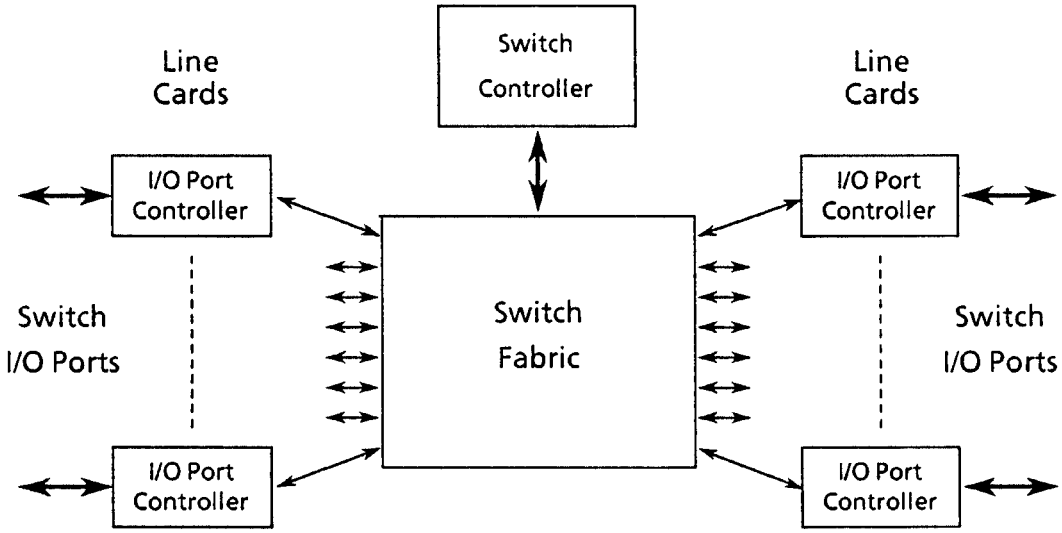

Figure 8.5: Structure of a fast packet switch implementation.

The Line Card

Each line card contains one or more I/O port controllers, the basic structure of which is given in fig. 8.6. Packets arrive in the input FIFO with a label at the head of the packet. This label is here referred to as the virtual circuit indicator (VCI) as it identifies the virtual circuit to which the packet belongs. The VCI of the packet at the head of the FIFO is latched and used to address a memory called the map. The map contains a replacement VCI for the next stage of the virtual circuit and thus maps the incoming VCI address onto the outgoing VCI address space. It also contains a tag which identifies the required output port of the switch. It may also contain information such as the priority of the connection, the type of traffic associated with the connection and details of a reverse connection, this information being available to the control hardware. The tag is prefixed to the packet, the new VCI replaces the old, and the packet is launched into the switch fabric. If the packet set-up attempt fails, the retry counter is incremented which may cause the most significant digits of the tag to change if the input port controller is searching through multiple paths. The contents of the map are updated by the switch controller possibly by the use of a separate control bus. In a large switch this method of control would be cumbersome thus an alternative is to define a special VCI that would allow the switch controller to update the map of each I/O port controller via special packets transmitted across the switch fabric.

The line code and packet framing function has been indicated in fig. 8.6 both on the I/O side of the port controller and also on the switch fabric side. The very different requirements on either side of the I/O port controller would almost certainly ensure that different line codes were used. If the switch port were interfaced to a local or a metropolitan area network the line code would be defined by the network and the switch interfaced to the network at the input and output FIFOs. If transmission

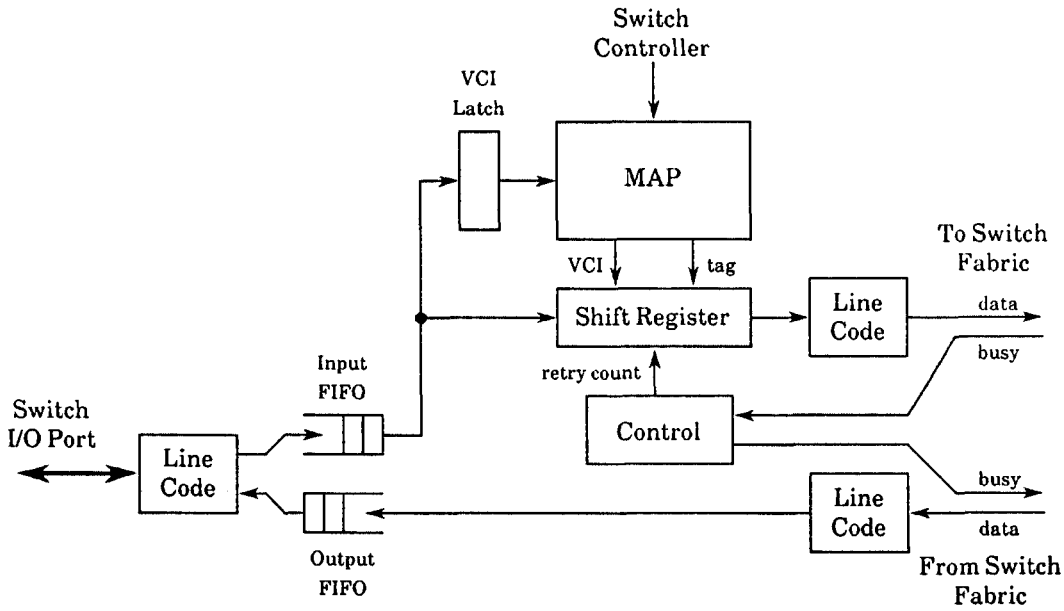

Figure 8.6: The I/O port controller.

across a point to point link is required three options are available for the line code:

- The packet may be delimited by a special bit pattern and the packet 'bit-stuffed' to avoid the occurrence of the special pattern within the packet, e.g. HDLC.
- Each N bit word may be coded using N+1 bits. The extra bit may be used to indicate the start and continuation of the packet, as in [100], or to provide a menu of special symbols as in FDDI [129].
- An alternative for fixed length packets is to divide the line into packet slots and to transmit a synchronisation pattern in the unused slots as proposed in the Prelude fast packet switch [141, 31], or to support the slotted structure in a conventional TDM frame.

It is probable that some form of error checking will be required on a per packet basis. The most suitable location for including this function is on entry to, or exit from, the input FIFO of the I/O port controller and not within every switching element within the switch fabric. Two levels of error checking may prove useful, one which protects the packet header and one for the information field. A packet with an error in the header should be discarded but if the error is in the information field it should be marked so that the destination may decide what action to take. Some classes of traffic are insensitive to occasional random errors but other classes will require retransmission of the packet.

If input queue by-pass is required to enhance the performance of the switch it must be implemented in the I/O port controller. It is possible, by the use of input

Size	*Gate Count*
2×2	250
4×4	600
8×8	1900
16×16	6000
32×32	21000

Table 8.1: Estimated complexity of crossbar switching elements.

queue by-pass, for packets arriving on the same virtual circuit to be delivered out of sequence. It may be possible to reconstruct the original sequence by the use of an end-to-end sequence number but this may prove undesirable. A simple way to avoid out of sequence errors would be to restrict the queue by-pass algorithm to just the first two packets on the queue that are routed to different ports. According to [100] this would offer about half of the performance improvement compared to applying the by-pass algorithm to all of the packets on the input queue. One possible method of implementing the full queue by-pass algorithm, while avoiding out of sequence errors, is to use a memory with one bit representing each of the output ports of the switch. The algorithm starts with the packet at the head of the queue and clears the memory. At every consecutive unsuccessful packet set-up attempt a note of the busy output port number is made in the memory and further attempts to that port are prohibited. The algorithm returns to the head of the queue and clears the memory on a successful packet transmission or on reaching the end of the input queue.

The Switch Fabric

The experimental model employed a very simple method of line coding and packet framing within each switching element. It required three signals for every link across the switch fabric, two forward and one reverse. While this may be acceptable for small switches it would in general be better to reduce the connections across the switch fabric to a single forward signal with a single reverse signal. This may be realised with a simple line code on the forward path in which a start bit is prefixed to the front of the tag at the head of the packet and the information field 'bit-stuffed' so that no more than N consecutive zeros occur. Whilst removing the relevant digits from the tag each switching element regenerates the start bit at the head of the packet. After the passage of the minimum packet length each switching element searches for a sequence of N+1 consecutive zeros to indicate the end of the packet. Assuming the use of a simple line code such as the above, an estimate of the complexity of the various possible sizes of crossbar switching element is presented in table 8.1 based upon the results of the experimental model. It is clear from the table that crossbar switching elements of up to size 16×16 may be fabricated in gate array technology.

For large sizes of switch, the use of two signals for every link in the switch fabric will be undesirable due to the constraints of interconnection and switch fabric par-

Technology	*Bandwidth per Port*
3 μm CMOS	10 Mbits/sec
2 μm CMOS	50 Mbits/sec
BiCMOS	250 Mbits/sec
ECL	500 Mbits/sec
GaAs	1 Gbit/sec
Photonic	$>$ 1 Gbit/sec

Table 8.2: Approximate maximum bandwidth per switch port for various implementation technologies.

titioning. The switch fabric may be constructed with a single connection for every link if it is operated in both directions in half-duplex mode. At the beginning of a set-up attempt all free switching elements in the path pass the packet tag forwards. When the tag has passed, all switches in the path pass the acknowledgement signal back across the path in the reverse direction. If the signal indicates collision the switching elements in the path are released. If the acknowledgement signal indicates successful connection to the output port, the switching elements pass the packet in the forward direction across the switch fabric. This 'ping-pong' packet set-up cycle slightly increases the time spent in packet set-up and may thus increase the internal blocking of the switch fabric but this impairment is likely to be outweighed by the reduction in the interconnections required within the switch fabric.

