INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

UMI®
800-521-0600

PHOTONIC BUS AND PHOTONIC MESH NETWORKS - DESIGN TECHNIQUES IN EXTREMELY HIGH SPEED NETWORKS

by

ALLAN M. BIGNELL, B.A.Sc., M.Eng.

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
Doctor of Philosophy
in Electrical Engineering

McMaster University

DOCTOR OF PHILOSOPHY (1997)
(Electrical Engineering)

McMaster University
Hamilton, Ontario

TITLE:	Photonic Bus and Photonic Mesh Networks - Design Techniques in Extremely High Speed Networks
AUTHOR:	Allan M. Bignell, B.A.Sc. (University of Waterloo), M.Eng. (University of Toronto)
SUPERVISOR:	Dr. Terence D. Todd
NUMBER OF PAGES:	i to xiii, 1-1 to 1-30, 2-1 to 2-62, 3-1 to 3-171, 4-1 to 4-5, R-1 to R-11

PHOTONIC BUS AND PHOTONIC MESH NETWORKS

Abstract

This work describes two network designs—Photonic Bus Networks and Photonic Mesh Networks. These two designs were developed in order to provide the basis for exploring the possibilities of novel design techniques in future fibre networks. In both cases, attempts are made to take advantage of emerging technologies in the area of fibre optics. The networks were not created to compete with other network designs, although obvious comparisons with other networks are possible. Rather, these networks were conceived to create a framework within which to study the behaviour of certain phenomenon that may be present in some future fibre networks. Both simulation studies and analytical modelling were used to improve understanding of the behaviour of the network design techniques.

Photonic Bus Networks build upon the emerging IEEE 802.6 network standard by considering a multihop implementation on a physical transmissive optical star topology. Media access techniques are designed to allow for slot reuse at each node, to improve network performance while ensuring fair network access. The combination of the new media access with the broadcast nature of the physical optical network allows for the use of new and novel bandwidth allocation design techniques that would not be possible in IEEE 802.6 networks.

Photonic Mesh Networks can be considered a two-dimensional generalization of Photonic Bus Networks. They are topologically regular two-dimensional grid networks that use a novel routing technique known as deflection routing and allow for easy bandwidth allocation. A framework is created for the consideration of traffic processing at a node in a Photonic Mesh Network. Algorithms are defined which consist of three separate but related strategies - the access strategy, the routing strategy

and the allocation strategy. Simulation studies indicate that the interaction of these separate strategies significantly affect the performance of the network.

Acknowledgements

I wish to extend my sincere appreciation to my supervisor, Dr. Terry Todd, who was a constant source of encouragement, guidance and thoughtful insight.

The opportunity to complete this research was made possible by the support of the Telecommunication Research Institute of Ontario (TRIO) and Westinghouse Canada. During the completion of my studies and research I was seconded from Westinghouse to a TRIO project under the guidance of Dr. Todd. I applaud the foresight of both TRIO and Westinghouse for allowing me to complete this research in a cooperative manner. In addition to financial support, both parties took a genuine interest in my progress. I found it a personally rewarding and broadening experience, and hope that both TRIO and Westinghouse view it as a very positive initiative. I would highly encourage other such initiatives in the future.

I also wish to extend a very special note of appreciation to my children, Eric, Jean, Sara and Marc. I especially want to thank my wife, Francine. During completion of this research I have demanded much in the way of patience and special consideration. Without the support of my family this work would never have been completed.

I dedicate this work to the memory of my father, Lloyd Bignell, who was not alive to see it finished, and to my mother, Christine Bignell, who never doubted me for a minute.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

LIST OF FIGURES

SECTION 1

SECTION 2

LIST OF FIGURES (Cont'd)

LIST OF FIGURES (Cont'd)

Section 3.3 (Cont'd)

LIST OF FIGURES (Cont'd)

Section 3.4

Section 3.5

LIST OF FIGURES (Cont'd)

Section 3.5 (Cont'd)

LIST OF TABLES

1.0 INTRODUCTION

1.1 Background

There is a great deal of speculation regarding the future of networking technologies. At the heart of much of this speculation is fibre optic technology. Currently, many researchers are considering the design of computer communication architectures that will effectively exploit the use of new fibre technology [Okos87]. In such systems, it is expected that there will be a merging of packet and digital switching techniques, since the capabilities of typical node processors will be overwhelmed by the bandwidth available on the lightwave fibres. As a result, the design of future networks will have to draw heavily upon fast-packet switching methods [Turn88], which require that transit switching nodes realize minimal data link control functions and perform tasks such as routing, flow control and buffer management in fast hardware implementations. At the time of this work, there are many researchers exploring various possible implementations [Lee88a, Magl87, Math87, Neum88].

It is also apparent that in the design of these networks, however, there are various (and largely unexplored) bandwidth/nodal-complexity trade-offs. This suggests that since bandwidth may be an abundant and inexpensive commodity in such networks, bandwidth utilization and efficiency may be exchanged for simplicity in the node design of Metropolitan Area Networks (MANs). This trade-off may permit future MANs and extended-LANs to be much more distributed, resulting in a higher

'bandwidth density', since the node design will allow for much less expensive switches than are currently possible.

A primary goal of recent research in this field is to provide access to the enormous bandwidths available on typical fibre channels. It is anticipated that networks based upon the use of Wavelength-Division-Multiplexing (WDM) and coherent laser modulation (i.e. FDM) will eventually permit Gbps operation using multichannel network protocols. An all-optical system may be implemented in this way by employing a reflective star topology [Acam88] interconnected by nodes transmitting on different wavelengths. This type of network also permits considerable design flexibility in that arbitrary 'virtual topologies' can be established. For example, in [Maxe85] this capability is used to structure the topology to minimize the buffering required at the switching nodes themselves.

One proposed network design is the multihop approach. In an optical multihop network, nodes are interconnected using a virtual-topology with attractive connectivity properties [Acam88]. In this type of network, the nodes may perform conventional store-and-forward buffering operations using Asynchronous Transfer Mode (ATM). It is expected that the cost of such an implementation may be quite high for small to moderately-sized interconnection networks.

There are two major initiatives that should be reviewed since they heavily impacted the work in this thesis. One is related to a MAN design known as Distributed Queue Dual Bus (DQDB), and the second is a network known as the

Manhattan Street Network (MSN). This thesis builds on both of these efforts. It is particularly targeted at extended-LAN and Metropolitan Area Network applications. An introduction to DQDB and MSN follows.

Present LANs consist of two basic types - buses and rings. In [Fine84], Fine and Tobagi attempted to classify a myriad of new LAN protocol proposals in a unified manner. They point out that in ring systems there is a requirement for an explicit method of accessing the media, most commonly by use of a circulating token. Such networks have the desirable properties of high channel utilization and bounded packet delay. In broadcast bus systems, random media access techniques may be employed that are not dependent upon an explicit token. These systems, however, tend to waste some bandwidth due to contention problems, and suffer from unbounded packet delay. In particular, the performance of this scheme deteriorates significantly as the parameter a (the end-to-end propagation delay normalized to the message transmission time) increases. This implies that in high speed applications, the access technique is only applicable in a local environment. The authors categorize a group of Demand Assignment Multiple Access (DAMA) schemes where token passing is implicit and that are applicable to bus networks. These access methods often execute on slotted networks, provide bounded delay and, more importantly for this work, offer throughput and delay that is much less sensitive to a. This makes such schemes well suited to networks with high bandwidth, small packet size and long distances.

The protocol designs summarized in [Fine84] laid the groundwork for the definition of networks with the necessary characteristics for extended LAN and MAN

applications. One such network, based on a logical bus topology, became the IEEE standard for Metropolitan Area Networking, and is defined in the IEEE 802.6 standard [IEEE90]. The network is now known as Distributed Queue Dual Bus (DQDB). DQDB utilizes a dual-bus topology with two counter-flowing unidirectional buses. The topology is identical to Fastnet [Limb82], which is classified as an attempt and defer scheme in [Fine84]. Consider Figure 1.1. Stations are connected between two

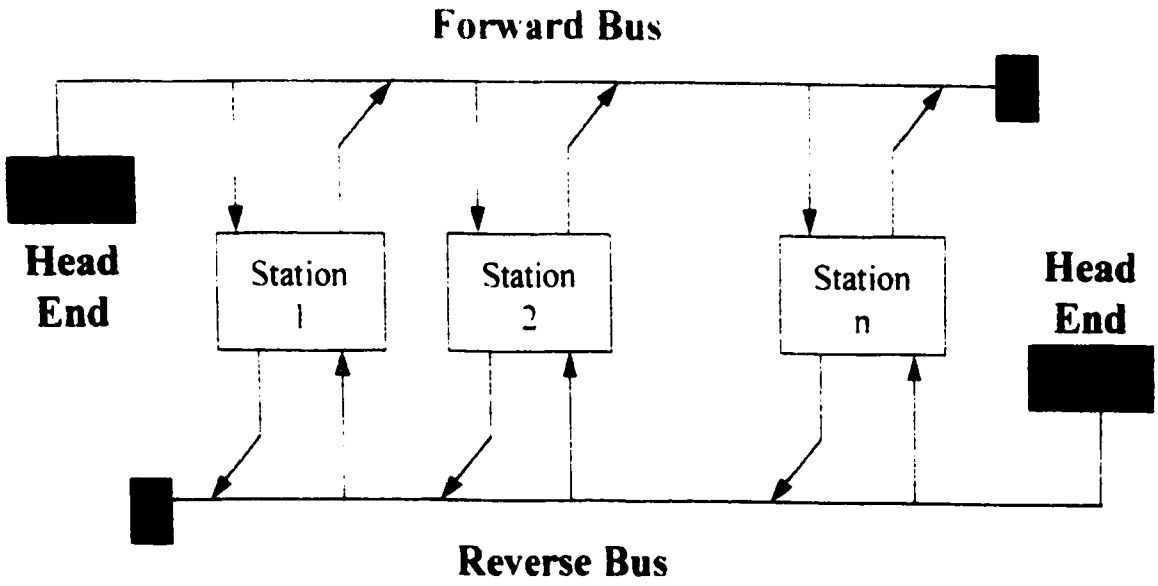


Figure 1.1 Distributed Queue Dual Bus Topology

unidirectional buses as shown. At one end of each bus is a special station called a head-end, which is responsible for dividing the bus into fixed length slots in time. With slotted systems, the bus bandwidth is efficiently utilized, by each station with data to transmit acquiring an empty slot. The difficulty is that the access to the empty slot is determined by the location of the station on the bus. Stations closer to the head-end have the first opportunity to use the empty slot and are therefore favoured. The schemes discussed in [Fine84] typically addressed this problem by creating a

round-robin access scheme, where each station is allowed access to, at most, one slot each round. This introduces a dependence on end-to-end propagation delay in order to determine when new rounds should be initiated. DQDB overcomes this dependence by using the reverse bus to reserve slots on the forward bus.

The basic operation of DQDB can be described by considering access to the forward bus only. Dual operations occur for the reverse bus. This discussion will be limited to the queued-arbitrated access of DQDB for non-isochronous traffic and will consider only a single priority level (for a complete description of the protocol see [IEEE90]). In this case, a DQDB slot contains two signalling bits - a Request bit and a Busy bit. The Busy bit on the forward bus indicates that the slot is busy and cannot be used by a station. The request bit on the reverse bus is used to inform upstream stations that a downstream station requires an empty slot for transmission. The distributed queuing mechanism is established through two counters at each station - a Request counter (denoted RQ_i) and a Countdown counter (denoted CD_i). Station i is said to be in the idle state when it has no data to transmit. While in this state, the station is constantly monitoring both the forward and reverse buses in order to keep track of the state of the global queue. It does so by incrementing RQ_i each time a slot passes on the reverse bus with the Request bit set. This indicates that a downstream station requires an empty slot. The theory is that this requesting station is in front of station i in the global queue. When an empty slot passes on the forward bus, a request is said to be satisfied and RQ_i is decremented. RQ_i thus keeps a record of stations that are ahead of station i in the global queue and should get earlier access to empty slots. When station i has a packet to transmit, it enters the global queue by

setting the Request bit on the reverse bus, copying RQ_i to CD_i and clearing RQ_i. The station is now in the countdown state and increments RQ_i for each request bit set on the reverse bus and decrements CD_i for each idle slot on the forward bus. When $CD_i = 0$, the outstanding requests have been satisfied and station i uses the next available slot.

Note that only one slot per station can be enqueued in the global queue at a time. Outstanding request counters are used to keep track of requests not yet submitted. In the basic implementation, all busy slots propagate the full length of the bus. In the ideal case of no propagation delay, all stations receive information regarding the state of the global queue instantaneously, and the media-access provides first-come-first-served service. The effect of propagation delay, which is inevitable in high-speed networks covering a wide geographical area, complicates the understanding of DQDB considerably. These discussions are deferred until Chapter 2.

A second initiative that serves as important background for this thesis relates to the need for considering network topologies that are more highly interconnected than traditional buses and rings. Such networks are typically mesh-connected, and historically have been used in the wide area networking arena. The technology in such networks is complicated by the requirement to deploy store-and-forward nodes with message routing capabilities, and the need to control the flow of data entering the network, to re-sequence packets at the destination and to recover packets with errors. A significant development in this direction was made by Maxemchuk [Maxe85, Maxe87a], who proposed the Manhattan Street Network or MSN. The concept

suggested was that in a very high-speed backbone network, it is expected that there will be many distributed nodes that are heavily interconnected. By enforcing specific constraints on the topology of the system, the routing, flow control and buffering functions that must be performed at a node can be drastically simplified. An illustration of the MSN topology is shown in Figure 1.2. Note that each node has

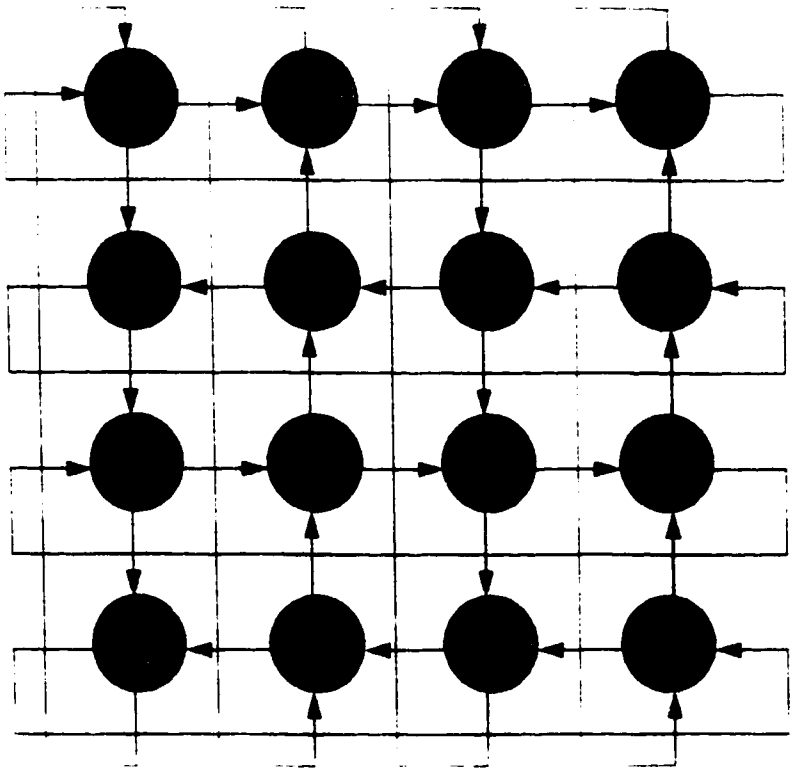


Figure 1.2 Manhattan Street Network Topology

exactly two incoming and outgoing fibre channels. Also, the network itself is defined to be 'topologically homogeneous' due to the presence of the 'wrap-around' links that tie together nodes on the physical boundary of the network. Under the assumption that each link is a unit hop, it is easy to verify that the network may be re-drawn into various 'logical topologies', with any single destination node in any position resulting in the same interconnection pattern. (Networks with this property are said to be

symmetric [Reed87].) In the MSN, a bandwidth/nodal-complexity trade-off is made by 'deflecting' packets onto alternate routes when contention for a particular channel exists at a node. This is a reasonable approach, since the topologies considered are expected to be heavily interconnected, and many different paths will exist between each source/destination node pair. Note that in conventional packet switching (and in current fast-packet designs [Turn88]), a packet would typically be buffered (thus consuming further node resources) until the desired output channel is available. The absence of node buffering is achieved in the MSN by insisting that the number of input and output channels is equal to two for each switch, hence the number of transit packets at a node can never exceed the number of available outgoing links. The use of 'deflection-routing' in this manner immediately relieves the nodes themselves from the task of transit packet buffer management, although it should be noted that the trade-off involves a sacrifice in total system capacity performance. In this case, the drop in usable capacity is a decreasing function of the network size [Maxe89]. It should also be noted that the use of memoryless deflection-routing in this manner is appropriate for data traffic where end-to-end packet sequencing is available.

Another remarkable property of MSNs is that they appear to implicitly provide input traffic flow control. This occurs because network entry traffic at a node is only allowed access to a channel if free channel slots are available. Published results [Maxe89, Gree86] suggest that this effect tends to automatically throttle the input traffic in areas of heavy congestion. As a result, it appears that there is very little decrease in throughput versus applied load as the system approaches and exceeds

capacity, even for networks with thousands of nodes [Maxe89]. Hence, very stable operation is expected in this type of architecture.

Extensions to the original MSN concept were made by Borgonovo and Cadorin [Borg87], who investigated routing methods in a bi-directional version of the MSN referred to as HR4net. In addition to the shortest path routing techniques originally proposed in [Maxe87a], a row/column scheme was also suggested.

1.2 Motivation

The work in this thesis is targeted at extended-LAN and MAN environments. It is clear that there will be an abundance of requirements for such applications in the future. At the time of this work, there are a variety of techniques that mark the first steps towards satisfying the requirements (FDDI, SMDS, T3). In this work, we will consider ways that fibre optic technology may change the manner in which such networking is done. We will limit our considerations to systems that are slotted in time, since it is anticipated that Asynchronous Transfer Mode (ATM) networks will dominate Wide Area Networking (WAN) in the future. ATM networks will be based on fixed length packets of data. In addition, we will limit our considerations to non-isochronous traffic, which is anticipated in many LAN and MAN applications.

There are two major parts to this work, with one chapter devoted to each. In both cases, attempts are made to study certain impacts of emerging fibre optic technology on networking by way of a specific and original network example. The first body of work is related to ways in which novel system design techniques might be used to improve the performance of emerging MANs based on the IEEE 802.6 networking standard. These networks are called Photonic Bus Networks (PBNets). At the heart of these techniques is the use of a physical optical transmissive star, and the design of a media access technique that allows the order of stations to change on a 'virtual' dual bus topology network. These networking concepts intentionally build upon the existing IEEE 802.6 standard. One particular technique, known as Receiver Allocation, allows the 'virtual' dual bus topology to be modified by selectively adding receivers to certain nodes. This marks the first step away from a bus topology

towards a mesh topology. In this case, the topology is a lightly interconnected and irregular mesh, which utilizes very similar media access to the original dual bus topology. Although the Receiver Allocation networks are considered as PBNets, they actually point the way to consideration of mesh network designs and lead into the second major thrust of this thesis, Photonic Mesh Networks (PMNets).

PMNets are highly interconnected networks, and assume the use of point-to-point fibre optic connections. The higher interconnection promises better performance. In addition to higher interconnection, the proposed approach utilizes the fibre network to create a logically regular topology that allows the use of a novel routing technique known as deflection routing. Little analytical work is available to model deflection routing networks. In this work, we propose a network that builds upon the Manhattan Street Network (MSN) [Maxe85]. This was done for several reasons. At the time of this research, the MSN was, and still is, the topic of much research interest. Our proposal is a slight modification of the MSN that is used to develop a framework within which to consider regular mesh deflection routing networks. Modifications of routing rules for our network can result in the MSN as one special case. Deflection routing networks are still poorly understood, and consequently we felt that a generalization of the MSN would be more appropriate for studying the behaviour of such networks. In particular, accurate analytical modelling is more difficult to achieve, but the development of such models give insights into network behaviour that would otherwise not be noticed. In addition, consideration of the more general proposed network leads to the development of three major strategies to be considered when designing regular grid topology deflection routing networks.

1.3 Contributions of the Thesis

The contributions of this thesis are in the areas of protocol design and performance modelling of Photonic Bus Networks (PBNets) and Photonic Mesh Networks (PMNets). Both networks use novel design techniques based upon the capabilities of emerging fibre technology.

In Chapter 2, a physical star topology, based upon a transmissive optical star, is used to create a virtual dual bus topology network similar to the IEEE 802.6 dual bus. Media access techniques are designed to allow for slot reuse at each station, and complete analytical delay models are developed. All analytical models are compared with exact simulations. The media access protocols are designed to provide fair performance under overload conditions, with as little impact as possible on performance under normal operating conditions. These protocols make way for the use of system design techniques, which allow the system bandwidth to be allocated in accordance with the traffic flows on the bus. The two main techniques suggested (others are possible) are Topological Design and Receiver Allocation. The analytical models developed are appropriate for both of these design techniques.

Chapter 3 explores a two-dimensional version of PBNet that utilizes deflection routing techniques on a regular grid network. The network is called PMNet. A general framework within which to consider the design of deflection routing networks is developed. Three strategies to the design are identified - Access, Allocation and Routing strategies. Simulation work is used to give insight into the various trade-offs that exist between these strategies, and how they affect performance. Many algorithms

are developed in order to explore these trade-offs. Exact simulations for the algorithms are used. It is necessary to consider the interplay of all three strategies when designing such networks. Analytical models are developed to test the applicability of certain independence assumptions in simplifying the analysis of these networks. All analytical models are verified using exact simulations and input queuing dynamics are considered in the models.

1.4 An Addendum

The work associated with this thesis has been completed on a part-time basis. This approach to study has its own unique rewards and challenges. One challenge is the time difference between the writing and the publishing of this thesis. In order that this thesis can be considered in the light of current literature at its time of publication, I have included this brief addendum. The interested reader should keep this additional background information in mind when considering this research.

During the 1980's, fibre optics became the dominant transmission medium. It offered very high speed point-to-point connection between electronic processing intensive network nodes. However, recent developments in optical devices are extending the role for optical networks beyond the simple point-to-point transportation of bits [Midw93]. Technology development, along with explosive growth in applications demanding higher per-user bandwidth, are forcing the consideration of new, novel network designs that capitalize on the advantages of optical networks. This has led to increased research interest in deployable optical networks, their architectures, protocols and design.

In order to appreciate the growth in traffic demand, consider the explosive nature of the Internet. The per-user bandwidth demand for World Wide Web (WWW) mode of PC usage has accelerated to a factor of eight fold per year. ***Point-and-click*** access to objects, independent of location, has grown dramatically. This eight-fold factor in faster growth is greater than any other known measure of user demand or technology performance [Gree96]. The trend is expected to continue with the

acceptance of techniques such as JAVA, where, increasingly, users will download applications, as well as objects. The growth of such non-isochronous traffic will play an increasingly important role in the design of future networks, due largely to its bursty nature. It has already played a huge role in the development of ATM networking technologies.

Increasingly, there is a recognition that traditional network design techniques involving fibre transmission and complex electronic processing at network nodes will not be adequate in the future. In particular, there is a general belief that electronic processing will become a bottle-neck [Midw93a]. As a result, some form of optical network is considered to be the likely generation of networks to follow ATM [Gree96]. Networks based on copper, coax or radio offer only temporary relief.

The advantages of optical networks include:

(a) high bandwidth per user,

(b) protocol transparency,

(c) high path reliability,

(d) simplified Operations and Management.

These networks do have many challenges, however. A review of the technology is in order.

1.4.1 Optical Technology

The field of optical devices is evolving quickly. For good summaries, see [Gree96, Midw93, Midw93a]. A few points are included below.

Optical carrier frequencies are at least 1000 times that of electronics. In addition, electronics becomes very complicated at high frequencies (GHz range) due to, among other things, parasitic capacitance, impedance mismatches, transmission loss and electromagnetic coupling. The problem is normally not the number of electronic gates, but how to interconnect them. This implies that the switching requirements for future networks will likely exceed the ability of electronics to process the data stream. The result is a trend which will see electronic processing move to the periphery of the network and optical technology move to the core of the network. Since optical devices are presently limited in their processing abilities, complexity must therefore be moved out of the network core. This is acceptable because electronic processing deals well with complexity and it will reside at the periphery of the network.

Some recent optical device developments are worth noting. Erbium Doped Fibre Amplifiers (EDFA) now allow long, high speed, repeaterless fibre cables. Planar integrated optics have developed simple, electronically controlled, switching devices based on Lithium Niobate technology. Simple crosspoint switches based on this technology are capable of carrying more than 10 Gbps of data with a reset time of 1 ns. Wavelength Division Multiplexing (WDM) technology increases the bandwidth previously available on a fibre by 10 to 100 times. This allows a massive expansion

of the present fibre system at low marginal cost. Acousto Optic Tunable Filter (ACOT) wavelength selective switches have been developed.

While there are still many questions about how all of this technology will impact networks, there are presently at least two areas where the abilities of optical technology are known to impose major limitations. In traditional networks, complicated electronic processing in network nodes is common and will continue for some time [Schu97, Chou96]. Optical processing, however, is presently limited to simple or very special functions. In addition, buffer storage, so common in present networks, is provided in the optical domain by Optical Delay Lines (ODL). These devices are expensive and are limited to short buffer sizes. Capitalizing on the many advantages of optical networks, while overcoming the above limitations, is the subject of many research efforts, including this thesis.

1.4.2 Relevant Background Research

There is a deep body of research going on in this area. In [Midw93], Torng and Daddis suggest a hierarchical classification scheme for switching architectures based on the following five requirements of the design:

(a) Links - either shared or dedicated.

(b) Transport - either assigned (scheduled) or statistical (dynamic).

(c) Routing - centralized or distributed.

(d) Link Contention - buffered or unbuffered.

(e) Switching Technique - space, time, wavelength or address filtration.

Many of the proposed switching techniques, including optical and hybrid optical-electronic techniques, are classified in this reference.

The class of network we will consider will be simple enough to allow for implementation as an optical network. All networks are considered to be WDM-based in order to take advantage of the huge optical bandwidth. In almost all cases, network node complexity is reduced by the imposition of a logical network topology and low node processing, normally requiring minimal buffering. These networks are often divided into two groups: single hop and multihop networks. Single hop networks involve establishing a path across the optical network, called a lightpath. Data sent on that lightpath always follows the same path through the network and is said to arrive at the destination in a single hop. Given that the number of wavelengths available in WDM networks is limited to less than 100 (and only forecasted to be as high as 1000), it is generally accepted that bandwidth reuse will be a permanent design

requirement in optical networks [Gree96]. Many research efforts have been undertaken to overcome this requirement in single hop networks.

In [Jano96], Janoska and Todd propose a spatial wavelength reuse technique to overcome available channel shortages. In [Hall96, Jue96], the single hop Rainbow network based on a broadcast star is described. [Bane96] describes a wavelength routing single hop network specifically dealing with the problems of routing and wavelength assignment. [Muir96] suggests several protocols to create a Distributed Queue for stations to access wavelengths in a WDM-based broadcast star. [Yate96] considers the use of optical wavelength translators to improve the blocking probability in wavelength-routed networks (networks which allow any user to be connected to any other user in the optical domain). [Rous95] attempts to reduce delays in single hop WDM networks by using Time Division Multiplexed (TDM) schedules. [Mokh95] tries to overcome optical transmitter tuning delays by introducing more channel sharing through techniques such as subcarrier frequencies or Code Division Multiple Access (CDMA). It should be noted that this complicates the design. [Bore95] overcomes the channel shortage on a broadcast star network by using tunable transmitters coupled with scheduled TDM channels for handling packet data. Clearly, there is much effort being expended in order to overcome the shortage of optical channels available. Each of these techniques is either a direct or indirect form of sharing (reusing) the available channels. Multihop networks inherently provide a higher degree of bandwidth reuse.

In a multihop network, a packet travels through intermediate nodes to get to its destination. At each of these nodes, the packet may be detected, stored, processed and retransmitted. This approach to optical networking is attracting attention since it directly addresses the problem of limited WDM channels, and because optical devices allowing simple node processing are now available or envisaged. The major research areas involve the development of protocols and flow control schemes which can take advantage of the fibre bandwidth [Minw93a]. Techniques usually involve using the abundant bandwidth to create logical topologies that simplify node processing. Major problems to overcome include congestion control and overflowing buffers in switches. Such problems are traditionally dealt with using complex electronic processing to impose flow control and retransmission techniques. At high speed, these can be challenging problems in the electronic domain [Chou96]. They would be impossible in the optical domain and consequently weigh heavily in the design considerations of multihop networks.

With these considerations, multihop optical networks are evolving away from the broadcast star topologies common in single hop networks, towards busses and more general structures based on wavelength routing and reuse. Much research is still needed [Midw93a]. Both circuit and packet switching techniques have been proposed for optical networks. Recently, a trend has developed which places greater emphasis on packet switching to provide added flexibility for advanced applications, and efficient allocation of bandwidth to multiple users with different transmission needs, especially at high speeds. Packet switching gains further support through the worldwide emergence of the ATM standard. For this reason, a major consortium of

academic and industrial enterprises has formed to investigate networking issues in optical contention resolution, the construction of experimental contention resolution optical devices (likely using switched optical delay lines) and the building of a prototype packet-switched optical network [Chla96].