The internal design of the switching element presented here uses a large number of paths in a fully connected topology with each path operating in serial mode at the full speed of the switch fabric. The alternative, proposed in a number of fast packet switch designs, is to group the serial bit stream arriving at each switch port into word parallel form on entry to the switching element and to use some form of shared medium interconnection mechanism within the switching element. This allows the switching element to operate more slowly, at the word rate rather than at the bit rate of the switch fabric, but requires a more complex design.

The design of switching element presented in this chapter may be implemented in current 2 μm CMOS to achieve speeds of around 50 MHz. Beyond this speed, the simplicity of the design requires only the data path to operate at full speed. The majority of the logic in the switching element handles packet set-up and if a small increase in overhead is permitted in the packet set-up time then this logic can operate at a slower speed than that within the data path. Thus implementation may be possible in BiCMOS in which the data paths use bipolar technology while the packet set-up logic uses CMOS. Likewise for ECL, only the data paths will be required to operate at high speed thus reducing the power dissipation of the majority of the logic. Considering implementation in GaAs technology, [150] gives details of a device of 2200 gates with a power dissipation of 600 mWatts operating at 900 MHz, which is of the same dimensions as that required by an 8×8 switching element. A table of the approximate operating speed, and thus the maximum bandwidth per

switch port, is given for various technologies in table 8.2.

The design of the switching element may lend itself well to wafer scale integration [21]. Rows of selectors and arbiters may be located and interconnected on a single wafer of silicon to form a very large switch fabric. Care would have to be taken in the design to ensure that the effect of faulty nodes was localised and that sufficient redundancy was provided to cope with both faulty nodes and I/O pads. If sufficient multiple paths were provided across the switch fabric it would be possible to exploit the inherent fault tolerance of the design. Care would also have to be taken in the distribution of the clock signal. Using wafer scale technology it may be possible to construct switch fabrics of up to 1000 switch ports on a single wafer.

If the required operating speed of all switch ports is less than the maximum operating speed of the switch fabric then blocking within the switch fabric may be reduced by operating it at a higher speed than the switch ports. In this case buffering will be required at both input and output ports to dissociate the speed of the switch fabric from that of the switch ports. It will also be necessary to completely receive a packet in the input buffer before transmitting it across the switch fabric but this will be of no significance if the packets are short. If only a few ports are required to operate at very high speed, e.g. those terminating high speed trunks, then the high speed lines may be shared between a number of switch fabric ports. Packets from the high speed line may be served by any input port controller of the group and switching elements in the last stage of the switch fabric may be modified to direct a packet to any free output port controller of the group. The use of this technique is liable to cause out of sequence errors between packets travelling across the same virtual circuit.

The Switch Controller

The major function of the switch controller is to participate in the signalling protocol of a network of fast packet switches, to set up and clear down connections across the switch and to update the connection tables (maps) within each of the I/O port controllers. If a congestion control algorithm is to be implemented within a network of fast packet switches then the switch controller may be required to participate in congestion control. The switch controller will also be required to undertake maintenance and fault location functions. It may also be required to support a simple datagram service. A datagram service may easily be implemented on top of a fast packet switched network by using permanent virtual circuits between the switch controllers of the network to handle datagram transfer. If the datagram traffic load were to become too heavy for the switch controller it could be transferred to special purpose datagram servers.

8.6 Summary

The implementation of a 4×4 crossbar switching element in 3 μm HCMOS gate array technology has been described in detail together with an experimental input port controller in the same technology. The operation of the switching element has been measured and its throughput at saturation performance shown to agree with that predicted by the simulation model to within 1%. Thus confidence in the simulation results for larger switch structures is increased. Due to design and implementation technology limitations the operating speed of the 4×4 switching element was restricted to 8 MHz. Improvements to the design have been suggested which should enable operation at 50 MHz in 2 μm CMOS without great difficulty. Implementation in higher speed technologies has also been discussed.

Various possible developments to the basic design have been discussed to construct a full-scale switch for use in communications applications. There are some applications that call for a stand-alone switch, such as a high capacity bridge between local and/or metropolitan area networks, but many applications require fast packet switches to be interconnected to form a network. Thus the issues of access control, flow control, high speed protocol and congestion control within a network of fast packet switches become essential areas for further study.

Chapter 9

Conclusion

The design of a fast packet switch has been presented which features a number of original aspects. Its performance has been investigated using a simulation model to gain an insight into the effect of the various design parameters upon switch performance. The performance of the switch for various models of multi-service traffic has also been investigated using the simulation model. Finally, the two fundamental components of the switch have been implemented in gate array technology to gain a detailed understanding of the construction of this design of fast packet switch.

A summary of the work will now be presented followed by a discussion of some of the more significant results. Some comments will be offered on possible areas of application for the switch and some comparisons drawn with other current fast packet switch designs. Finally, thinking of the future work required on this design of fast packet switch, some ideas on multicast switch operation will be considered. The dissertation draws to a close with the discussion of some of the problems remaining to be solved in the networking of fast packet switches.

9.1 Summary

Motivation

As the telecommunications industry continues to expand, two current trends are becoming apparent: the requirement for increased network capacity and the desire to support an increasing number of communications services (voice, video, image, text, etc.) In the public domain the current Integrated Services Digital Network (ISDN) offers integrated access to communications services that are to a large extent supported on separate networks. As the number of communications services on offer increases, so the requirement to support multi-service traffic over a single integrated network will grow in significance. In the private sphere, high speed local area networks are beginning to appear, capable of supporting video services in addition to high speed computer communications etc. Such networks may shortly require interconnection by means of high capacity packet switches both locally and in the wide

area. Furthermore, standards for the definition of metropolitan area networks are reaching agreement which also promise to support multiple services and will require interconnection across high capacity switches.

The precise traffic characteristics of future communications services are at present unknown and will change with time as the networks expand, thus flexibility becomes a key issue in the selection of a suitable switching mechanism. In addition, the majority of communications services exhibit a bursty behaviour, thus the efficient support of bursty traffic is also of considerable significance. The switching mechanism, however, will also have to support services that are sensitive to delay and to the variance of delay across the network. Conventional circuit switching offers an excellent delay performance and high capacity switches but is inflexible and is very inefficient for bursty traffic. Conventional packet switching handles bursty traffic well but has a very poor delay performance and switches of very high capacity are difficult to construct. A hybrid switch that offers both circuit and packet switching is certainly a solution for the near term but a single fully integrated switching mechanism will offer greater flexibility and will be able to adapt more quickly to the changing traffic requirements of new communications services. From a review of the available switching mechanisms, fast packet switching, a statistical switching mechanism, has been selected for further study on the grounds of flexibility, performance and implementation considerations.

Design

There are three basic classes of fast packet switch design: input buffered, output buffered and internally buffered. An input buffered switch is the simplest to construct but offers approximately half of the performance of an output buffered switch. An output buffered switch offers the best theoretical performance but requires at least an order of magnitude more hardware, whereas internally buffered switch designs fall somewhere between input and output buffered designs in terms of both performance and hardware complexity. The majority of existing fast packet switch designs are constructed in VLSI whereas the Cambridge Fast Packet Switch proposes a very simple design capable of implementation in gate array technology. A simple implementation offers flexibility, a wide range of potential applications and operation at both conventional speeds and also at possibly very high speeds. An input buffered design has been selected for simplicity of implementation but various techniques have been investigated that enhance the performance of the basic switch towards that of the output buffered design.

To enable the construction of a very high capacity switch, whilst retaining the simplicity of implementation of the fundamental switching element, the design has been based upon the use of a multi-stage interconnection network for the switch fabric. The delta network has been selected as a reasonable compromise between complexity and performance but to improve the performance, to increase reliability and to remove the sensitivity of the switch to the distribution of the incident traffic the Beneš network has also been investigated. Whereas the majority of previous work on the use of multi-stage interconnection networks has focussed upon the 2×2

switching element this design concentrates on the use of switching elements of up to 16×16. This improves the performance and reduces the number of interconnections required within the switch fabric which forms a major factor limiting the maximum size of the switch implementation. The use of a multi-plane switch fabric has also been proposed in order to improve reliability and performance. Switching elements of high degree and multiple switch planes both introduce multiple equivalent paths between the same input and output ports into the switch fabric. Three algorithms have been suggested in order to select a path across the switch fabric: searching, flooding and random selection. Input queue by-pass is a technique that has been investigated to improve the basic performance of the switch and for a two-plane design, output buffering across the two parallel planes also enhances the performance.

Performance

The influence of the various design parameters on the performance of the switch has been investigated by considering the throughput at saturation of each design using a simulation model. The delay performance for slotted traffic has also been investigated and the results have been compared to the performance of the crossbar network which represents the performance of the ideal input buffered switch.