Research in multihop networks is very active and many of the themes are similar to single hop networks. [Trid97] proposes sharing WDM channels on a multihop, broadcast star using a TDM technique. Others have proposed random access techniques such as ALOHA. In [Midw93a], Prucnal points out, as others have, that future sources will be very bursty and require multiple concurrent sessions. In such situations, packet switching is considered more appropriate. He points out that the key functions of a photonic fast-packet switch are to:

(a) route packets in a time interval less than the packet length (note that at 5 Gbps, an ATM cell is 85 ns long),

(b) perform conflict resolution in an interval less than the optical packet length,

(c) synchronize incoming packets.

[Cruz96] describes an ultra high speed packet switch using cascaded optical delay lines and a simple distributed electronic control algorithm to configure the photonic switches. In many proposals like this, packet header data is encoded at lower rates than the data of the packet. This allows headers to be processed electronically to generate control signals for photonic crossbar switches. [Seo96] studied the use of a shuffle network topology for an ultrafast multihop lightwave network. [Sabe96] studies the limits of using optical digital cross-connects to provide an optical transport network using WDM. [Qiao96] researches the trade-offs between space and time

switching in photonic networks. The trade-offs are concerned with switch complexity and cost. [Ines95] introduces GEMNET as a multihop, packet-switched, WDM-based, shuffle-exchange network implemented on a passive star. [Rama95] describes a WDM-based network of general topology that routes packets through optical switches that provide wavelength reuse using a routing and wavelength assignment algorithm [Lee95] investigates a WDM packet switch that uses wavelength conversion (shifting optical packets to a different wavelength) in order to avoid dropping packets due to wavelength contention. [Haas93] describes an optical switch targeted at WDM applications, called the Staggering switch, that distributes optical packets to the switch such that no output collisions (and thus contention) occur. There is some probability of packet loss. It is clear that this is a very broad and active area of research.

1.4.3 Related Research

We now consider research which is directly related to this thesis. This thesis deals with multihop optical networks known as Photonic Bus Networks (PBNets) and Photonic Mesh Networks (PMNets). PBNets are based on the IEEE 802.6 DQDB standard. We study the idea of bandwidth reuse (destination release) to allow development of techniques such as receiver allocation and topological design to improve network performance on a WDM broadcast star network. The work involves the design of a fair reuse protocol in the presence of the DQDB BandWidth Balancing (BWB) mechanism. PMNets build upon many of the techniques introduced by Maxemchuk in the Manhatten Street Network (MSN). Such networks are applicable for optical implementation because they simplify the node design. In particular, the use of deflection routing is considered important for optical implementations. The work provides a better understanding of this routing technique and the performance of deflection routing networks. We will now quickly review the related research.

As stated, it is accepted that bandwidth reuse will be necessary in foreseeable networks. This will involve spatial reuse of bandwidth and consequently the interest in multihop networks. Application of spatial reuse to simple linear networks (rings and busses) is quite common in the literature. In these networks, a very simple topology is adopted to simplify the technique used to access the network. When compared with higher density interconnection patterns, these linear networks trade throughput and reliability for simplicity. Such techniques are usually applied to LAN or MAN applications. A great deal of research effort is focused on methods of accessing these networks in a manner which is fair to all nodes. The introduction of

spatial reuse complicates the problem. In PBNets, we will study the use of spatial reuse in DQDB networks, recently standardized by the IEEE. The subject of reuse, combined with fair access, is seen to still be of extreme interest in the literature. A brief summary is in order.

In [Cido97], spatial reuse is proposed for ring networks. It is pointed out that, while reuse increases throughput, it often introduces problems of fairness where heavily loaded nodes prevent other nodes from accessing the network. This ***starvation*** problem is addressed through an allocation of transmission quotas. [Rubi96] imposes a special local regulation protocol to overcome the fairness problem introduced by applying destination release techniques. In [Kaba96], a Fair Distributed Queue (FDQ) protocol is proposed. The network uses the same node hardware as DQDB but is based on a folded bus topology. Unlike DQDB, FDQ allocates equal bandwidth under heavy load to all active users, without wasting bandwidth (through the BWB mechanism). However, since destination release is not allowed in FDQ, its throughput is limited to that of a single bus. Destination release is still an unsolved problem in FDQ.

Dual bus networks are particularly prominent in the literature due to the standardization of DQDB. Each time reuse is proposed, it is necessary to consider the fairness problem. [Shar95] suggests a protocol that, under most identified circumstances, can be proven to be fair. It is required to modify the Access Control Field of the DQDB slot to implement the technique. [Brew95] gives a good overview of recent literature before proposing a protocol that modifies the counters in the node

in order to improve fairness in DQDB networks employing erasure nodes. [Huan95] proposes the addition of tunable transmitters and receivers to dual bus networks as a way to improve throughput without confronting the reuse problem. Other references to the problem abound. For examples, see [Huan95a, Nara95, Shar94, Tant94, Pach93, Borg93, Karv93, Chen93].

A natural extension of the thinking around linear networks is the consideration of mesh networks. Mesh networks support higher throughput, without increasing the transmission rate, by using a smaller fraction of the network capacity (a form of reuse) to communicate between nodes. They increase network reliability and offer multiple paths between source and destination. If designed properly, they offer flexible bandwidth allocation capabilities for network evolution. All of this is achieved at the expense of more complex access and routing strategies than found in linear networks [Maxe93]. Once again, due to the restrictions in the number of possible WDM channels, multihop networks are necessary. Due to limitations in optical technology, nodes can only process a limited amount of information, typically determined by the number of ports at the switch. In addition, the use of a regular topology is known to simplify the complexity of the routing task when compared with arbitrary topologies. These constraints imply the use of a simple logical topology and impose a restriction on the maximum degree of any logical topology [Rama95a].

In the literature, there are many references to techniques used to simplify the design of nodes in an optical network. We will be concerned with the use of simple, regular, logical topologies in order to simplify distributed routing rules. We will also

include the use of deflection routing techniques to remove the node complexity associated with transit buffer management. Deflection routing resolves link contention, normally requiring buffers, by actually transmitting packets on a non-preferred link. In addition to simplicity, deflection routing has the advantage of diffusing packets away from congested links and automatically flow controlling input packets. Deflection routing has the drawback, however, of being a new networking technique that is not well understood. This thesis will contribute significantly to the understanding of deflection routing. Research in related areas continues to emerge. A summary is given below.

In [Liew97], a Shuffle Exchange network with deflection is considered. Deflection is used to resolve link contention but it is found that the performance of the network suffers considerably if contention is resolved randomly. This would be consistent with our findings in PMNets. In [Kova96], a proposal for a semi-random topology is submitted. Random mesh topologies imply large routing tables (which require periodic updating) and complicated shortest path routing algorithms. Regular topologies have more difficulty adapting to slow changing traffic patterns and cannot be optimized to specific traffic requirements. A semi-random approach is adopted in the reference. The PMNet was studied in this thesis since it allowed for the flexible allocation of bandwidth, a requirement to support semi-random topologies, and improving the adaptability to changing traffic patterns. In fact, the receiver allocation technique applied to PBNets forms a semi-random topology and leads to the consideration of PMNets. [Chla96a] describes an optical switch design that utilizes Optical Delay Lines and deflection routing techniques. In [Dobo96], an asynchronous

deflection routing network is proposed for application to Manhatten Street Networks (MSN). Routing and conflict resolution techniques are introduced to try and improve performance. The asynchronous implementation requires more buffering (which is difficult and expensive in the optical domain) and provides less ability to tailor the routing and conflict resolution techniques. We found this last point to be an important consideration in PMNets. An important class of future traffic will be multicasting. [Liew96] describes a technique for multicasting in a MSN. [Modi96] considers the same subject. In optical WDM multihop networks, it is generally accepted that the networks employing a node degree of three or four are preferred [Park95]. In [Park95], a hierarchical MSN is developed that uses WDM and deflection routing. [Lee95a] proposes a binary addressing scheme for MSN to allow for the future addition of nodes to the network. Optical delay line filters are expensive to incorporate into switch designs, therefore, many designs rely on deflection routing techniques [Bann95]. This places a greater emphasis on the need to understand the nature of deflection routing networks. [Bono94] describes some of the design implications of deflection routing switches for optical networks, with particular emphasis on optical memory schemes and their control algorithms. Simplicity is emphasized. [Borg94] discusses the design of an optical switch for a Bidirectional MSN (BMSN), hitting on the same theme of simplicity as does [Chla93].

In summary, the two primary sections of these thesis are associated with PBNets and PMNets. The research interest is in considering design techniques that will allow for flexible long-term allocation of bandwidth in future optical networks. The proposed techniques address the trade-off between the performance of the

networks and the need for node simplicity to overcome limitations in optical implementations.

2.0 PHOTONIC BUS NETWORKS (PBNETS)

2.1 Introduction and Architectural Overview

There are currently many efforts directed towards the application of coherent laser modulation in certain networks [Okos87]. Although commercial products are not yet available, this technology has the potential for tapping the enormous bandwidth on typical optical fibres. When this is achieved, hardware costs are expected to be a function of the *total* number of network transmitter/receiver pairs, independent of the number of channels used. For example, in an implementation with a total of n transmitters and receivers, a single channel may be used (to which all transmitter/receivers are tuned) or alternately, n different channels may be used (with each transmitter/receiver tuned to a distinct channel) with essentially the same cost regardless of n. The basic construction of PBNets, and their bandwidth allocation, exploits this property.

The design philosophy behind PBNets is that small to moderately-sized photonic LAN and MAN backbone networks can be realized very economically using an interconnection structure adapted from conventional unidirectional bus networks [IEEE90]. When the appropriate media access techniques are used in such systems, a simple node design results that has a very small and economical incremental channel cost. As a result, bandwidth allocation and network evolution in PBNets is expected to be accomplished very easily. Such properties will be very important in future optically-based LAN and MAN networks.

In this chapter, system design techniques and media access protocols are considered that support the PBNet approach to the design of small backbone networks. A PBNet is simply a multihop implementation of conventional LAN and MAN networks, based upon an optical transmissive star. Although it is clear that a variety of 'virtual topologies' can be implemented using the transmissive star physical topology, the intent of this work is to investigate the design of networks that are applicable to an existing and standardized bus topology. The basic building blocks for PBNet thus consist of linear topology multihop networks, interconnected in a dual unidirectional bus virtual-topology using point-to-point optical channels. Since a multihop approach is taken, the protocols considered achieve slot reuse at each node in the network. Slot reuse simply implies that once a slot of information has reached its destination, it is removed from the bus and the bandwidth is made available to other downstream stations. A number of new media-access protocols are considered that are extensions of the IEEE 802.6 DQDB protocol [IEEE90]. The intent is to design a protocol that ensures the fair allocation of bandwidth in a general slot reuse network, without unduly reducing the performance of the network. The limitations of the present algorithms in this regard are considered.

The optical implementation utilized in PBNet will result in considerable flexibility in PBNet topological design and bandwidth deployment. This permits inherent network optimization and performance evolution that is difficult in similar networks based on physical point-to-point fibre links. Techniques such as Topological Design and Receiver Allocation can easily be used to evolve and tailor the performance of the system to accommodate changing user traffic patterns. Formal

algorithms for this purpose, including comparisons with other protocols, are included in [Todd91, Todd91a]. An overview of the concepts is included in the next section, in order to put the media-access protocol design into context.

The physical topology of a photonic bus network is based upon passive transmissive optical star technology. In Figure 2.1, the physical structure of an N-node system is depicted. In a transmissive star, each station is attached with an inbound and outbound fibre as shown in the figure. The transmissive star creates a broadcast system such that any inbound transmission is split and sent to all stations on the outbound fibres.

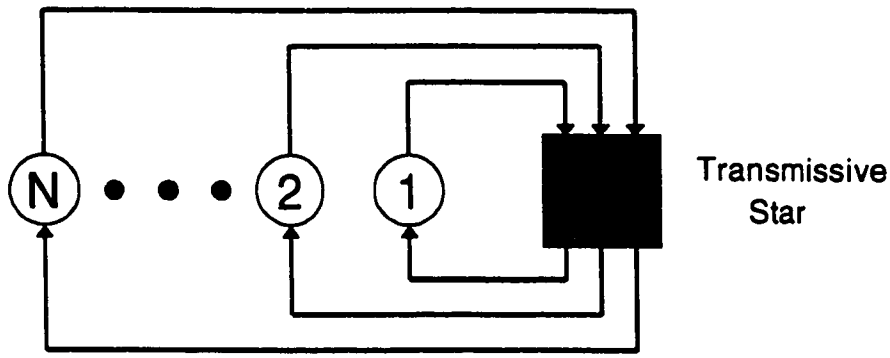


Figure 2.1 PBNet Physical Topology

A basic PBNet building block is shown in Figure 2.2. Here, N nodes have been illustrated with a dual unidirectional bus interconnection topology. It should be noted that the illustrated network is a virtual-topology implemented on the transmissive star system shown in Figure 2.1. Each arrow in the figure corresponds to a single channel allocated on the physical star using WDM, subcarrier multiplexing,

FDM or a combination of multichannel optical technologies. Also note that link (i, j) consists of a channel directed from node i to node j. Empty slots are generated at each end of the network and are propagated in a store-and-forward fashion in the downstream direction on both forward and reverse buses.

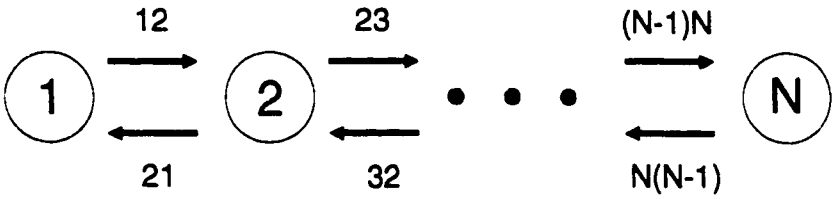


Figure 2.2 Base PBNet Virtual Topology

The system in Figure 2.2 is given in a minimal bandwidth configuration; that is, for an N-node PBNet, the channel assignments shown are the minimum that are required. A PBNet of this kind, without additional allocated bandwidth, is referred to as a *Base PBNet*. Clearly, a Base PBNet requires 2(N-1) distinct optical channels. Since a Base PBNet is implemented as a multihop network, full slot reuse is available at each node. A media access protocol is used to ensure fair and orderly access to the network. This topic will be discussed in detail in the next several sections. However, it should be noted that due to the small per-channel buffering requirement, the node implementation and incremental channel costs of the network are expected to be very low.

2.2 An Overview of PBNet Design Concepts

Performance evolution of a Base PBNet is possible in a number of ways. The concept of flexibility in system design to improve performance through PBNets forms part of this work. The specific design algorithms developed to achieve this improvement are included in [Todd91, Todd91a]. The basic concepts and original motivation will be introduced here. The two design techniques considered are Topological Design and Receiver Allocation.

Due to the optical implementation, topological optimization is very simple. By virtue of the broadcast realization, the station ordering in an N-node Base PBNet may be chosen to optimize the network performance. This is accomplished simply by determining the specific optical channels that the individual stations use for transmission and reception. Clearly, Figure 2.2 corresponds to only one of the N!/2 unique designs that are possible for an N-node network. Even for small networks, this number may be exceedingly large (e.g. for a 20-node PBNet, the number of possibilities is of the order 10^{18}). The topological design problem consists of determining which of these designs is most appropriate, given the network traffic flow matrix. It can be shown that the problem of finding the design that minimizes the maximum over all network link utilizations, is an NP-complete problem. For this reason, several heuristic topological design algorithms based upon the system traffic flow matrix are considered. It has been demonstrated that greatly improved delay-throughput performance is possible using these design techniques [Todd91a].

In an N-node network, the NxN traffic flow matrix is defined to be

$$[F_{ij}] = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1N} \\ f_{21} & f_{22} & \cdots & \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ f_{N1} & f_{N2} & \cdots & f_{NN} \end{bmatrix} \qquad (2.2.1)$$

Note that the nodes are numbered from 1 to N, and each f_{ij} corresponds to the total average flow rate in cells per slot transferred between nodes i and j. The objective of the algorithms is to create an ordering of the nodes so that stations with larger mutual flows are located closer in the virtual topology. Since re-use protocols are being used, this will generally result in a reduction of the average bandwidth required to satisfy arriving transmission requests. Several possible design algorithms are summarized below:

(a) **Fast-Add Algorithm**

This algorithm quickly builds the new network by repeatedly choosing the remaining node with the highest offered traffic to the last node added. This node is then added to the right of the previous node.

1. Find nodes i and j such that $f_{ij} + f_{ji} > f_{mn} + f_{nm}$ for all n,m. Form a 2-node PBNet i,j.
2. Set $f_{ij} = 0$ and $f_{ji} = 0$. Set k = j.
3. Find maximum of $f_{kn} + f_{nk}$ for all n.
4. Add node n to the right of the network.

5. Set $f_{km} = 0$, $f_{nk} = 0$.

6. Set k = n. Repeat 3-6 until design is complete.

(b) **Min-Max Algorithm**

The motivation behind this algorithm is that when nodes are added to the designed network, an attempt is made to locally minimize the maximum link utilization on the new PBNet. Un-added nodes are selected on the basis of maximum traffic between them and a 'super-node', which is formed by the collection of all nodes currently belonging to the designed network. The new node is added to the side of the network, which minimizes the maximum link utilization.

1. Find nodes i and j such that $f_{ij} + f_{ji} > f_{mn} + f_{nm}$ for all n,m. Form a 2-node design PBNet i,j.
2. Form a new traffic matrix $[f_{ij}]$ where the entire design network is treated as a single node 0 with respect to the remaining (unadded) nodes.
3. Find a node i such that $f_{0i} + f_{i0} > f_{0n} + f_{n0}$ for all n.
4. Try node i on the right of the design network. Find $U_R = \max U_k$ where U_k is the utilization of link k in the design network.
5. Try node i on the left of the design network. Find $U_L = \max U_k$ where U_k is the utilization of link k in the design network.
6. If $U_R > U_L$ then add node i on the left side of the design network.
7. Repeat 2 to 6 until all nodes have been added to the design network.

(c) **Min-Bus Algorithm**

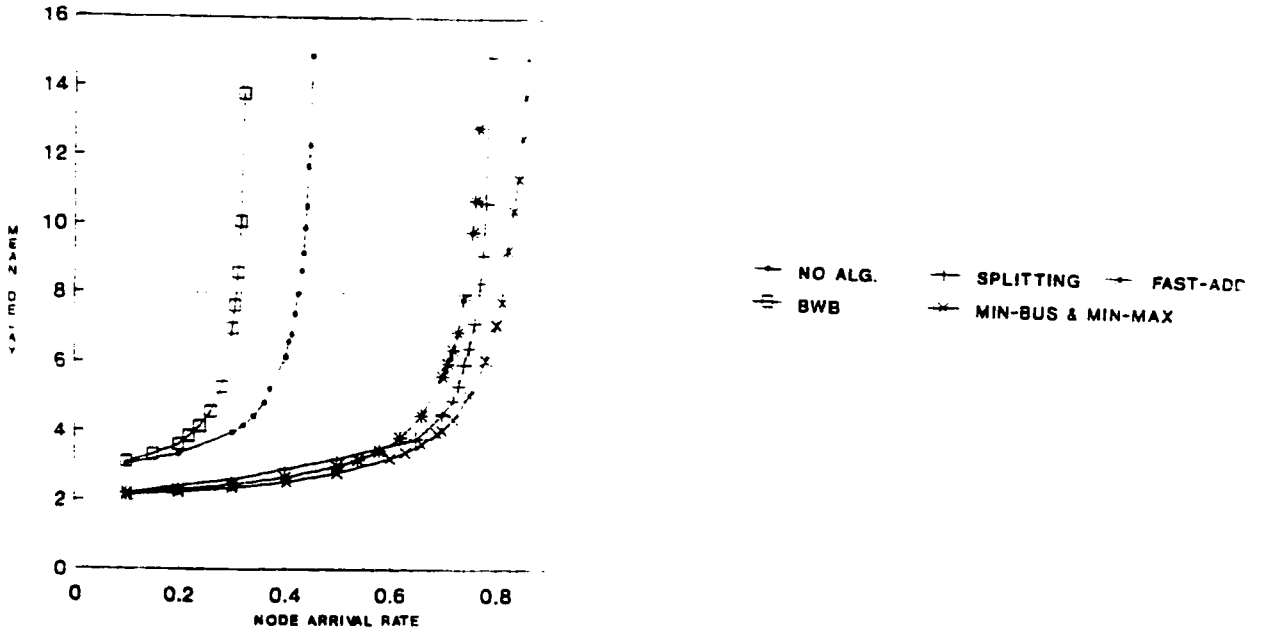
This algorithm is identical to (b), except that U_R and U_L are defined to be the total of all link utilizations when the node is tried on the right and left.

(d) **Splitting Algorithm**

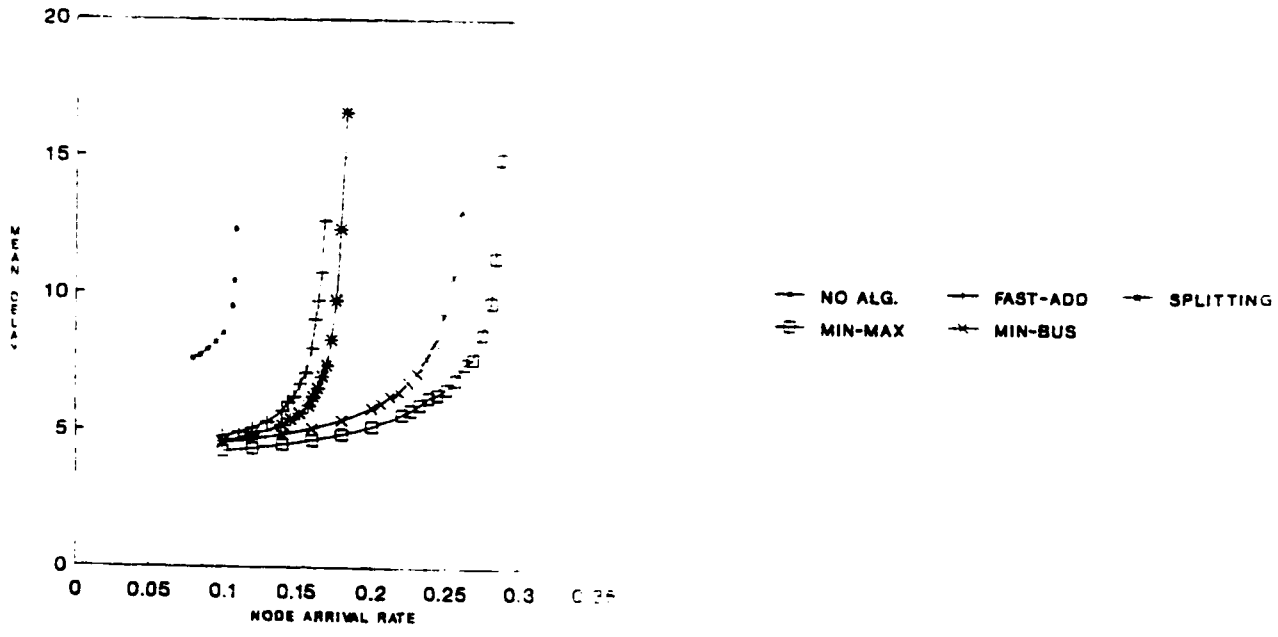
This algorithm is a variation of Algorithm (b) in that the decision on where the new node is placed is made on the basis of total flow across the boundary between the divided network.

1. Find nodes i and j such that $f_{ij} + f_{ji} > f_{mn} + f_{nm}$ for all n,m. Form a 2-node PBNet i,j.
2. Form a new traffic matrix $[f_{ij}]$ where the entire design network is treated as a single node 0 with respect to the remaining (unadded) nodes.
3. Find a node i such that $f_{0i} + f_{i0} > f_{0n} + f_{n0}$ for all n.
4. Divide the design network into two equal halves, right and left. Find S_L and S_R, the total flow between node i and the left and rights halves respectively. If $S_L > S_R$, place node i on the left, else place it on the right of the design network.
5. Repeat 2 to 4 until all nodes have been added to the new network.

The above design algorithms were used to realize PBNets of various sizes and traffic flow matrices. Figure 2.3 gives examples of the improvement in performance that is possible using topological design techniques for the traffic flow probability matrices of Figure 2.4.



(a)



(b)

Figure 2.3 Topological Design Algorithm Comparisons

0.000	0.002	0.051	0.840	0.107
0.090	0.000	0.001	0.209	0.700
0.920	0.023	0.000	0.001	0.056
0.030	0.850	0.091	0.000	0.029
0.097	0.200	0.700	0.003	0.000

(a)

0.000	0.001	0.020	0.002	0.040	0.010	0.005	0.014	0.030	0.600	0.001	0.263	0.004	0.008	0.002
0.010	0.000	0.002	0.010	0.200	0.003	0.050	0.500	0.002	0.060	0.003	0.030	0.093	0.007	0.030
0.005	0.030	0.000	0.030	0.004	0.040	0.005	0.020	0.700	0.130	0.004	0.003	0.020	0.007	0.002
0.216	0.005	0.007	0.000	0.006	0.001	0.600	0.040	0.004	0.050	0.001	0.040	0.007	0.010	0.013
0.060	0.600	0.003	0.060	0.000	0.006	0.005	0.060	0.120	0.050	0.020	0.006	0.002	0.004	0.004
0.600	0.004	0.006	0.080	0.060	0.000	0.050	0.006	0.006	0.008	0.003	0.100	0.005	0.070	0.002
0.100	0.040	0.400	0.008	0.060	0.002	0.000	0.004	0.050	0.231	0.002	0.090	0.003	0.006	0.004
0.600	0.007	0.197	0.030	0.003	0.010	0.050	0.000	0.003	0.020	0.004	0.070	0.003	0.002	0.001
0.030	0.030	0.260	0.010	0.600	0.005	0.001	0.005	0.000	0.030	0.003	0.006	0.007	0.010	0.003
0.300	0.006	0.040	0.500	0.004	0.050	0.003	0.050	0.027	0.000	0.002	0.001	0.006	0.010	0.001
0.003	0.001	0.002	0.005	0.727	0.006	0.003	0.002	0.006	0.220	0.000	0.006	0.006	0.005	0.008
0.003	0.140	0.060	0.600	0.003	0.006	0.100	0.005	0.007	0.007	0.020	0.000	0.008	0.001	0.040
0.010	0.004	0.001	0.409	0.020	0.006	0.392	0.100	0.005	0.008	0.003	0.002	0.000	0.010	0.030
0.005	0.008	0.006	0.610	0.100	0.200	0.005	0.006	0.007	0.030	0.002	0.010	0.001	0.000	0.010
0.004	0.001	0.310	0.626	0.010	0.020	0.002	0.005	0.007	0.001	0.001	0.004	0.003	0.006	0.000

(b)

Figure 2.4 Topological Design Traffic Flow Probability Matrices

Receiver allocation is a very simple and powerful method of performance evolution based upon the inherent broadcast property of the optical implementation. A simple example will be presented illustrating this capability. In receiver allocation, selected nodes are given additional receiver(s), and each receiver may be tuned to one of the existing Base PBNet channels.

When node j tunes an allocated receiver to a channel (i-1 i), node i assumes a 'shadow destination' responsibility for node j and performs a destination release function at that point on the network. This technique is simple and requires very little change to the media access protocols. A 6-node example is shown in Figure 2.5. In

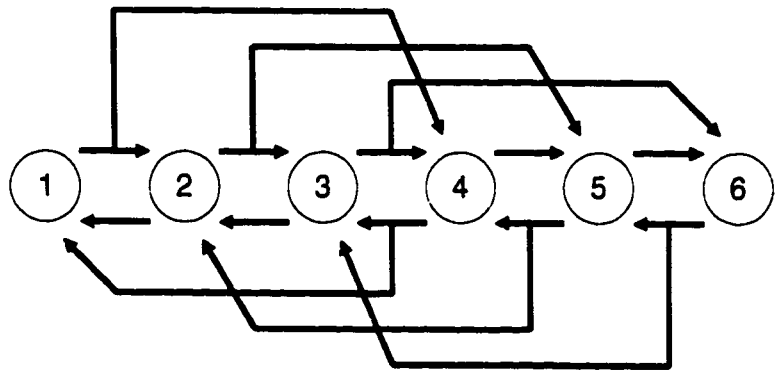


Figure 2.5 Receiver Allocation

this figure, we have given each node in the Base PBNet a single additional receiver. For example, the added receiver for node 6 is tuned to the (3 4) link. Thus, traffic originating in nodes 1 to 3 can reach node 6 as it makes the 3 to 4 link hop. When this happens, the cell in question has reached its destination and should not proceed further downstream. The shadow destination function in node 4 performs this cell release operation. In receiver allocation, each node maintains a table indicating the nodes for which it is a shadow receiver on both buses. When a cell arrives, the cell destination ID is used to index the shadow table. If the node in question is a shadow for this destination, the cell is marked idle. No other changes are required to the protocol.