For delta networks, the use of a two-plane switch fabric is recommended on the grounds of increased throughput, increased reliability and ease of maintenance but the use of more than two switch planes in parallel is not justified from the point of view of increased performance. The use of switching elements of degree 8 or 16 is preferred against those of degree 2 or 4 because of improved throughput performance and reduced interconnections within the switch fabric. For delta networks the searching algorithm offers a performance only slightly lower than that of a flooding algorithm and a hybrid algorithm which searches within each switch plane but floods across the planes is recommended for its ease of implementation. A two-plane regular delta network offers a throughput performance only slightly inferior to that of a crossbar switch fabric. The introduction of input queue by-pass together with output buffering yields a performance comparable to the two plane crossbar switch fabric with output buffering, and only slightly lower than that of the output buffered switch.

In general, a Beneš switch fabric is unlikely to be favoured above a delta network purely on the grounds of its throughput performance. It is of interest because it reduces the sensitivity of the switch to the destination distribution of the incident traffic. The performance of the Beneš switch fabric lies between that of the equivalent delta and crossbar networks. For applications that require a short packet length the random path selection algorithm is recommended whereas a flooding algorithm may yield improved performance for longer packet lengths.

An extension to the design of the switch has been proposed to support two fundamental classes of traffic. Reserved service traffic receives a higher priority in the switch fabric and handles classes of traffic that are sensitive to delay whilst the unreserved service caters for traffic that is less delay sensitive. Simulation results indicate that for a Poisson reserved service traffic loading of up to 80% of the throughput

at saturation of the switch fabric, the upper bound on delay for 99% of all incident reserved service packets is in the region of 20 packet lengths. Further, unreserved service traffic may be multiplexed with reserved service traffic, at every input port of the switch, so as to operate the switch continuously at saturation, without affecting the bounded delay performance of the reserved service. This result has been shown to hold for a wide range of switch structures and switch fabric design parameters. Also the reserved service throughput and delay performance appears insensitive to the arrival distribution and to the destination distribution of unreserved service traffic.

A closer investigation of the delay performance of the switch for voice traffic modelled as a superposition of individual packet voice sources on each switch port, both with and without silence detection, reveals no significant departure from the delay performance of the Poisson model. This traffic model was observed to give a packet arrival distribution closely approximating that of a Poisson source.

For delay sensitive, reserved service traffic performance, the packet length for both reserved and unreserved service traffic should be kept short and constant. No performance impairment is introduced by a $\pm 10\%$ variation in packet length but an exponential distribution of packet lengths causes a loss in throughput performance of the order of 12% for a 64×64 regular two-plane delta network. For a single service implementation, moderately insensitive to delay, variable length packets of any reasonable maximum length may be supported.

A cursory inspection of the buffer overflow probability suggests that an input buffer length of 20 packets is sufficient to offer a packet loss probability of less than 10^{-6} for slotted traffic at a traffic load of 80% of the throughput at saturation for all switch designs.

Implementation

The implementation of a 4×4 self-routing crossbar switching element in 3 μm HCMOS gate array technology has been investigated in detail together with an experimental input port controller in the same technology. The operation of the switching element has been measured and its throughput at saturation shown to agree with that predicted by the simulation model to within 1%. Insight gained from the construction of the 4×4 switching element has allowed estimates to be made of the complexity of switching elements of larger degree. Due to limitations in the available technology the operating speed of the 4×4 switching element was restricted to 8 MHz but improvements to the design have been suggested which should enable operation at 50 MHz in 2 μm CMOS without great difficulty. Implementation in higher speed technologies has also been discussed and various possible developments to the basic design have been considered to construct a full-scale switch for use in communications applications.

9.2 Discussion

The work has demonstrated that it is possible to construct a high capacity fast packet switch from very simple components. The design is easily partitioned both at the gate array and circuit card level and all but the largest of switches should experience few serious implementation difficulties at conventional speeds. The use of switching elements of high degree (8 or 16) has been shown to offer significant advantages in both performance and in the number of interconnections required in the switch fabric over previous designs based upon 2×2 switching elements. The use of input queue by-pass, a two-plane switch fabric, and output buffering across two switch planes, has shown that the basic performance of the switch may be enhanced to approach that of the ideal switch at a cost of increased complexity. The provision of two levels of traffic priority has demonstrated that although the switch is probabilistic in nature, certain statistical guarantees can be made on the delay of high priority traffic across the switch given that the mean load of the high priority traffic is kept below an upper limit. This guarantee has been shown to be unaffected by the load or distribution of the lower priority traffic.

For asynchronous transfer mode applications within broadband ISDN at conventional speeds, the complexity of the switching element is unlikely to be an issue. Modularity, incremental growth, reliability and ease of maintenance will be much more significant. To support a large number of communications services, many priority levels may be desirable within the switch. In addition, to reduce the variance of delay and buffer overload probability at high loads, a switch design which handles packets within each priority level in a strict first in first out manner, across all switch ports, is likely to be preferred.

The design issue of whether to use serial or parallel data paths within the switching elements is dependent upon technology and at conventional speeds both approaches appear possible. At very high speeds serial techniques are likely to be preferred due to their hardware simplicity with the ultimate possibility of implementing the data paths of the switching elements using photonic devices. If the broadband ISDN adopts asynchronous transfer mode and organises the network hierarchically it may be advantageous for the upper layers of the network, corresponding to the current trunk network, to group packets (cells) destined for the same major urban areas together. Thus large trains of packets may be routed through the upper layers of the network without reference to individual packet headers. Switches based upon photonic switching elements might find an area of application within the upper layers of such a network. If this were the case, simplicity of implementation might become a very desirable feature together with the ability to handle variable length 'packet trains' and to offer some statistical guarantee on delay for higher priority traffic.

Switches for applications within the local area, serving as multi-port bridges for LANs, interconnecting high speed LANs, or performing the function of a high speed LAN themselves, will not be required to work continuously at high loads. Also the delay requirements are likely to be much less stringent than those of the public network as the major part of the voice service will probably remain circuit switched for

some time. Thus applications involving video and image services, possibly including local or stored voice, are those most likely to introduce the requirement for high capacity fast packet switching in the local area. In this environment a switch design that does not require investment in VLSI may be attractive. To facilitate further experimental work in this area the Cambridge Fast Packet Switch has been designed to be capable of interfacing to the Cambridge Fast Ring [68, 67] to form a multi-port bridge between a large number of rings.

It has been demonstrated that for delay sensitive traffic, the delay across a fast packet switch may be comparable with that across a circuit switch. The variance of delay, however, may be much greater than that of a circuit switch. For fast packet switches operating at speeds above about 10 MHz, the variance in delay will be much less than the packetisation delay for the voice service. Thus simple techniques are available to deal with it. For the voice service, the greatest delay component due to packet mode operation is the packetisation delay which in most cases exceeds current regulations concerning delay in the local arm of the public telephony network. Thus for the immediate future the voice service is likely to remain circuit switched and fast packet switching may be introduced in the context of a hybrid switch. The packetisation delay, however, is not a fundamental problem and as work progresses on asynchronous transfer mode operation, the regulations may be relaxed to encourage a fully integrated switching mechanism.

The delay measurements presented in this dissertation refer to traffic with random arrival characteristics. It has been shown that a large enough superposition of periodic sources will approximate to a random arrival process given that no external synchronisation is applied to coordinate the individual traffic sources and their destination requests. It is possible to coordinate the arrival characteristics and destination distribution of periodic traffic such that any statistical switch will offer poor performance in terms of delay and packet loss. Such situations, however, are very unlikely to occur naturally and will persist only for a very short duration. It may be necessary to ensure that no coordination of periodic traffic arrival characteristics is present in a system design. Suggestions have been made that the coding of periodic traffic sources be modified to add a measure of variance to the packet departure times to ensure random arrival characteristics at the first switch in the path.

All of the fast packet switch designs reviewed in this dissertation are capable of supporting switch implementations of moderate size operating at conventional speeds. The input buffered Batcher-banyan design suffers from the heavy overhead imposed by the three phase contention resolution algorithm and is very inefficient for short packets. Further work is required on the contention resolution algorithm for this design to become competitive for delay sensitive traffic. The output buffered Batcher-banyan design (Starlite) requires a much larger switch fabric to handle the recirculated packets than other switches of the same size. An electronic implementation of the Knockout switch is limited in speed and requires much more hardware than other designs but offers incremental growth and the possibility of variable length packets. There is little to choose between the designs of internally buffered switch beyond cost and performance considerations. In current technology, designs using buffered

switching elements with parallel internal data paths appear attractive for switches with port bandwidths in the range 100–200 Mbits/sec while very high speed switches with port bandwidths above 500 Mbits/sec favour non-buffered designs using serial data paths in the switch fabric.

The Cambridge Fast Packet Switch may be constructed from simple low cost devices. It may support variable length packets, constant length packets or a range of discrete packet lengths. It is reasonably efficient for very short packets and can support two levels of packet priority with little difficulty. It may be possible to modify the design to support a wide range of packet priorities at the lowest level within the switch fabric. This could be achieved by modifying the arbiter in each switching element to select packets according to a priority field in the header of each packet and operating the switch fabric synchronously at the packet level. The switch, however, does not guarantee strict first in first out operation within a priority level across all switch ports at high loads. These characteristics suggest applications:

- as a high speed local area network;
- as a high capacity multi-port bridge between high speed LANs;
- for multi-service traffic within the local area;
- for the wide area interconnection of multi-service local area networks;
- for the packet switching function within a hybrid switch, e.g. within an integrated services PABX or within a metropolitan area network;
- or possibly for switches operating at very high speeds.