Clearly, receiver allocation gives a very flexible way of upgrading the transmission capability of a PBNet without adding additional channels. In [Todd91a], several design algorithms were considered. By virtue of the optical star implementation, the station tuning for an allocated receiver may be chosen to optimize the network performance. In an N-node network, the NxN traffic flow matrix is defined by the set of f_{ij} for i, j ε {1, ... N}. Note that the nodes are numbered from 1 to N, and each f_{ij} corresponds to the total average flow rate in segments per slot transferred between nodes i and j. The objective of the algorithms is to determine the receiver tuning for a single allocated receiver added to each station. The design algorithms are summarized below:

(a) **Global Min/Max Algorithm**

1. Find the link in the network with the largest utilization. The station receiving the most traffic over this link, which has not yet been assigned a receiver tuning, is given the next receiver assignment.
2. Test the receiver tuning for this station at each possible location. For each case, note the maximum link utilization over the entire PBNet. Use the tuning that minimizes the maximum link utilization.
3. Repeat the above two steps until all receivers have been placed.

(b) **Local Min/Max Algorithm**

Each station independently tunes its allocated receiver to each possible location. For each tuning, the maximum link utilization is calculated based

upon its own traffic flow only. The tuning is used that gives the minimum of the maximum utilization.

(c) **Local Hop Algorithm**

Each station tunes its receiver to the location which minimizes the total segment-hop rate of traffic destined to it.

(d) **Max Node Algorithm**

Each station tunes its allocated receiver to the output link of the non-adjacent station which is sourcing it the most traffic.

A typical example of performance improvement is given in Figure 2.6. The traffic flow-probability matrix is given in Figure 2.7. The traffic flow matrix is obtained from this by multiplying each row by the total arrival rate for that node. A bursty traffic model was used, where single segment per packet Poisson arrivals were considered at each station. The arrival rates were taken to be identical for all nodes. An infinite input buffer was considered, and the results were obtained from an exact simulation of each system. It can be seen that a significant improvement in performance was attained by attempting to optimize the placement of the allocated receivers. Compared to a system with no allocated receivers, the capacity was increased by more than 100%. The performance improvement tended to increase with the size of the PBNets considered [Todd91a].

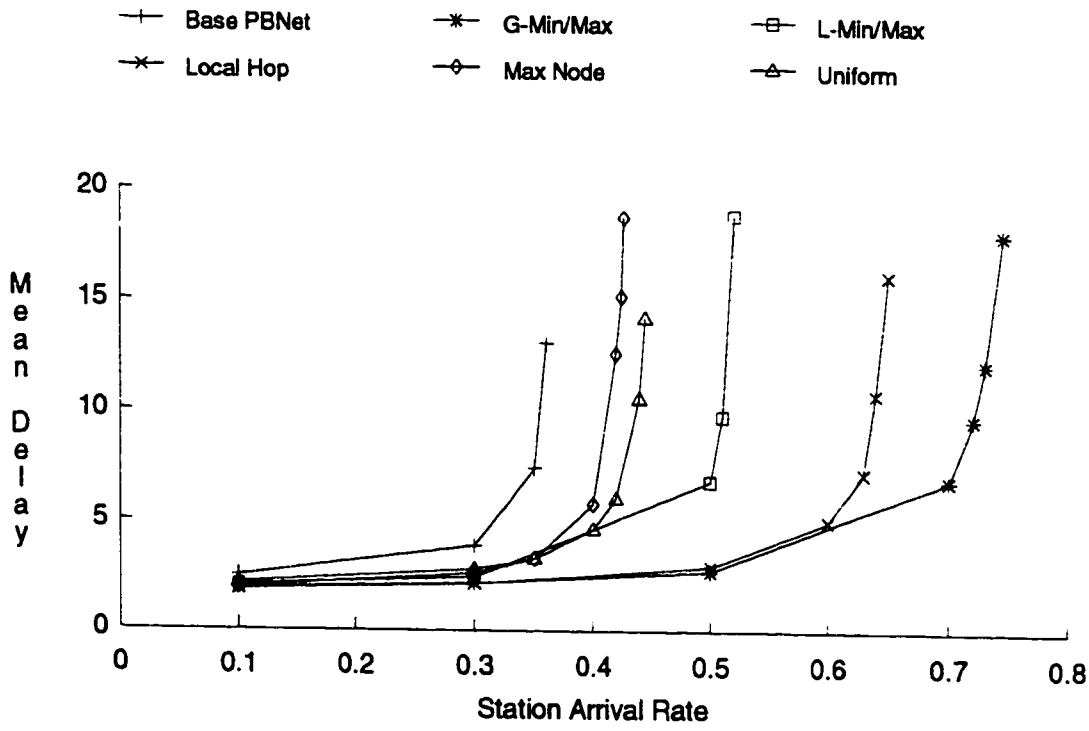


Figure 2.6 Receiver Allocation Algorithm Comparison (N=6)

0.0	0.11	0.1	0.31	0.09	0.39
0.0	0.0	0.1	0.79	0.1	0.01
0.01	0.0	0.0	0.79	0.2	0.0
0.0	0.1	0.6	0.0	0.1	0.2
0.0	0.0	0.2	0.2	0.0	0.6
0.0	0.0	0.0	0.8	0.2	0.0

Figure 2.7 Receiver Allocation Traffic Flow Probability Matrices

2.3 PBNet Media Access Protocol Design

In this section, a media access protocol is considered for photonic bus networks. Attention will be focused on possible slot reuse extensions to the distributed queueing mechanism introduced by the IEEE 802.6 DQDB protocol [IEEE90]. In particular, the 'request absorption' approach is considered. In the discussion, particular emphasis will be made on traffic overload performance. We are attempting to design a media access protocol that allocates bandwidth fairly during traffic overload, but which does not adversely affect performance during normal operating conditions.

In [Hahn90], the bandwidth balancing technique (BWB) was introduced to correct problems with bandwidth allocation during station overload. During the research, it was found that the impact of the BWB mechanism was not as obvious as first thought. For this reason, and in order to facilitate a more complete understanding of the following sections, the BWB mechanism will be presented here in a different light than traditionally done in the literature. The basic operation of the BWB mechanism can be introduced in the following manner.

In general, slotted networks can perform better than token passing networks since they are not dependent upon the passing of explicit tokens [Fine84]. However, one difficulty with slotted networks is the variable performance of the stations depending upon the position of the station on the bus. A major contribution of DQDB was to use the reverse bus to reserve access to the forward bus (and vice versa) which, under the ideal condition of zero propagation delay, provides fair access for all

stations to the bus bandwidth. A true First-Come-First-Served global queueing mechanism is established. In fact, if there is zero propagation delay between nodes and infinite reservation channel bandwidth, then DQDB

(a) never wastes any slots,

(b) services single slot messages in the order in which they arrive, and

(c) services multi-slot messages from several nodes one slot at a time in a round-robin discipline.

In a real application, however, a network may span up to 50 km utilizing a transmission rate of up to 150 Mbps. If a slot size of 53 bytes is selected in order to maintain compatibility with emerging ATM standards, then many slots may be propagating between adjacent stations on the bus. This leads to cases of extreme unfairness when several nodes are competeing for bandwidth as, for example, would happen when several stations are attempting to simultaneously transmit files across the network. The problem arises when one station begins to transmit in many consecutive slots during an idle period on the bus. The many slots propagating between adjacent stations are thus busy slots on one bus and the reservation bits on the other bus are also fully utilized. Any other station wishing to access the bus must first send a request to the busy station asking it to defer an empty slot downstream. It cannot send a second request until it has successfully transmitted the presently enqueued slot. The upstream station, upon receipt of the request from downstream, will pass one idle slot downstream and immeditely thereafter use all available slots for its own transmission. It is obvious that the upstream station in this example will completely dominate the bus bandwidth since the downstream station has no ability to break the upstream flow of busy slots. Much literature has documented the unfair behaviour of

DQDB in such situations [Hahn90, Hahn92, Cont89, Cont90, Brea90, Fili89, Wong89, Zuke90]. The implication is that, when propagation delay is considered, the request channel does not provide an explicit request. In fact, in many cases, by the time the request is received by other stations, the requesting station will have already transmitted in the first available idle slot. This leads to the consideration of the request procedure in DQDB as a mechanism that, in a steady state sense, allocates bandwidth between station. A request received by a station may not be an explicit request for bandwidth, but rather an indication that a downstream station is active and is in need of bandwidth. The amount of bandwidth needed will be indicated by the number of requests.

Hahne et al [Hahn90] proposed BWB as a simple way to overcome the unfairness of DQDB when several nodes are simultaneously competing for bandwidth. The technique has been extended to handle multiple priority traffic [Hahn91], although we will only consider the single priority case. The BWB technique uses a counter (which we call a trigger counter) that increments each time a slot is transmitted at a station. When a trigger counter reaches a threshold value, the node increments its RQ counter. This forces the node to let one idle slot propagate downstream each time the trigger counter 'fires'. After firing, the trigger counter is reset and the process starts again. The net effect of this procedure is to allow a feedback path between two competing stations. By 'wasting' bandwidth at one station (this is simply forcing a node to use only a fraction of the remaining bus bandwidth), you are allowing another station to request an increasing amount of bandwidth. If the two stations in question are competing, then they will both be 'rate controlled' to the same throughput. How

rapid the convergence to fair operation occurs, depends upon the amount of 'wasted' bandwidth.

An alternate manner in which to consider the BWB mechanism (which is particularly useful when considering slot reuse protocols), involves determining the system forcing function that drives the network to the rate controlled equilibrium point. First consider that the firing rate of the trigger counter is directly proportional to the throughput of the station. Each time the trigger counter fires, a station is forced to waste a slot. Therefore, the amount of bandwidth that a station is forced to waste, in absolute terms, is directly proportional to the throughput of the node. If multiple stations are competing for bandwidth, the stations that are utilizing more of the bandwidth are forced to give up slots to the other stations. These other stations will utilize the additional slots, and therefore increase the firing rate of their trigger counter. This will allow competing stations of lower throughput to 'throttle' competing stations of higher throughput by utilizing more bus bandwidth or increasing the flow of requests. The process will continue until the trigger counters in the competing stations are firing at the same rate. At this point, the throughput of the competing stations are rate controlled to the same level.

For the same reasons as in standard DQDB, bandwidth balancing is also highly desirable in a slot reuse system. However, since slots may now be reused along the bus, it is necessary for requests to be absorbed and reused on the reverse bus. (Note that if requests are not absorbed, the equilibrium forward bus throughput would be limited to the maximum throughput of the request bus; namely one segment per slot,

thus providing a severe performance bottleneck.) In [IEEE90], it was shown that a set of linear equations could be solved to determine the relative station throughputs in an overloaded network with different node priorities. It is also possible to analyze the overload performance of slot reuse protocols in a similar fashion. First, consider a network of N-nodes numbered 1 through N where node i is considered downstream of node j if i>j. We will assume that a subset of the nodes is active and experiencing overload conditions. All other nodes are assumed to be idle. We are interested in a protocol's bandwidth deployment under worst-case overload situations where all active nodes are experiencing rate-control. For each active node i, define d_i to be the destination node of station i's slot transmissions. The analysis is as follows.

For each active node i, write the following Overload Balance Equation (OBE).

$$S_i = \alpha(1 - F_i - R_{i+1}) \tag{2.3.1}$$

where

$$F_i = \sum_{j=1}^{i-1} S_j \, u(d_j - i)$$

$$R_{i+1} = \sum_{j=i+1}^{N} (S_j - K_j) \tag{2.3.2}$$

Note that u(i) = 1 for i>0 and 0 otherwise. In the above equation, S_i is the equilibrium overload throughput attained by node i, R_{i+1} is the equilibrium request rate of all nodes downstream of node i, and K_i is the rate at which requests are absorbed at node i. Also, $\alpha = B/(B + 1)$ where B is the bandwidth balancing counter trigger value

[Hahn90], and F_i is the total flow of busy slots passing station i, which were generated by stations upstream of station i.

It is apparent how Equation (2.3.1) has been written. When bandwidth balancing is used, a node in equilibrium always uses a fraction α of the residual bandwidth after both upstream and downstream needs are accounted for. This equilibrium is reflected in Equation (2.3.1) and takes into account the possibility of request reuse at each node. The above formulation gives a set of M equations in the M unknowns S_i, where M is the number of overload stations. This formulation will be used in the subsequent discussion.

It is also necessary to consider the *optimum* behaviour of a Base PBNet under overload conditions. Unlike a system without slot reuse, it is not always obvious what the desired station overload flows should be. We now define the Link Fairness Criterion (LFC). Under LFC, a transmitting node should always rate-control itself so that it assumes no more than an equal share of residual bandwidth over each link, when more stringently rate-controlled node flows are accounted for first. More stringently rate controlled nodes are those nodes that are rate controlled to a lower flow rate than the node in question. This fairness criterion is reasonable, since it treats each link as a resource whose residual bandwidth must be shared equally by competing nodes. To calculate the LFC flow assignment, consider each link in decreasing order of the number of nodes competing for it. At each step, subtract the flows of more stringently controlled nodes from the total capacity. Then assign the residual bandwidth equally amongst the remaining nodes competing over the link. It

should be noted that LFC is identical to the Max/Min fairness property in wide area network flow control [Bert87].

We will now consider a brief discussion of the request-kill approach to slot reuse. This technique is motivated by past submissions to the IEEE 802.6 standards group concerning destination release mechanisms associated with erasure node operation. This is discussed in Part (d) below. We first present some examples of slot reuse protocols that are trivial extensions to IEEE 802.6, focusing on their overload flow problems. The objective is to illustrate the use of (2.3.1) and to introduce various problems associated with slot reuse under overload. Comparisons of various algorithms under a bursty traffic model was given in [Todd91, Todd91a].

(a) **Conventional DQDB**

This is a photonic multihop implementation of the IEEE 802.6 Metropolitan Area Network standard. In accordance with the standard, no bandwidth reuse is permitted at the nodes. This protocol is included for comparative purposes and its detailed description may be found in [IEEE90]. As an example of overload bandwidth deployment, consider a set of 3 overloaded nodes labelled 1, 2 and 3. Since destination release is not possible, for modelling purposes we assume that $d_i = 4$ for all i, and thus all nodes have competing transmission paths. In addition, since requests propagate to the end of the reverse bus, $K_i = 0$ for all i, with $R_3 = S_3$, $R_2 = S_2 + S_3$ and $R_1 = S_1 + S_2 + S_3$. The OBEs for nodes 1, 2 and 3 are therefore

$$S_1 = \alpha(1 - S_2 - S_3) \quad (2.3.3)$$

$$S_2 = \alpha(1 - S_1 - S_3) \quad (2.3.4)$$

$$S_3 = \alpha(1 - S_1 - S_2) \quad (2.3.5)$$

Solving the above equations we obtain the well known result [Hahn90], namely $S_i = \frac{\alpha}{1 + 2\alpha}$. In this conventional case, bandwidth balancing has split the bandwidth equally amongst the competing stations. This bandwidth allocation is also consistent with the LFC.