It is also possible to operate the switch as a full-duplex circuit switch and in such a role it may find applications within the field of parallel processing, e.g. to support processor to memory interconnection.

9.3 Multicast Operation

In considering further work on the Cambridge Fast Packet Switch, multicast operation forms perhaps the foremost requirement. A multicast connection is a one-to-many or distributive connection in which a single incoming packet is replicated and each copy routed individually over different outgoing virtual circuits. For reserved service traffic a distributive connection may be required to support audio conferencing or TV distribution, whereas for unreserved service traffic, multicast connections will be required to support the interconnection of groups of local area networks.

The Starlite switch [70] suggests a receiver initiated approach in which the receiver launches empty packets into the switch as required which receive a copy of the relevant incoming multicast packet in a copy fabric. This approach, however, assumes synchronisation between the source and destinations and is therefore not suitable as

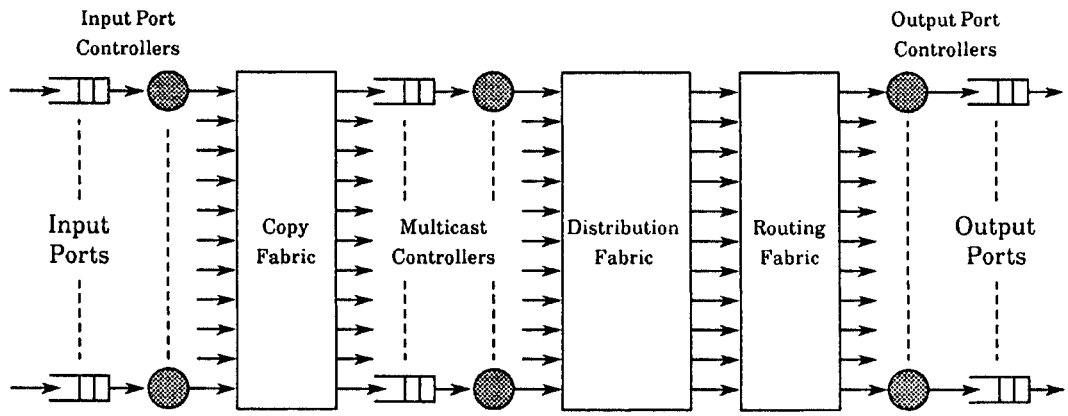

Figure 9.1: Switch structure for multicast operation.

a general method for both reserved and unreserved traffic. The switching element of the Prelude design [141, 31] is able to handle multicast packets directly but the majority of switch designs suggest a structure of the form illustrated in fig. 9.1. The structure of the fast packet switch has been augmented by the addition of a copy fabric and multicast controllers. Multicast packets are replicated in the copy fabric and routed to the outgoing virtual circuits by the multicast controllers using table look-up and manipulation of the label field in the packet header.

Two proposals have been made for the construction of the copy network: a buffered banyan [145, 148, 19] and a running adder network followed by a non-buffered banyan [89, 88]. The non-buffered copy network is also non-blocking and thus its performance is more easily predicted than the buffered banyan network but both approaches can handle multicast traffic at reasonably high loads. Neither solution, however, offers a particularly simple implementation.

For fast packet switches aimed at the interconnection of local area networks, operation under continuous high loads of multicast traffic is unlikely to be a requirement and a simple implementation may offer an advantage. In this environment a possible solution for the construction of the copy fabric is a destination release slotted ring. A tag is prefixed to a multicast packet by the input port controller which defines the number of copies required. It is then launched into the ring on the arrival of the first available empty slot. A copy of the packet is taken at every station that it passes and the tag decremented. When the tag reaches zero the slot is released.

The technique is suitable for implementation in gate array technology and should handle both reserved and unreserved traffic at moderate loads. For low loads of multicast traffic or for multicast traffic that is insensitive to delay it may be possible to implement the copy fabric and the multicast controller function within the input port controller. At higher loads it may be necessary to preface a ring based copy fabric by a single stage distribution fabric to avoid the blocking of a downstream node by a busy upstream node.

9.4 Network Aspects

The design and implementation of a fast packet switch is not an exceedingly difficult task. The Cambridge Fast Packet Switch has demonstrated a switch design capable of very simple implementation but nine other designs have also been reviewed. A simple comparison of the performance of the various switch designs has been offered based upon the throughput at saturation, the mean delay for slotted traffic and the 99th percentile of delay for Poisson traffic. A reasonably self contained topic for further study would be a comparison of the packet loss probability of the various designs of switch against traffic load and buffer length. The integration of both delay sensitive and delay insensitive traffic across a single fast packet switch has been investigated using two levels of priority and the results suggest that acceptable performance characteristics may be attained for the various classes of traffic. The extension of this technique to a wide range of priority levels might be of some interest, but the major problem that remains to be solved is the support of multi-service traffic across a network of fast packet switches.

Amongst the many issues still to be addressed in the networking of fast packet switches are:

- the characterisation of source traffic profiles;
- the determination of the service performance requirements;
- effective policing mechanisms for traffic sources;
- the allocation of reserved service bandwidth;
- access control, flow control and congestion control.

A policing mechanism for traffic sources and a simple approach to the problem of determining the effective bandwidth required by a source is discussed in [5]. Flow control refers to the control of traffic flow across a connection from end to end whereas congestion control is the action taken by the switches to avoid congestion within the network. A range of congestion control algorithms are possible [55] and include:

- discarding packets when switch queues become full,
- discarding local packets in preference to packets that have already travelled some distance across the network,
- the use of choke packets originated by overloaded switches to cause traffic to be throttled back at the source,
- backpressure across individual virtual circuits,
- congestion prevention by the use of bandwidth reservation mechanisms.

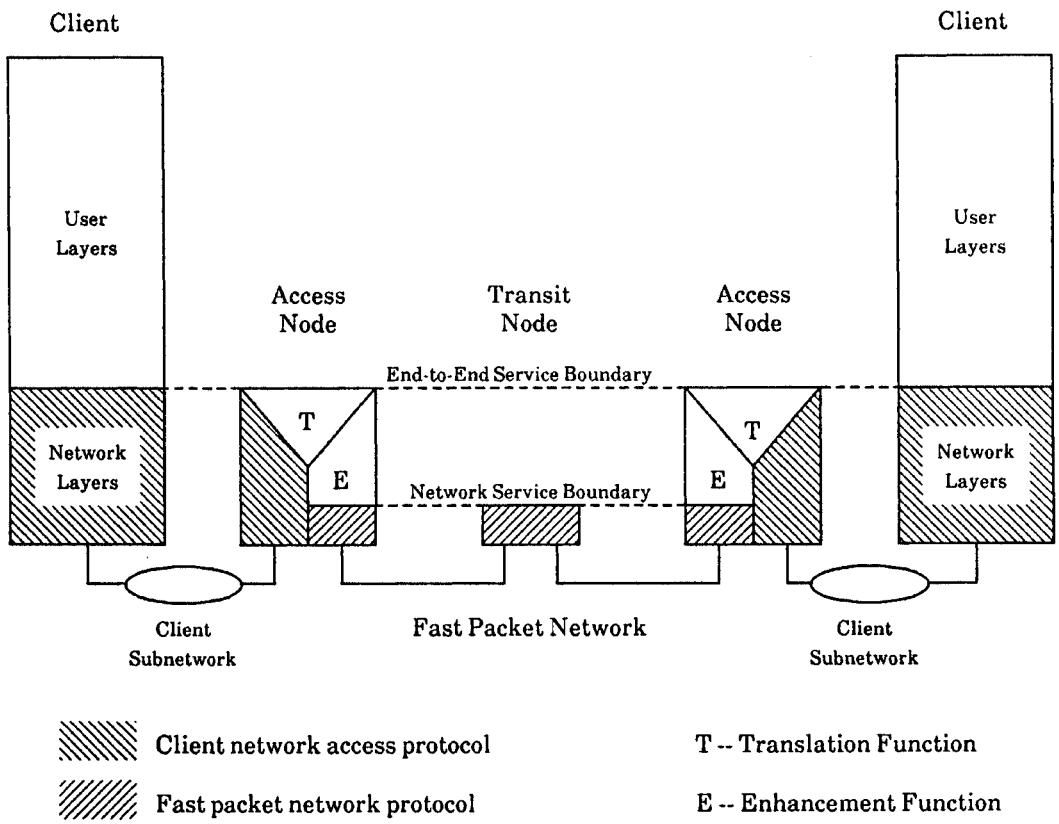

Figure 9.2: General model of protocol structure for a network of fast packet switches.

All of these mechanisms need to be evaluated in the light of the requirements of the different classes of source traffic.

The protocol functions required across a network of fast packet switches constitute another area for further study. Delay sensitive services require a lightweight protocol structure but a network of fast packet switches will be expected to offer interconnection according to the various networking access standards. A general model of the protocol structure likely to be considered for use within a network of fast packet switches is illustrated in fig. 9.2. Access to the fast packet network is controlled by the access node which connects the client across the client subnet to the network according to an established standard. This standard defines the end-to-end service required across the network. This service is supported by taking the fundamental service offered by the fast packet network and extending it by means of the enhancement and translation functions in the access node. Thus by selecting appropriate enhancement and translation functions the various classes of traffic and communications standards may be accommodated. The selection of the fundamental service offered by the network is thus of considerable significance.

Appendix A

Simulation Results for Throughput at Saturation

Crossbar Networks

The simulation results for the throughput at saturation of the various switch fabrics are presented in this appendix to enable the comparison of the different switch structures. Table A.1 presents the results for the crossbar switch fabric with blocked packets resubmitted both for synchronous operation and for asynchronous operation with a retry delay of 10% of the packet length. The synchronous and asynchronous results for the regular structure (pure input buffered with no queue by-pass or output buffering) are plotted in fig. 6.1.

Size	Synchronous		Asynchronous (10% retry delay)			
	Regular	Double Buffer	Regular	Queue By-Pass	Double Buffer	De-Luxe
2	.747	1.0	.742	.878	1.0	1.0
4	.650	.941	.640	.808	.931	.981
8	.613	.916	.601	.771	.897	.971
16	.594	.905	.583	.752	.883	.965
32	.586	.900	.575	.746	.876	.961
64	.585	.898	.572	.740	.872	.958
128	.585	.896	.572	.740	.872	.957
256	.584	.896	.572	.739	.872	.954
512+	.585	.896	.572	.739	.872	.951

Table A.1: Throughput at saturation for crossbar switch fabrics.

Delta Networks with a Searching Algorithm

Tables A.2 to A.5 give the throughput at saturation of delta networks constructed from switching elements of degree 2 to degree 16 with a searching algorithm and a retry delay of 10% of the packet length. These results derive from the simple model, the accuracy of which is presented in tables 6.2 and 6.3.

Size	Single Plane		Two-Plane			
	Regular	Queue By-Pass	Regular	Queue By-Pass	Double Buffer	De-Luxe
2	.742	.878	.740	.879	.975	.971
4	.597	.791	.639	.799	.895	.930
8	.517	.711	.598	.742	.817	.885
16	.462	.642	.566	.705	.755	.839
32	.423	.583	.547	.672	.703	.795
64	.392	.536	.532	.648	.662	.756
128	.366	.496	.515	.622	.624	.719
256	.344	.463	.502	.599	.593	.684
512	.325	.435	.484	.577	.564	.654

Table A.2: Throughput at saturation of delta networks with switching elements of degree 2 for a searching algorithm.

Size	Single Plane		Two-Plane			
	Regular	Queue By-Pass	Regular	Queue By-Pass	Double Buffer	De-Luxe
4	.640	.808	.645	.813	.915	.945
8	.575	.693	.599	.743	.847	.884
16	.514	.684	.575	.726	.805	.879
32	.502	.599	.563	.685	.771	.816
64	.446	.596	.551	.682	.725	.814
128	.453	.530	.549	.646	.714	.755
256	.397	.530	.531	.645	.663	.755
512	.413	.478	.533	.611	.665	.703

Table A.3: Throughput at saturation of delta networks with switching elements of degree 4 for a searching algorithm.

Size	Single Plane		Two-Plane			
	Regular	Queue By-Pass	Regular	Queue By-Pass	Double Buffer	De-Luxe
8	.601	.771	.603	.775	.879	.931
16	.553	.652	.576	.719	.816	.861
32	.535	.649	.572	.712	.808	.857
64	.487	.647	.561	.710	.772	.857
128	.498	.560	.555	.667	.750	.786
256	.474	.559	.554	.665	.739	.786
512	.421	.558	.539	.664	.691	.786
1024	.452	.493	.543	.619	.694	.719
2048	.424	.493	.536	.619	.678	.719
4096	.371	–	–	–	–	–

Table A.4: Throughput at saturation of delta networks with switching elements of degree 8 for a searching algorithm.

Size	Single Plane		Two-Plane			
	Regular	Queue By-Pass	Regular	Queue By-Pass	Double Buffer	De-Luxe
16	.583	.752	.583	.754	.862	.921
32	.542	.635	.570	.704	.799	.845
64	.536	.632	.564	.698	.794	.843
128	.519	.632	.562	.700	.788	.843
256	.477	.630	.556	.697	.756	.842
512	.493	.544	.553	.653	.733	.769
1024	.485	.543	.553	.650	.734	.768
2048	.459	.542	.548	.649	.721	.767
4096	.409	.541	.533	–	.674	–

Table A.5: Throughput at saturation of delta networks with switching elements of degree 16 for a searching algorithm.

Delta Networks with a Flood-Planes Algorithm

Tables A.6 to A.9 give the throughput at saturation of two-plane delta networks constructed from switching elements of degree 2 to degree 16 with a flood-planes algorithm and a retry delay of 10% of the packet length. The results for the two-plane regular structure with a flood-planes algorithm are plotted in fig. 6.7. The flood-planes algorithm selects between multiple paths to the same destination by flooding across all planes in parallel but searching within a plane thus the results for a single plane structure are equivalent to the searching algorithm.

Size	Regular	Queue By-Pass	Double Buffer	De-Luxe
2	.742	.878	1.0	1.0
4	.648	.813	.930	.979
8	.599	.775	.850	.950
16	.576	.743	.786	.916
32	.560	.723	.733	.879
64	.543	.702	.688	.842
128	.529	.682	.650	.807
256	.514	.662	.618	.773
512	.499	.642	.589	.741

Table A.6: Throughput at saturation of delta networks with switching elements of degree 2 for a flood-planes algorithm.

Size	Regular	Queue By-Pass	Double Buffer	De-Luxe
4	.640	.808	.931	.981
8	.605	.767	.887	.947
16	.576	.752	.836	.942
32	.575	.725	.818	.895
64	.560	.724	.755	.892
128	.566	.697	.762	.842
256	.541	.696	.690	.840
512	.551	.668	.714	.791

Table A.7: Throughput at saturation of delta networks with switching elements of degree 4 for a flood-planes algorithm.

Size	Regular	Queue By-Pass	Double Buffer	De-Luxe
8	.601	.771	.897	.971
16	.581	.741	.861	.927
32	.575	.738	.848	.924
64	.566	.731	.799	.922
128	.568	.707	.806	.865
256	.563	.705	.783	.864
512	.548	.706	.718	.863
1024	.560	.674	.759	.804
2048	.553	.674	.726	.804
4096	.526	.674	.652	.801

Table A.8: Throughput at saturation of delta networks with switching elements of degree 8 for a flood-planes algorithm.

Size	Regular	Queue By-Pass	Double Buffer	De-Luxe
16	.583	.752	.883	.965
32	.577	.728	.849	.915
64	.573	.728	.842	.913
128	.573	.725	.829	.913
256	.562	.725	.786	.912
512	.566	.698	.797	.851
1024	.567	.695	.794	.847
2048	.563	.695	.769	.847
4096	.543	–	.703	.850

Table A.9: Throughput at saturation of delta networks with switching elements of degree 16 for a flood-planes algorithm.

Sub-Equipped Beneš Networks

Tables A.10 to A.13 present the throughput at saturation for the sub-equipped Beneš structure constructed from switching elements of degree 2 to 16. A retry delay of 10% of the packet length was used and results are given for both random and flooding algorithms with and without input queue by-pass. The results for the regular sub-equipped Beneš structure with a flooding algorithm are plotted in fig. 6.9(b).

Size	Regular		Queue By-Pass	
	Random	Flooding	Random	Flooding
2	.742	.742	.878	.878
4	.615	.634	.791	.804
8	.562	.579	.712	.741
16	.520	.548	.642	.697
32	.489	.532	.583	.662
64	.463	.521	.534	.636
128	.440	.514	.495	.614
256	.419	.508	.462	.598
512	.400	.503	.433	.584

Table A.10: Throughput at saturation of sub-equipped Beneš networks with switching elements of degree 2.

Size	Regular		Queue By-Pass	
	Random	Flooding	Random	Flooding
4	.640	.640	.808	.808
8	.582	.598	.734	.760
16	.551	.572	.686	.724
32	.524	.560	.629	.701
64	.510	.554	.596	.682
128	.485	.550	.551	.668
256	.473	.548	.529	.656
512	.450	.545	.494	.647

Table A.11: Throughput at saturation of sub-equipped Beneš networks with switching elements of degree 4.

Size	Regular		Queue By-Pass	
	Random	Flooding	Random	Flooding
8	.601	.601	.771	.771
16	.566	.582	.707	.747
32	.545	.572	.666	.728
64	.538	.567	.648	.715
128	.515	.567	.597	.708
256	.503	.565	.571	.698
512	.496	.565	.559	.693

Table A.12: Throughput at saturation of sub-equipped Beneš networks with switching elements of degree 8.

Size	Regular		Queue By-Pass	
	Random	Flooding	Random	Flooding
16	.583	.583	.752	.752
32	.557	.575	.695	.744
64	.544	.571	.657	.735
128	.538	.571	.640	.728
256	.535	.571	.632	.725
512	.513	.571	.584	.722

Table A.13: Throughput at saturation of sub-equipped Beneš networks with switching elements of degree 16.