(b) **Random Slot Reuse**

In the Random protocol, slots are used by the nodes in a greedy fashion and no request mechanism is used. Destination slot release is employed however. A free slot arriving at a busy station is immediately used without any regard for downstream stations. The above results may be modified and applied for the Random algorithm. Since no request mechanism is used, $R_i = 0$ for all i. In addition, due to its greedy nature, $\alpha = 1$. Thus, for a set of active competing nodes 1 to M, $S_1 = 1$ and $S_i = 0$ for i>1, giving extreme unfairness. This protocol is of interest because there is no scheduling overhead, and in many uniform traffic situations, it has been shown by exact simulation to attain the best observed global delay-throughput performance [Todd91]. The simulations included the queueing delay required to access the network.

(c) **DQSRU**

In this protocol, DQDB is modified so that slots are released at their destinations for downstream use. DQSRU represents the most straightforward change to DQDB resulting in slot reuse. Clearly this system will give the same overload performance as DQDB when no slot reuse is possible. We consider an example that illustrates the major deficiency of this protocol. Consider four nodes - 1, 2, 3 and 4. Node 1 transmits to node 2 (where the slots are released) and node 3 transmits to node 4. Ideally, we would hope that a 'good' reuse protocol would permit close to the entire bus bandwidth for each of nodes 1 and 3 in this case. However, writing the equations one obtains $S_3 = \alpha = R_3$ and $S_1 = \alpha(1 - R_3) = \alpha(1 - \alpha)$. Notice that in this case, we have transposed the unfairness of the Random protocol in that as $\alpha \rightarrow 1$, the most downstream active station consumes most of the bandwidth. In fact, it can be shown that for a station competing with n downstream nodes in this fashion, its throughput is given by $\alpha(1 - \alpha)^n$. Obviously, neither Random nor DQSRU protocols yield the desired LFC assignments. It is clear that in overload situations that involve bandwidth reuse, the request channel itself may quickly become a bottleneck. In equilibrium, the total slot throughput on the forward channel can never exceed the request throughput on the reverse channel. In the above example, we would like to see a total throughput approaching 2 on the forward channel. However, since the request channel can at most process 1 request per slot, the total forward channel throughput can never exceed this value. This limitation may be overcome in a number of ways. The most obvious method for increasing the request channel capacity is

to permit request channel reuse. This is the concept that motivates the next protocol.

(d) **REQKILL**

This protocol is a modified version of the erasure node proposals submitted to [IEEE90]. REQKILL implements request channel reuse by having destination stations delete requests on the reverse channel. A 'request kill' counter (RKC) is maintained for each bus at every node. Whenever the node clears a busy forward data slot, the reverse RKC is incremented. A non-zero value for RKC allows a station to delete requests on the appropriate channel and to decrement RKC each time it does so. Note that unlike [IEEE90], RKC is maintained independently of the RQ status of the node.

The request-kill approach performs very well in many types of PBNet overload situations. An example will be given, however, that illustrates a basic incompatibility with the bandwidth balancing counter implementation. Consider the overload situation shown in Figure 2.8.

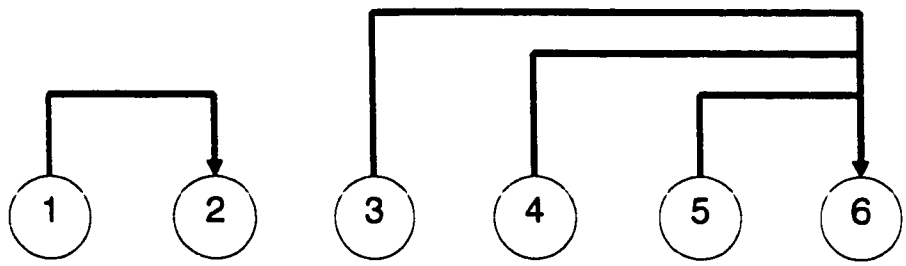


Figure 2.8 6-Node Overload Flow Example

We have node 1 transmitting to node 2, while nodes 3, 4 and 5 compete downstream for the bus. Intuitively, we would have node 1 assume the full channel bandwidth with nodes 3, 4 and 5 equally splitting the same (reused) bandwidth downstream. However, we will see that this is not possible. The overload balance equations are given as follows.

$$\begin{aligned} S_1 &= \alpha(1 - S_3 - S_4 - S_5 + K_2) \\ S_3 &= \alpha(1 - S_4 - S_5) \\ S_4 &= \alpha(1 - S_3 - S_5) \\ S_5 &= \alpha(1 - S_3 - S_4) \end{aligned} \tag{2.3.6}$$

Here, it can be seen that the rate at which slots are 'killed' at node 2 is given by

$$K_2 = \min(S_1, S_3 + S_4 + S_5) \tag{2.3.7}$$

Since the values of the S_i's are unknown, we must test both possibilities for K_2. It is easily seen that $K_2 = S_1$ is the consistent and correct result, with the other option leading to a simple contradiction. Solving the equations completely we find that $S_1 = S_3 = S_4 = S_5 = \frac{\alpha}{1 + 2\alpha}$. We see that instead of using the total bandwidth, node 1 is rate-controlled to the *same* throughput as nodes 3, 4 and 5! Thus, as $\alpha \rightarrow 1$, $S_1 \rightarrow 1/3$. It is obvious that the total bandwidth should be available to node 1 under LFC. This illustrates a basic problem with the request-kill approach. Under certain conditions, rate-controlled downstream stations can wrongfully inhibit upstream stations from using the available bandwidth. This introduces undesirable coupling between otherwise disjoint communities of interest on the network. For example, if 10 nodes were competing downstream of two competing upstream nodes, *all* nodes would be rate-controlled to a maximum throughput of 10% (rather than the desired 50% for each of the upstream nodes). This unacceptable behaviour can be verified easily by writing the overload balance equations.

It is instructive to discuss the details of the example just given (again, see Figure 2.8). Ideally, we would like to have node 1 use the entire channel bandwidth. Let us assume that this is initially the case, with nodes 3, 4 and 5 using 1/3 of the bandwidth each. We will give an argument to show that node 1 is quickly rate-controlled to a throughput value of 1/3. Since node 1 is initially using bandwidth at a rate of 1 slot per slot, its bandwidth balancing counter will be 'timing out' at a rate 3 times that of node 3. Each time this occurs, an idle slot will be sent to node 2 in accordance with the usual BWB mechanism. When this happens, the RKC in 2 fails

to increment, resulting in the passage of a request upstream to node 1. This is the important action to note. In allowing an idle slot to pass from node 1 to 2 due to the BWB mechanism, the ability of node 2 to kill requests has been affected. As a result, node 1 recognizes a request from upstream and increments its RQ counter to defer bandwidth downstream. This situation repeats itself until the BWB counter timeout rate of node 1 is equal to that of node 3. At this point, node 1 has been rate-controlled to the same bandwidth level as node 3! We can see from this example the fundamental dilemma. Node 1 has no way of knowing whether a request that penetrates past node 2 is from a node situated *between* nodes 1 and 2 or from a node beyond 2. Its proper reaction to the request is different in both cases.

In the following sections, new protocols are introduced that do not have these shortcomings. In these proposals, requesting nodes must identify the source of the request. In this way, upstream nodes can make unambiguous decisions concerning the validity of bandwidth requests from downstream. Bandwidth-balancing counters are used to ensure an equitable sharing of bandwidth. The conventional DQDB protocol is a special case of the new proposals, in that when an overload situation occurs where no bandwidth reuse is possible, the system behaves identically to DQDB.

2.3.1 The REQPASS Protocol

We have shown that problems may arise in a request-kill slot reuse protocol due to the fact that downstream requests are not identified as to their originating nodes. In this section, a generalization of the DQDB and REQKILL protocols is presented that does not have these shortcomings. This system is referred to as the REQPASS Protocol. All of the protocols discussed in this section have been restricted to the single priority case. A multi-priority version of the REQPASS protocol should be considered in a future work.

In REQPASS, each station i is equipped with a RKC counter as in the REQKILL approach. In addition, requests include the station ID of the requesting node. This is referred to as the Request ID. We will focus our attention on the forward bus operation only. In addition to the RKC counter, stations also have a Destination Trigger Register or DTR. The DTR is used by a station to store the node ID of a downstream station under certain conditions. Note that as in DQDB, each node maintains the RQ, CD and bandwidth balancing counters. The new features of the protocol are described as follows.

(a) As in REQKILL, the RKC (request kill counter) is incremented with each slot released. The clearing of requests and decrementing of RKC are also performed in the same fashion as in REQKILL.

(b) Whenever there is a BWB counter timeout, the DTR is loaded with the destination station ID of the slot being transmitted. In this way, the DTR

samples all destinations for the node according to their relative frequency. The DTR therefore defines the downstream nodes with which the local node will compete for bandwidth, during each BWB counter timeout interval.

(c) When a request arrives on the reverse channel, the request ID is compared with the contents of the DTR. If the DTR is not empty and its contents are less than or equal to the request ID, then the RQ counter at the station is *not* incremented and the DTR is cleared immediately. The request is then processed as in REQKILL. That is, it is either killed or passed upstream, depending upon the RKC state.

In REQPASS, the slot transmission algorithm is identical to that in DQDB. The use of the bandwidth balancing counter is also the same. The purpose of the DTR is to decouple individual or groups of rate-controlled stations which do not have competing bandwidth requirements. At the same time, it permits bandwidth balancing amongst stations which have competing requests.

It is important to note that REQPASS is based upon matching request and slot reuse rates, and thus attempts to provide equitable bandwidth levels across the network. This philosophy is in contrast to attempting to match requests to reused slots on a one-to-one basis.

The overload performance of REQPASS is now briefly discussed. Because the protocol identifies the source of requests, a node only allocates upstream bandwidth to

a downstream station when that request cannot be satisfied by the reuse of its own or any upstream transmissions. Again consider the example of Figure 2.8. Requests generated by nodes 3, 4 and 5 are cleared at node 2 at a rate of α per slot. Thus the request flow rate reaching node 1 is $\frac{2\alpha(1 - \alpha)}{1 + 2\alpha}$. However, the maximum rate at which node 1 can ignore those requests is larger and given by $1 - \alpha$. Thus node 1 is completely decoupled from nodes 3, 4 and 5. Also, the total request flow rate upstream of node 1 is given by $\frac{3\alpha}{1+2\alpha}$, which is the desired aggregate rate generated within this network. Note that whenever a request from 3, 4 or 5 is passed upstream by node 2, the DTR will contain the destination ID of node 2. This will prevent node 1 from allocating idle bandwidth to the request, thus decoupling itself from the actions of nodes 3, 4 and 5. As a result, the total bandwidth may be used by node 1. An exact simulation of the REQPASS protocol has been used to validate the overload performance in this and many other known problem situations.

Another brief example is given. An overload pattern involving 8 nodes is shown in Figure 2.9. By writing and solving the overload balance equations for the REQKILL protocol, we find that all nodes are rate-controlled to a throughput of 1/3 as $\alpha \rightarrow 1$. Using the LFC, we first consider link 5-6. This gives $S_5 = S_4 = S_3 = 1/3$. Moving to link 4-5 and using the assignments for nodes 3 and 4, we obtain $S_2 = 1/3$ also. Then at link 2-3, we first subtract out S_2, leaving a throughput of 2/3. This value is split evenly amongst the (single) remaining node, giving $S_1 = 2/3$. It is clear

that REQKILL does not yield overload flow assignments that are consistent with LFC. When this happens, the residual bandwidth is simply wasted. REQPASS, however, permits node 1 to achieve a throughput of 2/3 while rate-controlling all others to 1/3. While a formal proof will not be presented, it can be argued that since REQPASS nodes always ignore requests that they or upstream nodes can satisfy, the overload flows obtained are generally consistent with LFC. This has been validated in many of the examples we have verified through exact simulations of the protocol.

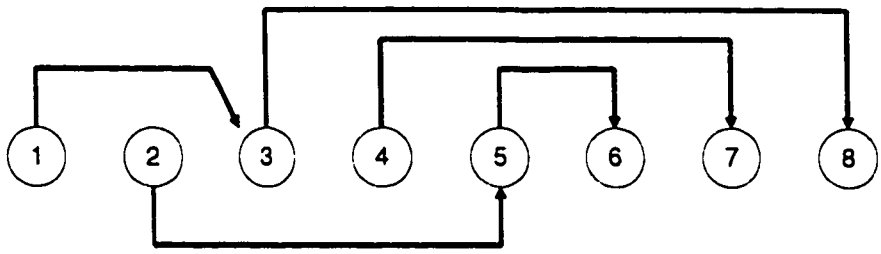


Figure 2.9 8-Node Overload Flow Example

The protocol is simple to implement since it implies the addition of only one register at each station for each bus. In effect, this register determines, for each interval of the BWB counter firing, the reach of downstream stations with which the station will compete. By ignoring a request originating downstream of the DTR, the station is decoupling its BWB counter with those downstream of the DTR. This eliminates the forcing function, which has been discussed, for rate controlling nodes.

On the face of it, this implies that a station competes with nodes upstream of the DTR and does not compete with nodes downstream from the DTR. In a majority of the tested situations, this was the case. It is possible, however, to define a competing situation where, through an intermediate node upstream of the DTR value, a request (which would have been ignored by an upstream station in order to decouple its BWB counter) is killed by the intermediate station and replaced with a request that will not be properly ignored. As a result, the BWB counters of stations that are not competing remained coupled, and the LFC is not achieved. This is a 'second order' effect, the likelihood of which is dependent upon the traffic flows on the network. The REQPASS protocol is simple and alleviates the known problems with the REQKILL protocol. A more complicated protocol design, which will be reviewed in the following section, can alleviate the additional problems arising from such 'second order' effects.

2.3.2 The Selective Killing Protocol (SELKILL)

We begin discussion of this more complicated protocol by considering Figure 2.10. In the figure, there are two general regions of competition on the forward bus.

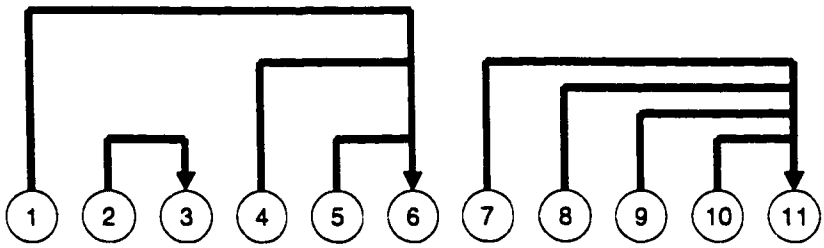


Figure 2.10 11-Node Overload Example

The first involves nodes 1 through 5 and the second involves nodes 7 through 10. Ideally the two competing groups should not interact, since all slots utilized in the first group are freed up at node 6 for use by nodes 7 through 10. Using the same mechanisms as in REQPASS, in particular RKC and DTR, nodes 1, 2, 4 and 5 can ignore one request originating upstream from node 6 for each firing of their BWB counter. This function performs the BWB decoupling that has been referred to in REQPASS. Ideally, according to the LFC, the node throughputs should be $S_1 = S_4 = S_5 = 0.33$, $S_2 = 0.67$ and $S_7 = S_8 = S_9 = S_{10} = 0.25$. We will now

introduce a phenomenon called 'replacement killing', which defeats the REQPASS mechanism and leads to flows other than those determined by the LFC.