References

[1] G B Adams and H J Siegal. The Extra Stage Cube: A fault tolerant interconnection network for super systems. *IEEE Trans. Computers*, **C-31** (5), 443–454, May 1982.

[2] J L Adams. The Orwell Torus communications switch. In *Proc. CEPT Seminar on Broadband Switching*, pages 215–223, Albufeira, Portugal, Jan. 1987.

[3] S Ades, R Want and R Calnan. Protocols for real time voice communication on a packet local network. In *Proc. IEEE Int. Conf. Commun. (ICC '86)*, Toronto, June 1986.

[4] H Ahmadi, W E Denzel, C A Murphy and E Port. A high-performance switch fabric for integrated circuit and packet switching. In *Proc. IEEE Infocom*, pages 9–18, New Orleans, Mar. 1988.

[5] S Akhtar. *Congestion control in a fast packet switching network*. Master's thesis, Washington Univ., St. Louis, Missouri, Dec. 1987.

[6] S R Amstutz. Burst switching — An introduction. *IEEE Commun. Mag.*, **21**, 36–42, Nov. 1983.

[7] J M Appleton and M M Peterson. Traffic analysis of a token ring PBX. *IEEE Trans. Commun.*, **COM-34** (5), 417–422, May 1986.

[8] H Armbruster. Applications of future broad-band services in the office and home. *IEEE J. Select. Areas in Commun.*, **SAC-4** (4), 429–437, July 1986.

[9] H Armbruster and G Arndt. Broadband communication and its realisation with broadband ISDN. *IEEE Commun. Mag.*, **25** (11), 8–19, Nov. 1987.

[10] F Backes. Transparent bridges for interconnection of IEEE 802 LANs. *IEEE Network Mag.*, **2** (1), 5–9, Jan. 1988.

[11] G Barberis and D Pazzaglia. Analysis and optimal design of a packet voice receiver. *IEEE Trans. Commun.*, **COM-28** (2), 217–227, Feb. 1980.

[12] K E Batcher. Sorting networks and their applications. In *Proc. Spring Joint Computer Conf.*, pages 307–314, 1968.

[13] V E Beneš. On rearrangeable three-stage connecting networks. *Bell Systems Tech. J.*, **41** (5), 1481–1492, Sept. 1962.

[14] R W Blackmore, W J Stewart and I Bennion. An opto-electronic exchange for the future. In *Proc. IEEE Int. Switching Symp. (ISS '84)*, pages 41A4.1–7, Florence, May 1984.

[15] R J Boehm, Y C Ching and R C Sherman. SONET (Synchronous optical network). In *Proc. IEEE Globecom*, pages 1443–1450, New Orleans, 1985.

[16] A Boleda and D Lasker. The architecture of Meridian SL integrated services networks. *Telesis two, (Bell Northern Research)*, 27–33, 1985.

[17] P T Brady. A statistical analysis of on-off patterns in 16 conversations. *Bell Systems Tech. J.*, **47**, 73–91, Jan. 1968.

[18] J R Brandsma. PHILAN: A fibre-optic ring for voice and data. *IEEE Commun. Mag.*, **24** (12), 16–22, Dec. 1986.

[19] R G Bubenik and J S Turner. Performance of a broadcast packet switch. In *Proc. IEEE Int. Conf. Commun. (ICC '87)*, pages 1118–1122, June 1987.

[20] K Bullington and J M Fraser. Engineering aspects of TASI. *Bell Systems Tech. J.*, **38**, 353–364, March 1959.

[21] J B Butcher and K K Johnstone. Wafer Scale Integration. *Proc. IEE pt. E*, **135** (6), 281–288, Nov. 1988.

[22] P Bylanski and D G W Ingram. *Digital transmission systems.* Peter Peregrinus for IEE, 1976.

[23] L M Casey, R C Dittburner and N D Gamage. FXNET: A backbone ring for voice and data. *IEEE Commun. Mag.*, **24** (12), 23–28, Dec. 1986.

[24] P Y Chen, P C Yew and D Lawrie. Performance of packet switching in buffered single-stage shuffle-exchange networks. In *Proc. 3rd Int. Conf. on Distributed Computing Systems*, pages 622–627, 1982.

[25] T M Chen and D G Messerschmitt. Integrated voice/data switching. *IEEE Commun. Mag.*, **26** (6), 16–26, June 1988.

[26] G L Chesson and A G Fraser. Datakit network architecture. In *Proc. IEEE Compcon Spring*, pages 59–61, 1980.

[27] W Chou. *Computer communications: Volume II systems and applications.* Prentice-Hall, NJ, 1985.

[28] I Cidon, I S Gopal and H Heleis. PARIS: An approach to integrated private networks. In *Proc. IEEE Int. Conf. Commun. (ICC '87)*, pages 764–768, Seattle, June 1987.

[29] G Clapp, S Karr and M Singh. MAN architecture and services. AT&T submission to IEEE 802.6, Nov. 1987.

[30] C Clos. A study of non-blocking switching networks. *Bell Systems Tech. J.*, **32**, 406–424, March 1953.

[31] J P Coudreuse and M Servel. Prelude: An asynchronous time-division switched network. In *Proc. IEEE Int. Conf. Commun. (ICC '87)*, pages 769–773, Seattle, June 1987.

[32] G J Coviello and P A Vena. Integration of circuit/packet switching by a SENET (slotted envelope network) concept. In *Proc. Nat. Telecommun. Conf.*, pages 42.12–17, Dec. 1975.

[33] J N Daigle and J D Langford. Models for analysis of packet voice communications systems. *IEEE J. Select. Areas Commun.*, **SAC-4** (6), 847–855, Sept. 1986.

[34] C Day, J Giacopelli and J Hickey. Applications of self-routing switches to LATA fiber optic networks. In *Proc. IEEE Int. Switching Symp. (ISS '87)*, pages 519–523, Mar. 1987.

[35] M De Prycker and J Bauwens. A switching exchange for an asynchronous time division based network. In *Proc. IEEE Int. Conf. Commun. (ICC '87)*, pages 774–781, Seattle, June 1987.

[36] M De Prycker and M De Somer. Performance of an independent switching network with distributed control. *IEEE J. Select. Areas Commun.*, **SAC-5** (8), 1293–1301, Oct. 1987.

[37] M Decina. Broadband ISDN, whither ATM? In *IBM European Telecommun. Workshop*, Montpellier, France, Sept. 1988.

[38] J D DeTreville. A simulation based comparison of voice transmission on CSMA/CD networks and on token busses. *Bell Systems Tech. J.*, **63** (1), 33–55, Jan. 1984.

[39] J D DeTreville and W D Sincoskie. A distributed experimental communications system. *IEEE J. Select. Areas in Commun.*, **SAC-1** (6), 1070–1075, Dec. 1983.

[40] D M Dias and J R Jump. Analysis and simulation of buffered delta networks. *IEEE Trans. Computers*, **C-30** (4), 273–282, Apr. 1981.

[41] D M Dias and J R Jump. Packet switching interconnection networks for modular systems. *IEEE Computer Mag.*, **14** (12), 43–53, Dec. 1981.

[42] C Ellis. Voice and data integration – The PABX as a local area network. In *Proc. Networks '84*, pages 445–450, Online, London, July 1984.

[43] K Y Eng. A photonic Knockout switch for high-speed packet networks. *IEEE J. Select. Areas in Commun.*, **SAC-6** (7), 1107–1116, Aug. 1988.

[44] K Y Eng, M G Hluchyj and Y S Yeh. A Knockout switch for variable-length packets. *IEEE J. Select. Areas Commun.*, **SAC-5** (9), 1426–1435, Dec. 1987.

[45] K Y Eng, M G Hluchyj and Y S Yeh. Multicast and broadcast services in a Knockout packet switch. In *Proc. IEEE Infocom*, pages 29–34, New Orleans, Mar. 1988.

[46] R M Falconer and J L Adams. Orwell: A protocol for an integrated servces local network. *British Telecom Tech. J.*, **3** (4), 27–35, Oct. 1985.

[47] T Feng. A survey of interconnection networks. *IEEE Computer Mag.*, **14** (12), 12–27, Dec. 1981.

[48] M J Fischer and T C Harris. A model for evaluating the performance of an integrated circuit and packet switched multiplex structure. *IEEE Trans. Commun.*, **COM-24** (2), 195–202, Feb. 1976.

[49] J W Forgie. Speech transmission in store and forward networks. In *National Computer Conference, vol. 44*, 1975.

[50] J W Forgie and A G Nemeth. An efficient packetized voice/data network using statistical flow control. In *Proc. IEEE Int. Conf. Commun. (ICC '77)*, pages 44–48, 1977.

[51] G Foster and J L Adams. The ATM zone concept. In *Proc. IEEE Globecom*, 1988.

[52] A G Fraser. Datakit — modular network for synchronous and asynchronous traffic. In *Proc. Int. Conf. Commun. (ICC '79)*, pages 20.1.1–3, June 1979.

[53] I D Gallagher. Multi-service networks. *British Telecom Tech. J.*, **4** (1), 43–49, Jan. 1986.

[54] I D Gallagher. A multi-service network based on the Orwell protocol. In *Proc. IEEE Int. Switching Symp. (ISS '87)*, pages 979–983, Mar. 1987.

[55] M Gerla and L Kleinrock. Congestion control in interconnected LANs. *IEEE Network Mag.*, **2** (1), 72–76, Jan. 1988.