We refer once again to Figure 2.10. The replacement killing mechanism takes place at nodes 2 and 3. Requests flowing from nodes upstream of node 6 are largely killed at node 6. Those which are not killed do not, in most cases, impact nodes 1, 4 and 5 since RQ, in accordance with the DTR, is not incremented at those nodes (see the REQPASS protocol description). The addition of an active source of requests at node 2 and the ability to kill requests at node 3 changes this situation. Consider the effect at node 1 of killing requests at node 3 and recall that for every request killed at node 3, there has been a request generated at node 2. Three cases may arise.

(a) **A Node 4 or 5 Request is Killed at Node 3**

This case has no impact at node 1. Requests killed at node 3 are replaced with requests by node 2 which must be recognized by node 1. This is appropriate since node 1 must recognize requests from nodes 4 and 5 anyway.

(b) **Complete Killing at Node 3**

In this case, node 3 has enough kill capability to kill all requests. There would be no impact at node 1 because if node 3 kills all requests, it must be releasing enough bandwidth for upstream stations and node 1 does not need to recognize any requests.

(c) **A Node 7, 8, 9 or 10 Request is Killed at Node 3**

This is the 'replacement killing' situation that adversely affects node 1. For the purposes of this discussion, the flow of requests from nodes 7, 8, 9 or 10 that are not killed at node 6 will be referred to as a residual request. First note that the flow of requests on the reverse bus is stochastic. An observer on the reverse bus would see requests from a variety of sources in an order that is dependent upon the arrival processes of packets to the nodes and the availability of the request channel at the nodes. If node 3 cannot kill all requests, it may (at times) kill a residual request. Recall that node 2 replaces requests killed at node 3 with requests that directly impact node 1. Due to the REQPASS mechanism, node 1 was prepared to ignore a residual request, but due to the replacement killing, it was unable to do so. In this situation, kill capability, which was used to kill the residual request, is not available to kill a request from node 4 or 5. The net result is that, as seen at node 1, a flow of one request that should be ignored and one request that should be recognized, has been replaced with two requests that must be recognized. This alters the flow of requests so that a feedback mechanism is created between node 1 and nodes upstream from 6. The outcome, as verified by exact simulation, is a set of steady state flows that are not in accordance with the LFC ($S_1 = S_7 = S_8 = S_9 = S_{10} = 0.2$, $S_4 = S_5 = 0.4$ and $s_2 = 0.8$). It is possible to define other situations similar to Figure 2.10 that produce very unfair behaviour due to the replacement killing phenomenon.

The solution to this problem involves adding more intelligence to the killing procedure at a node. We do this by defining a set of counters at each node that will determine the reach, much the same as DTR in REQPASS, for the killing process. The basic premise is that if a node uses its kill capability to kill requests from nodes closest to it first, then the request replacement phenomenon cannot occur. Consider Figure 2.11 for a more general argument of this point. We are concerned with the possible impact of replacement killing on node i. The range of competition for node i is defined by the destination, in this case node l.

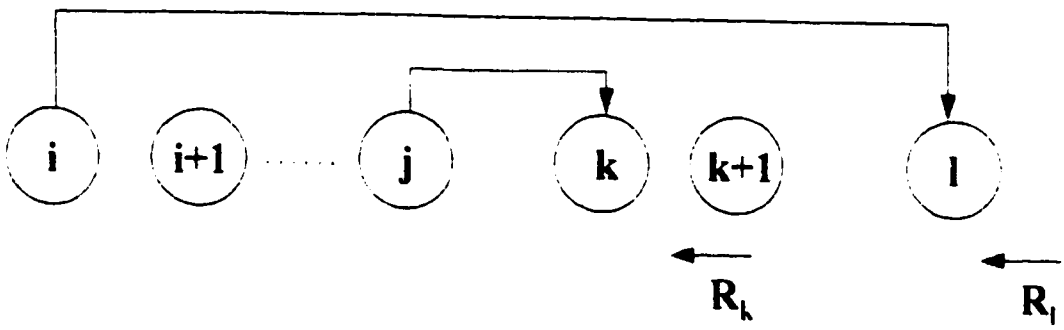


Figure 2.11 Replacement Killing

For replacement killing to occur, we require a traffic flow that both originates and terminates within the range of competition for node i. This is shown as nodes j and k and is the potential source of replacement killing. Define R_n as the total flow of requests into node n. Also define K_n as the total kill capability of node n. For replacement killing to occur, we must have $K_k<R_k$ and have a request from R_l killed at

node k (see Figure 2.11). If we impose selective killing on node k, so that it must kill all flows from the closest nodes first (flows from node n where $k < n \leq l$ in Figure 2.11), then, by definition, replacement killing cannot occur.

The implementation of selective killing is as follows. Assume there are N nodes in the PBNet. We will discuss node i on the forward bus only. The same operations apply on the reverse bus. Node i has a set of counters called Forward Reach Counters FRC[i] representing nodes $i < j \leq N$. Each time a request is received on the reverse bus from node j, then FRC[k] is incremented for $i < k \leq j$. FRC[j] therefore gives the cumulative request flow for all nodes downstream of node i, up to and including node j. For every slot released on the forward bus at node i, decrement FRC[k] for $i < k \leq N$. Note that at all times there will be some number n, where $i < n \leq N-1$, FRC[k]<0 for $i < k \leq n$ and FRC[k+1]>0 for n<k<N. Also note that, due to the evolution of the counters, they define the range of node requests that may be killed at a node i while still preserving the 'closest first' relationship, which is necessary to prevent replacement killing. In our discussion, node i has enough kill capability to kill requests up to node n and possibly some from node n+1. The following algorithm defines the operation of selective killing at node i. Note that the Forward Reach Counters are used to make the killing of requests selective, but are not associated with the killing itself.

When RKC>0

1) Kill any request from node k if $FRC[k-1] \leq 0$.

2) Do not kill any other requests.

When $RKC \leq 0$

Define node n where FRC[k]<0 for $i < k \leq n$ and FRC[k+1]>0 for n<k<N

1) If a node k request passes where $FRC[k] \leq 0$, then it should be killed and replaced with a request from node n+1.

2) Pass all other requests.

An exact simulation of the SELKILL protocol was used to verify correct performance according to the LFC for a variety of known problem situations, including that of Figure 2.10. It has performed correctly in all tests run to date.

In the next section, an analytic model is given for calculating the individual node delays obtained in a PBNet operating either the REQPASS protocol or the SELKILL protocol. It will be shown that under a Poisson traffic model, the technique proposed can accurately predict the mean delays in the system.

2.4 An Analytic Station Delay Model

A number of authors have considered the analytic performance of conventional DQDB without slot reuse [Bisd90, Mukh90]. In this development, we focus on the individual station delays under the REQPASS and SELKILL protocols introduced in the previous sections. For convenience, we will again consider only the forward bus. Each of N stations maintains an infinite input buffer fed by a Poisson arrival process with mean λ_i for i ε {1, 2, ... N}. Single segment packet arrivals to station i are assumed to be destined to downstream station k with probability d_{ik}. At node i, we define

$$f_i = \sum_{k=1}^{i-1} \sum_{j=i+1}^{N} \lambda_k \, d_{kj} \qquad (2.4.1)$$

Note that f_i gives the total flow rate of segments that pass by station i on the bus from upstream stations. We also define R_i to be the total flow rate of requests leaving station i, including those from stations that are downstream of i. R_i can be calculated by noting that

$$R_i = \max(\lambda_i + R_{i+1} - \sum_{k=1}^{i-1} \lambda_k \, d_{ki} \, , \, 0) \qquad (2.4.2)$$

The request kill capability of node i has been incorporated into the above expression. Since station N does not transmit on the forward bus, $R_N = 0$ and Equation (2.4.2) can be solved recursively by decreasing i starting with i = N. Note that the rate at which station i yields to requests from downstream is easily calculated and given by

$$r_i = \max(R_{i+1} - \max(\sum_{k=1}^{i-1} \lambda_k d_{ki} - \lambda_i , 0) , 0) \qquad (2.4.3)$$

The nested maximum in this expression gives the 'residual kill' rate of station i after it has erased its own self-enqueued requests. This residual is then applied to the requests that have arrived from stations downstream. The difference is the rate at which station i must defer idle bandwidth downstream.

The effective service time of a packet is defined from the time it becomes self-enqueued at the station until it is transmitted to completion on the forward bus. An exact calculation of the effective service statistics of the packets is very difficult. It can be seen, however, that the effective service time experienced by a segment in the presence of a local queue backlog is considerably different from that of a segment arriving to an empty queue. In the former case, the initial CD counter value after the segment is self-enqueued has resulted from a sampling of the request bus where valid requests increment RQ only (and idle slots decrement CD). In the latter case, however, RQ is both incremented and decremented while the queue is empty, prior to the first segment arrival. Thus, the expected value of RQ and subsequent effective service time distribution is much different. Accordingly, the station is modeled as an M/G/1 queuing system where the initiator of a busy period receives exceptional service [Haye84]. In order to use this model, it is necessary to apply two simplifying assumptions. The assumptions will be stated below.

Consider the effective service time seen by an arrival to an empty local queue. Define $\bar{m}$ to be the effective service time of a tagged segment arrival. We will

assume that the arrival samples the equilibrium distribution of the local RQ counter. This value is transferred to the CD counter upon arrival of the segment. The distribution of the RQ counter under these circumstances will be approximated as follows. Define $n_{RQ}(i)$ to be the state of the RQ counter during slot i. In the absence of any queued segments, the RQ counter at the node increments for every reverse bus request and decrements for each idle slot passed on the forward bus. It will be assumed that the probability of each is determined independently from slot to slot by r_i and f_i respectively. In this formulation, a $(1-\alpha)$ is appended to the expression for f_i in order to account for the loss of bandwidth due to the BWB mechanism. The state of the RQ counter is thus described by a discrete-time Birth-Death Process as shown in Figure 2.12.

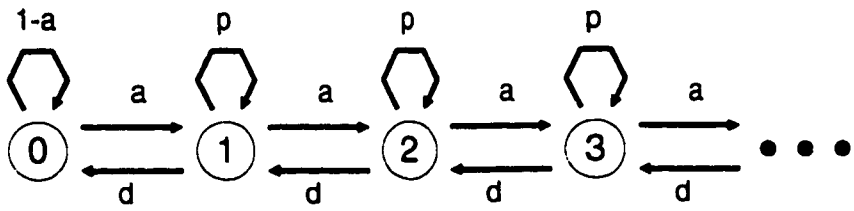


Figure 2.12 Birth-Death Process

In the figure, we have made the following definitions

$$\begin{aligned} a &= r_i f_i \\ d &= (1 - f_i)(1 - r_i) \\ p &= r_i(1 - f_i) + f_i(1 - r_i) \end{aligned} \tag{2.4.4}$$

Solving the system in the usual way gives the equilibrium distribution of the RQ counter state under the stated modelling assumptions, namely

$$p_n \equiv P(n_{RQ} = n) = (1 - \frac{a}{d})(\frac{a}{d})^n \tag{2.4.5}$$

This expression will be used to approximate the first two moments of the effective service time of a segment arriving to an empty local queue. Once the segment arrives, it self-enqueues and the value of the RQ counter is transferred to CD. Clearly, the total effective service time is given by

$$\tilde{m} = T_W + T_{WCD} \tag{2.4.6}$$

where T_{WCD} is the time required for the CD counter to decrement to zero and T_W is the time to wait for an empty slot once the CD counter has reached zero. Invoking the traffic independence assumption (as done when deriving (2.4.4)), it can be seen that the distribution of the former is geometric, given by

$$P(T_W = k) = f_i^{k-1}(1 - f_i) \tag{2.4.7}$$

for $k > 0$. In addition, it can be seen that if $n_{RQ} = j$ upon arrival to the queue, T_{WCD} is the sum of j independent geometrically distributed intervals, each with the same

distribution as T_W. Accordingly, the conditional moments of the effective service time are given by

$$\begin{aligned} E[\tilde{m} \mid n_{RQ} = n] &= (n + 1)\bar{T}_W \\ E[\tilde{m}^2 \mid n_{RQ} = n] &= (n + 1)\,\overline{T_W^2} + n(n + 1)\bar{T}_W^2 \end{aligned} \tag{2.4.8}$$

Equations (2.4.5) and (2.4.7) can then be used to solve the above expressions. This results in

$$E[\tilde{m}] = \frac{1-r_i}{1-r_i-f_i} \tag{2.4.9}$$

and

$$E[\tilde{m}^2 \mid n_{RQ} = n] = (n+1)\frac{1+f_i}{(1-f_i)^2} + \frac{n(n+1)}{(1-f_i)^2} \tag{2.4.10}$$

Removing the condition and manipulating, we obtain

$$E[\tilde{m}^2] = \frac{1+f_i}{1-f_i}\,E[\tilde{m}] + \frac{2r_if_i(1-r_i)(1-f_i)}{(1-r_i-f_i)^2} \tag{2.4.11}$$

Note that the development thus far has modeled the system in continuous time. To correct for this, we define the actual mean and second moment of effective service time seen by the busy period initiator as $\tilde{x} = \tilde{m} + \tilde{v}$. The extra term is due to the time an arrival must wait for the start of the next slot. It can be readily shown that

$$pdf_{\tilde{v}}(\tau) = \frac{\lambda e^{-\lambda(1-\tau)}}{1-e^{-\lambda}} \tag{2.4.12}$$

for $0 < \tau < 1$. Using this, the first two moments can be calculated and are given by

$$\begin{aligned} E[\tilde{v}] &= \frac{1}{1-e^{-\lambda}} - \frac{1}{\lambda} \\ E[\tilde{v}^2] &= \frac{2}{\lambda^2} - \frac{2-\lambda}{\lambda(1-e^{-\lambda})} \end{aligned} \tag{2.4.13}$$

Thus we have that

$$\begin{aligned} E[\tilde{x}] &= E[\tilde{m}] + E[\tilde{v}] \\ E[\tilde{x}^2] &= E[\tilde{m}^2] + 2E[\tilde{m}]E[\tilde{v}] + E[\tilde{v}^2] \end{aligned} \tag{2.4.14}$$

The effective service time for segments arriving to a non-empty local queue will now be considered. A development similar to the above will be undertaken. That is, we will attempt to determine the state of the RQ counter seen by segments that arrive to a non-empty queue. Recall that it is this value of the RQ counter that will be copied to the CD counter when a segment is self-enqueued. In addition, at that time, the RQ counter is reset to zero and begins to monitor requests on behalf of the next segment to enter the global queue at that station. Define this point in time as the enqueuing point.