[56] S Giorcelli et al. Experimenting with fast packet switching techniques in first generation ISDN environments. In *Proc. IEEE Int. Switching Symp. (ISS '87)*, pages 388–394, Mar. 1987.

[57] L R Goke and G J Lipovski. Banyan networks for partitioning multiprocessor systems. In *Proc. First Annual Symp. Computer Architecture*, pages 21–28, Dec. 1973.

[58] J Gruber and L Strawczynski. Judging speech in dynamically managed voice systems. *Telesis two, (Bell Northern Research)*, 30–34, 1983.

[59] J G Gruber. Delay related issues in integrated voice and data networks. *IEEE Trans. Commun.*, **COM-29** (6), 786–800, June 1981.

[60] J G Gruber and N Le. Performance requirements for integrated voice/data networks. *IEEE J. Select. Areas in Commun.*, **SAC-1** (6), 981–1005, Dec. 1983.

[61] E F Haselton. A PCM frame switching concept leading to burst switching network architecture. *IEEE Commun. Mag.*, **21** (6), 13–19, Sept. 1983.

[62] M Hatamian and E G Bowen. Homenet: A broadband voice/data/video network on CATV systems. *Bell Systems Tech. J.*, **64** (2), 347–367, Feb. 1985.

[63] H Heffes and D M Lucantoni. A Marcov modulated characterisation of packetized voice and data traffic and related statistical multiplexer performance. *IEEE J. Select. Areas Commun.*, **SAC-4** (6), 856–867, Sept. 1986.

[64] A M Hill. Switching and distribution networks for wideband optical signals. In *Proc. IEEE Int. Switching Symp. (ISS '87)*, pages 681–688, Mar. 1987.

[65] M G Hluchyj and M J Karol. Queueing in space division packet switching. In *Proc. IEEE Infocom*, pages 334–343, New Orleans, March 1988.

[66] W L Hoberecht. A layered network protocol for voice and data integration. *IEEE J. Select. Areas in Commun.*, **SAC-1** (6), 1006–1013, Dec. 1983.

[67] A Hopper and R M Needham. *The Cambridge Fast Ring networking system (CFR).* Technical Report No. 90, Computer Laboratory, University of Cambridge, June 1986.

[68] A Hopper and R M Needham. The Cambridge Fast Ring Networking System. *IEEE Trans. Computers*, **37** (10), 1214–1223, Oct. 1988.

[69] A Hopper and D J Wheeler. Binary Routing Networks. *IEEE Trans. Computers*, **C-28** (10), 699–703, Oct. 1979.

[70] A Huang and S Knauer. Starlite: A wideband digital switch. In *Proc. IEEE Globecom*, pages 121–125, Nov. 1984.

[71] J Y Hui and E Arthurs. A broadband packet switch for integrated transport. *IEEE J. Select. Areas Commun.*, **SAC-5** (8), 1264–1273, Oct. 1987.

[72] D Hutchison. *Local area network architecture.* Addison-Wesley, 1988.

[73] M Ilyas and H T Mouftah. Quasi cut-through: New hybrid switching technique for computer communication networks. *Proc. IEE Pt. E*, **131** (1), 1–9, Jan. 1984.

[74] Y Jenq. Performance analysis of a packet switch based on a single-buffered banyan network. *IEEE J. Select. Areas Commun.*, **SAC-1** (6), 1014–1021, Dec. 1983.

[75] M J Karol and M G Hluchyj. Using a packet switch for circuit switched traffic: a queueing system with periodic input traffic. In *Proc. IEEE Int. Conf. Commun. (ICC '87)*, pages 1677–1682, Seattle, June 1987.

[76] M J Karol, M G Hluchyj and S P Morgan. Input versus output queueing on a space-division packet switch. *IEEE Trans. Commun.*, **COM-35** (12), 1347–1356, Dec. 1987.

[77] P Kermani and L Kleinrock. Virtual cut-through: A new computer communications switching technique. *Computer Networks*, **3**, 267–286, Sept. 1979.

[78] B G Kim. Characterisation of arrival statistics of multiplexed voice packets. *IEEE J. Select. Areas Commun.*, **SAC-1** (6), 1133–1139, Dec. 1983.

[79] P Kirton, J Ellershaw and M Littlewood. Fast packet switching for integrated network evolution. In *Proc. IEEE Int. Switching Symp. (ISS '87)*, pages B.6.2.1–7, Mar. 1987.

[80] A Kitamura et al. High speed and high capacity packet switching system architecture for ISDN. In *Proc. IEEE Int. Switching Symp. (ISS '87)*, pages 809–813, Mar. 1987.

[81] R W Klessig. Overview of metropolitan area networks. *IEEE Commun. Mag.*, **24** (1), 9–15, Jan. 1986.

[82] C P Kruskal and M Snir. The performance of multistage interconnection networks for multiprocessors. *IEEE Trans. Computers*, **C-32** (12), 1091–1098, Dec. 1983.

[83] P J Kuehn. Fast packet switching. In *IBM European Telecommun. Workshop*, Montpellier, France, Sept. 1988.

[84] J J Kulzer and W A Montgomery. Statistical switching architectures for future services. In *Proc. IEEE Int. Switching Symp. (ISS '84)*, pages 43.A.1–6, Florence, May 1984.

[85] M Kumar and J R Jump. Performance of unbuffered shuffle-exchange networks. *IEEE Trans. Computers*, **C-35** (6), 573–578, June 1986.

[86] D H Lawrie. Access and alignment of data in an array processor. *IEEE Trans. Computers*, **C-24** (12), 1145–1155, Dec. 1975.

[87] C A Lea. The load sharing banyan network. *IEEE Trans. Computers*, **C-35** (12), 1025–1034, Dec. 1986.

[88] T T Lee. Non-blocking copy networks for multicast packet switching. In *Proc. IEEE Int. Zurich Seminar on Digital Commun.*, pages 221–229, Mar. 1988.

[89] T T Lee, R Boorstyn and E Arthurs. The architecture of a multicast broadband packet switch. In *Proc. IEEE Infocom*, pages 1–8, New Orleans, Mar. 1988.

[90] J O Limb. Performance of local area networks at high speed. *IEEE Commun. Mag.*, **22** (8), 41–45, Aug. 1984.

[91] L R Linnell. A wide-band local access system using emerging-technology components. *IEEE J. Select. Areas Commun.*, **SAC-4** (4), 612–618, July 1986.

[92] M Littlewood, I D Gallagher and J L Adams. Evolution toward an ATD multiservice network. *British Telecom Tech. J.*, **5** (2), 52–62, April 1987.

[93] B Maglaris and M Schwartz. Performance evaluation of a variable frame multiplexer for integrated switched networks. *IEEE Trans. Commun.*, **COM-29** (6), 800–807, June 1981.

[94] J W Mark and J O Limb. Integrated voice/data services on Fasnet. *Bell Labs. Tech. J.*, **63** (2), 307–336, Feb. 1984.

[95] G M Masson et al. A sampler of circuit switching networks. *IEEE Computer Mag.*, **12** (6), 32–48, June 1979.

[96] P W Matthewson and S R Wilbur. An integrated services switching system based upon a single-buffered banyan. In *Proc. IEEE Infocom*, pages 766–772, Mar. 1986.

[97] N F Maxemchuck and A N Netravali. Voice and data on a CATV network. *IEEE J. Select. Areas Commun.*, **SAC-3** (2), 300–311, Mar. 1985.

[98] N F Maxemchuk. Regular mesh topologies in local and metropolitan area networks. *AT&T Tech. J.*, **64** (7), 1659–1685, Sept. 1985.

[99] R J McMillen. A survey of interconnection networks. In *Proc. IEEE Globecom*, pages 105–113, Nov. 1984.

[100] D R Milway. *Binary routing networks.* Technical Report No. 101, Computer Laboratory, University of Cambridge, Dec. 1986.

[101] S E Minzer. Broadband user-network interfaces to ISDN. In *Proc. IEEE Int. Conf. Commun. (ICC '87)*, pages 11.2.1–6, Seattle, June 1987.

[102] R H Moffett. Echo and delay problems in some digital communications systems. *IEEE Commun. Mag.*, **25** (8), 41–47, Aug. 1987.

[103] J F Mollenauer. Standards for metropolitan area networks. *IEEE Commun. Mag.*, **26** (4), 15–19, Apr. 1988.

[104] W A Montgomery. Techniques for packet voice synchronisation. *IEEE J. Select. Areas in Commun.*, **SAC-1** (6), 1022–1028, Dec. 1983.

[105] P J Mountain. The design and application of wideband switches. *British Telecom Tech. J.*, **1** (1), 73–81, July 1983.

[106] J M Musser et al. A local area network as a telephone local subscriber loop. *IEEE J. Select. Areas Commun.*, **SAC-1** (6), 1046–1053, Dec. 1983.

[107] M J Narasimha. The Batcher-banyan self-routing network: universality and simplification. *IEEE Trans. Commun.*, **36** (10), 1175–1178, Oct. 1988.

[108] W E Naylor and L Kleinrock. Stream traffic communication in packet switched networks: Destination buffering considerations. *IEEE Trans. Commun.*, **COM-30** (12), 2527–2535, Dec. 1982.

[109] R M Needham and A J Herbert. *The Cambridge Distributed Computing System.* Addison-Wesley, London, 1982.

[110] P Newman. An investigation of high-speed data communications network techniques. Memorandum TRL/943, The GEC Hirst Research Centre, Nov. 1981.

[111] P Newman. Message switching: A flexible approach to communications network design. Memorandum TRL/1001, The GEC Hirst Research Centre, July 1982.

[112] P Newman. Data Signal Switching Systems. UK Patent GB 2 151 880 B, Dec. 1983.

[113] P Newman. Message switching: An experimental model. Unpublished manuscript, The GEC Hirst Research Centre, Apr. 1983.

[114] P Newman. Self-routing switching element for an asynchronous time switch. UK Patent Application 8824058.5, Oct. 1987.

[115] P Newman. A broad-band packet switch for multi-service communications. In *IBM European Telecommun. Workshop*, Montpellier, France, Sept. 1988.

[116] P Newman. A broad-band packet switch for multi-service communications. In *Proc. IEEE Infocom*, pages 19–28, New Orleans, Mar. 1988.

[117] P Newman. *A fast packet switch for the integrated services backbone network.* Technical Report No. 142, Computer Laboratory, University of Cambridge, July 1988.

[118] P Newman. A fast packet switch for the integrated services backbone network. *IEEE J. Select. Areas in Commun.*, **SAC-6** (9), Dec. 1988.

[119] R M Newman, Z L Budrikis and J L Hullett. The QPSX MAN. *IEEE Commun. Mag.*, **26** (4), 20–28, Apr. 1988.

[120] S Nojima et al. Integrated services packet network using bus matrix switch. *IEEE J. Select. Areas Commun.*, **SAC-5** (8), 1284–1292, Oct. 1987.

[121] G Nutt and D Bayer. Performance of CSMA/CD networks under combined voice and data loads. *IEEE Trans. Commun.*, **COM-30** (1), 6–11, Jan. 1982.

[122] P O'Reilly. Burst and fast packet switching: performance comparisons. In *Proc. IEEE Infocom*, pages 653–666, 1986.

[123] P O'Reilly and S Ghani. Data performance in burst switching when the voice silence periods have a hyperexponential distribution. In *Proc. Int. Conf. Commun. (ICC'86)*, pages 537–542, Toronto, June 1986.

[124] K Padmanabhan and D H Lawrie. A class of redundant path multistage interconnection networks. *IEEE Trans. Computers*, **C-32** (12), 1099–1108, Dec. 1983.

[125] S N Pandhi. The universal data connection. *IEEE Spectrum*, 31–37, July 1987.

[126] J H Patel. Performance of processor-memory interconnections for multiprocessors. *IEEE Trans. Computers*, **C-30** (10), 771–780, Oct. 1981.

[127] S D Personick and W O Fleckenstein. Communications switching — from operators to photonics. *Proc. IEEE*, **75** (10), 1380–1403, Oct. 1987.

[128] G Perucca. Research on advanced switching techniques for the evolution to ISDN and broadband ISDN. *IEEE J. Select Areas Commun.*, **SAC-5** (8), 1356–1364, Oct. 1987.

[129] F E Ross. FDDI — A tutorial. *IEEE Commun. Mag.*, **4** (5), 10–17, May 1986.

[130] A Schill and M Zieher. Performance analysis of the FDDI 100 Mbit/sec optical token ring. In *IFIP WG6.4 Workshop High Speed LANs*, pages 57–78, Feb. 1987.

[131] H J Siegel. *Interconnection networks for large-scale parallel processing.* Lexington Books, 1985.

[132] H J Siegel, R J McMillen and P T Mueller. A survey of interconnection methods for reconfigurable parallel processing systems. In *Proc. Nat. Computer Conf., AFIPS*, pages 529–542, 1979.

[133] W D Sincoskie and C J Cotton. Extended bridge algorithms for large networks. *IEEE Network Mag.*, **2** (1), 16–23, Jan. 1988.

[134] R A Spanke. Architectures for guided-wave optical space switching systems. *IEEE Commun. Mag.*, **25** (5), 42–48, May 1987.

[135] D R Spears. Broadband ISDN switching capabilities from a services perspective. *IEEE J. Select. Areas Commun.*, **SAC-5** (8), 1222–1230, Oct. 1987.

[136] K Sriram and W Whitt. Characterising superposition arrival processes in packet multiplexers for voice and data. *IEEE J. Select. Areas Commun.*, **SAC-4** (6), 833–846, Sept. 1986.

[137] H S Stone. Parallel processing with the perfect shuffle. *IEEE Trans. Computers*, **C-20** (2), 153–161, Feb. 1971.

[138] D T W Sze. A metropolitan area network. *IEEE J. Select. Areas Commun.*, **SAC-3** (6), 815–824, Nov. 1985.

[139] T Takeuchi et al. Synchronous composite packet switching — a switching architecture for broadband ISDN. *IEEE J. Select. Areas Commun.*, **SAC-5** (8), 1365–1376, Oct. 1987.

[140] A S Tanenbaum. *Computer networks.* Prentice-Hall, NJ, 1981.

[141] A Thomas, J P Coudreuse and M Servel. Asynchronous time-division techniques: An experimental packet network integrating video communications. In *Proc. Int. Switching Symp. (ISS '84)*, pages 32C.2.1–7, Florence, May 1984.

[142] R H Thomas et al. Diamond: A multimedia message system built on a distributed architecture. *IEEE Computer Mag.*, **18** (12), 65–78, Dec. 1985.

[143] K J Thurber. Interconnection networks—A survey and assessment. In *Proc. Nat. Computer Conf., AFIPS*, pages 909–919, 1974.

[144] F A Tobagi, F Borgonovo and L Fratta. Expressnet: A high performance integrated-services local area network. *IEEE J. Select. Areas in Commun.*, **SAC-1** (5), 898–913, Nov. 1983.

[145] J S Turner. Design of a broadcast packet network. In *Proc. IEEE Infocom*, pages 667–675, 1986.

[146] J S Turner. Design of an integrated services *packet* network. *IEEE J. Select. Areas in Commun.*, **SAC-4** (8), 1373–1380, Nov. 1986.

[147] J S Turner. New directions in communications (or which way to the information age). In *Proc. IEEE Int. Zurich Seminar on Digital Commun.*, pages 25–33, March 1986.

[148] J S Turner. Design of a broadcast packet switching network. *IEEE Trans. Commun.*, **36** (6), 734–743, June 1988.

[149] J S Turner and L F Wyatt. A packet network architecture for integrated services. In *Proc. IEEE Globecom*, pages 45–50, Dec. 1983.

[150] J von Baardewijk. An experimental all-in-one multi-service broadband switch. In *Proc. IEEE Int. Switching Symp. (ISS '87)*, pages 779–783, Mar. 1987.

[151] J F Wakerley. A voice/data/packet switching architecture. In *Proc. Compcon Spring*, pages 194–199, San Francisco, Feb. 1985.

[152] R Want. *Reliable management of voice in a distributed system.* Technical Report No. 141, Computer Laboratory, University of Cambridge, July 1988.

[153] C J Weinstein. Fractional speech loss and talker activity model for TASI and for packet switched speech. *IEEE Trans. Commun.*, **COM-26** (8), 1253–1257, Aug. 1978.

[154] C J Weinstein and J W Forgie. Experience with speech communication in packet networks. *IEEE J. Select. Areas in Commun.*, **SAC-1** (6), 963–980, Dec. 1983.

[155] C J Weinstein, M L Malpass and M J Fisher. The traffic performance of an integrated circuit and packet switched multiplex structure. *IEEE Trans. Commun.*, **COM-28** (6), 873–878, June 1980.

[156] S B Weinstein. Personalized services on the intelligent wideband network. In *Proc. IEEE Int. Zurich Seminar on Digital Commun.*, pages 13–18, March 1986.

[157] P E White. The broadband ISDN — The next generation telecommunications network. In *Proc. IEEE Int. Conf. Commun. (ICC '86)*, pages 385–390, Toronto, June 1986.

[158] C Wu and T Feng. On a class of multi-stage interconnection networks. *IEEE Trans. Computers*, **C-29** (8), 649–702, Aug. 1980.

[159] L T Wu. Mixing traffic in a buffered banyan network. *Proc. 9th Data Commun. Symp.; ACM SIGCOM Computer Commun. Review*, **15** (4), 134–139, Sept. 1985.

[160] L T Wu and N C Huang. Synchronous wideband network — an interoffice facility hubbing network. In *Proc. IEEE Int. Zurich Seminar on Digital Commun.*, pages 33–39, Mar. 1986.

[161] L T Wu, S H Lee and T T Lee. Dynamic TDM — A packet approach to broadband networking. In *Proc. IEEE Int. Conf. Commun. (ICC '87)*, pages 1585–1592, Seattle, June 1987.

[162] H Yamada et al. High-speed digital switching technology using space-division-switch LSI's. *IEEE J. Select. Areas in Commun.*, **SAC-4** (4), 529–535, July 1986.

[163] Y S Yeh, M G Hluchyj and A S Acampora. The Knockout switch: A simple modular architecture for high-performance packet switching. *IEEE J. Select. Areas Commun.*, **SAC-5** (8), 1274–1283, Oct. 1987.