We start with consideration of our geometric flow assumption. Assume a segment enters the global queue when the RQ counter is equal to n. Since, by assumption, the flow of idle slots on the forward bus is geometrically distributed, it will take n+1 iid geometric intervals before the CD counter decrements to zero and the segment is transmitted on the bus. Each interval i will consist of a number, say k_i, of slots determined by the geometric flow. Define m as the total number of slots needed to transmit the segment. m may be stated as

$$m = k_1 + k_2 + k_3 + \ldots + k_{n+1} \tag{2.4.15}$$

The distribution of m is required in order to determine the moments of segment service time. The distribution of k in each interval is given by

$$P(k=i) = f^{i-1}(1 - f) \tag{2.4.16}$$

as in (2.4.7).

The distribution of m may therefore be determined by an (n+1)-fold convolution of (2.4.16). Performing this operation allows us to define the probability of a segment requiring m slots to be serviced, conditional on the RQ counter being equal to n at the previous enqueuing point. This probability may be expressed as

$$P(m=i \mid RQ=n) = \begin{cases} \binom{i-1}{n}(1-f)^{n+1}f^{i-1-n} & f > 0\ ,\ i \geq n+1 \\ 0 & f > 0\ ,\ i < n+1 \\ 1 & f = 0\ ,\ i = n+1 \\ 0 & f = 0\ ,\ i \neq n+1 \end{cases} \tag{2.4.17}$$

The value of the RQ counter at the time that a packet enters the global queue will be determined by the service time of the packet that was enqueued in front of it, and the

flow of requests on the reverse bus. We may therefore write an expression, similar to (2.4.17), for the probability of the RQ counter having a certain value at an enqueuing point, given that the previously enqueued packet took m slots to service.

$$P(RQ = j \mid m = i) = \begin{cases} \binom{i}{j} r^j (1-r)^{i-j} & r > 0 \ , \ j \le i \\ 0 & r > 0 \ , \ j > i \\ 1 & r = 0 \ , \ j = 0 \\ 0 & r = 0 \ , \ j \neq 0 \end{cases} \tag{2.4.18}$$

(2.4.17) and (2.4.18) may now be used to define the probability of the RQ counter having a certain value at one enqueuing point, say t, given its value at the previous enqueuing point, say t-1.

$$P(RQ_t = j \mid RQ_{t-1} = n) = \sum_{i=n+1}^{\infty} P(RQ_t = j \mid m = i) P(m = i \mid RQ_{t-1} = n) \tag{2.4.19}$$

This allows us to define a set of linear equations relating the value of the RQ counter and the conditional probability of (2.4.19).

$$P(RQ_t = j) = \sum_{n=0}^{\infty} P(RQ_t = j \mid RQ_{t-1} = n) P(RQ_{t-1} = n) \tag{2.4.20}$$

Equation (2.4.20) defines the probability of the RQ counter state in terms of the conditional probability of (2.4.19) and the RQ counter state. Assuming equilibrium, the probability of the RQ counter state may thus be isolated in the expression to arrive at the required set of linear equations.

We solve (2.4.20) by assuming that the distribution may be adequately represented by considering only the first N+1 counter states. The normalizing condition of (2.4.21) will be introduced into the N+1 equations.

$$P(RQ = 0) = 1 - \sum_{i=1}^{N} P(RQ = i) \tag{2.4.21}$$

The resulting N equations are of the familiar form Ax=y, where

$$a_{ij} = \begin{cases} 1 + P(RQ_t = i \mid RQ_{t-1} = 0) - P(RQ_t = i \mid RQ_{t-1} = i) & i = j \\ P(RQ_t = i \mid RQ_{t-1} = 0) - P(RQ_t = i \mid RQ_{t-1} = j) & i \neq j \end{cases} \tag{2.4.22}$$

$$y_i = P(RQ_t = i \mid RQ_{t-1} = 0) \tag{2.4.23}$$

for i=1,2,3,....,N. After solving for P(RQ=i), we may use it to remove the condition in (2.4.17) and obtain the required density function.

$$P(m = i) = \sum_{n=0}^{i-1} P(m = i \mid RQ = n)P(RQ = n) \tag{2.4.24}$$

The first and second moments of message service time for the non-initiator of a busy period may thus be determined from

$$E[m] = \sum_{i=1}^{N} iP(m = i) \tag{2.4.25}$$

$$E[m^2] = \sum_{i=1}^{N} i^2P(m = i) \tag{2.4.26}$$

Using the above results, the mean station delay is given by [Haye84]

$$\bar{d}_Q = \frac{E[\tilde{x}]}{1 - \lambda(E[m]-E[\tilde{x}])} + \frac{\lambda E[m^2]}{2(1-\lambda E[m])} + \frac{\lambda(E[\tilde{x}^2]-E[m^2])}{2(1-\lambda E[m]+\lambda E[\tilde{x}])} - 1 \quad (2.4.27)$$

Here we have subtracted off a single slot to obtain the mean queuing delay.

2.4.1 Mean Delay Performance Results

In this section, several samples will be given of the analytic model presented in the previous section. The model results will be compared with exact simulation results for a PBNet operating the REQPASS and SELKILL protocols. In the simulations, we assume Poisson arrivals to an infinite input queue at each station. The destination stations for the segment arrivals are assumed to be chosen uniformly over all stations.

In Figure 2.13, we start with a comparison of the protocols described in section 2.3. Note that the slot reuse protocols perform consistently better once the

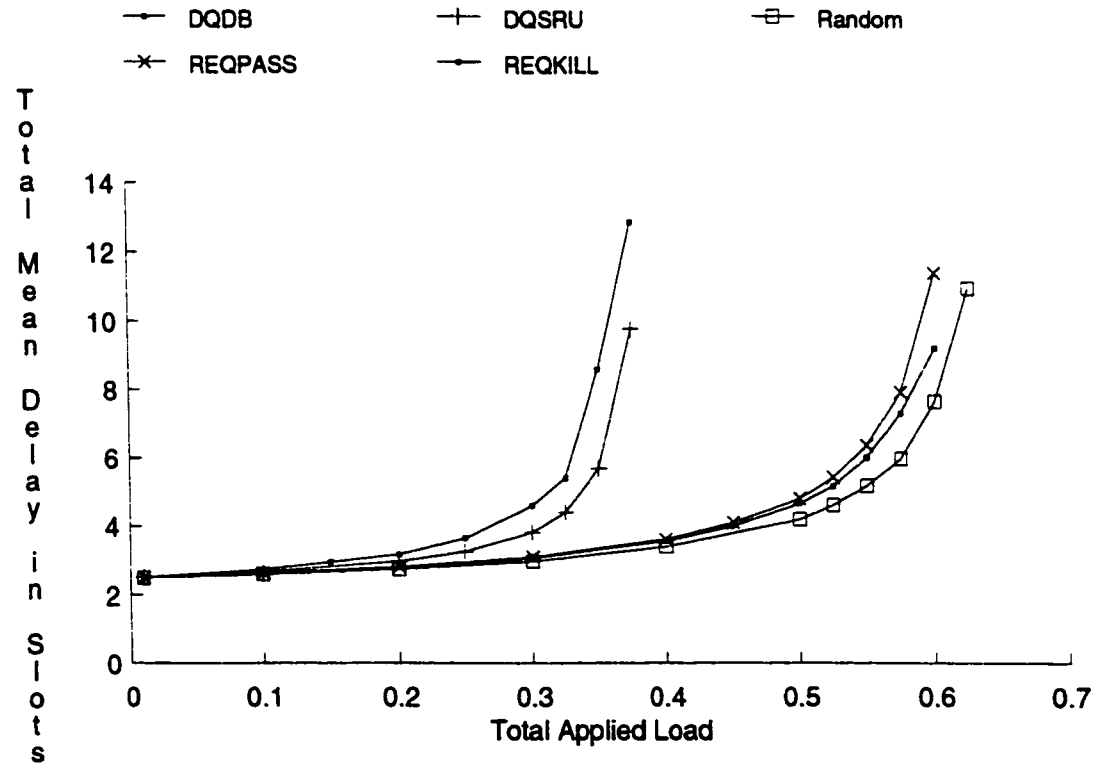


Figure 2.13 Comparison of Reuse Algorithms of 5-Node Network

issue of reservation bandwidth is addressed. Also note that the performance of the Random protocol is the best, although it is known to be extremely unfair.

In Figure 2.14, total mean delay results are shown for all nodes in a 5-node PBNet. Also given are the results obtained using the proposed analytic model. In this example, all nodes were loaded uniformly, with the appropriate arrival rates. It can be seen that the model predicts the station performance very closely. This is representative of the results seen for relatively small PBNets. Results taken for the individual nodes in the 5-node PBNet were equally good and are not presented here.

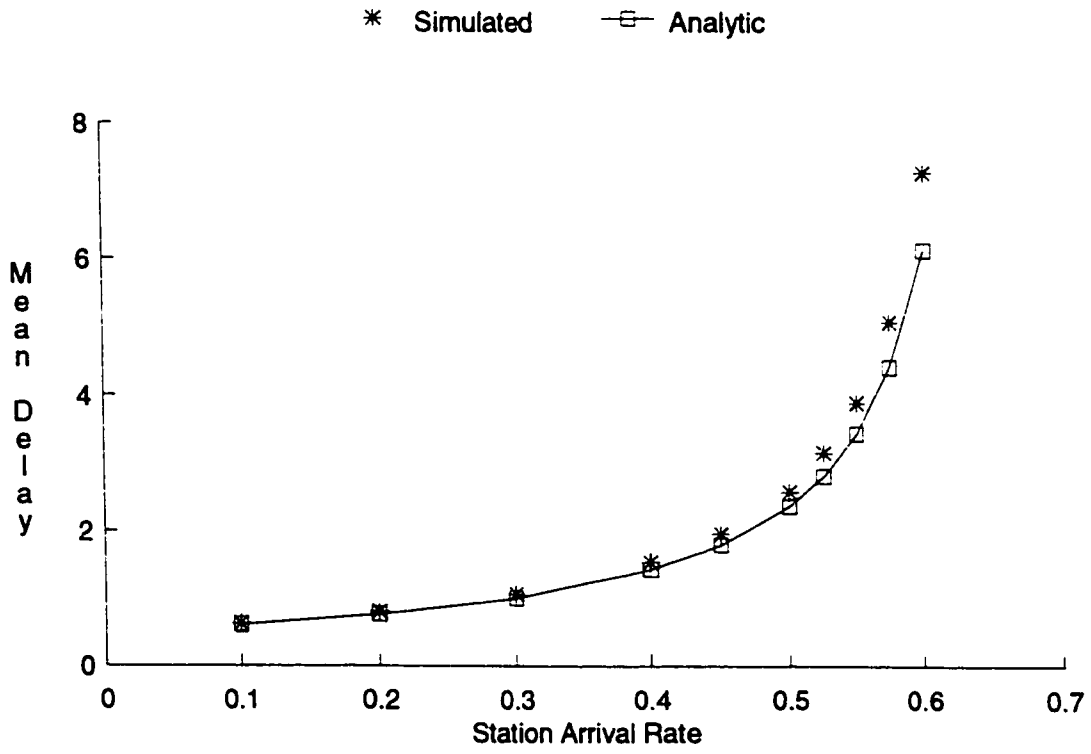


Figure 2.14 Total Mean Delay of 5-Node PBNet

Figure 2.15 shows further results for a 5-node PBNet. In this case, we include the mean delay performance of node 0 when the network is differentially loaded. A background load (BL) is held constant and uniformly distributed at all nodes except node 0. The curves show the mean delay, as the load at node 0 is varied in the

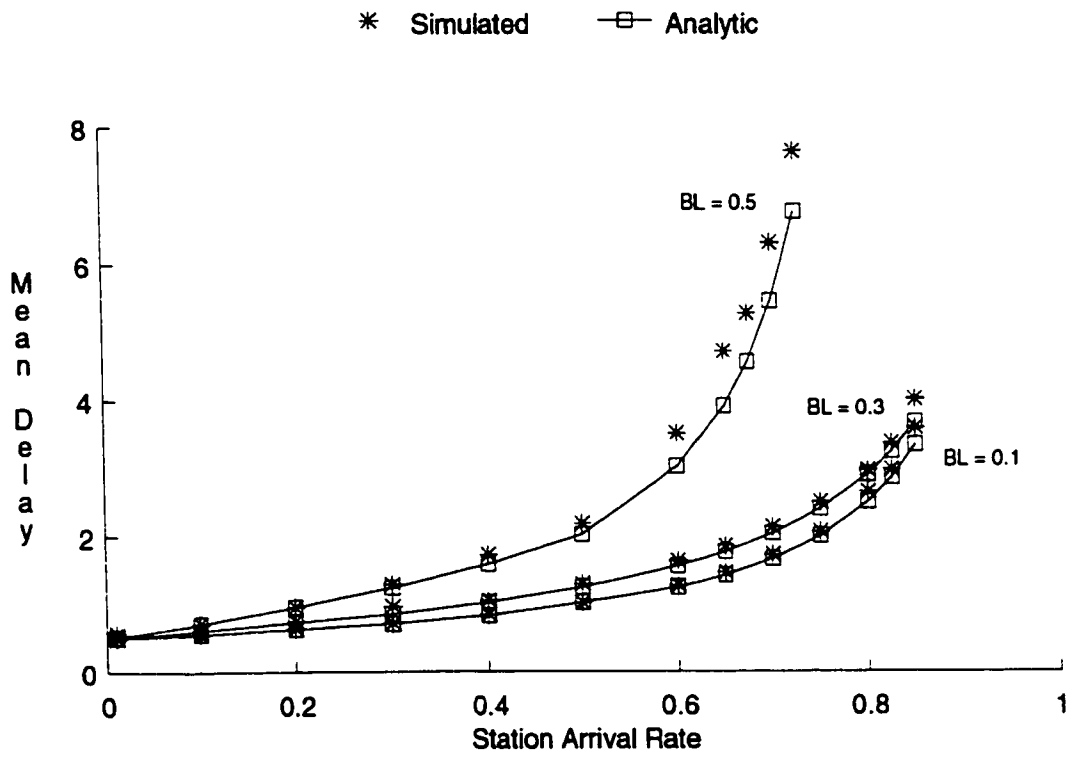


Figure 2.15 Mean Delay Node 0 (N=5)

presence of this constant background load. In all cases, the analytical model corresponds closely to the simulation results. The variation in the performance of the model as capacity is approached is due to the truncation of the distribution defining the message service time. As the actual delays become larger, the contribution of higher delays in the distribution becomes larger. In addition, as the background load increases, the queue at node 0 tends to see correlation effects in the request stream.

Since the model neglects these correlation effects, it is less accurate at values of high background load.

The results given in Figure 2.16 and Figure 2.17 are samples of those taken for an 11-node PBNet. Figure 2.16 gives the mean queuing delay for nodes 0 and 3.

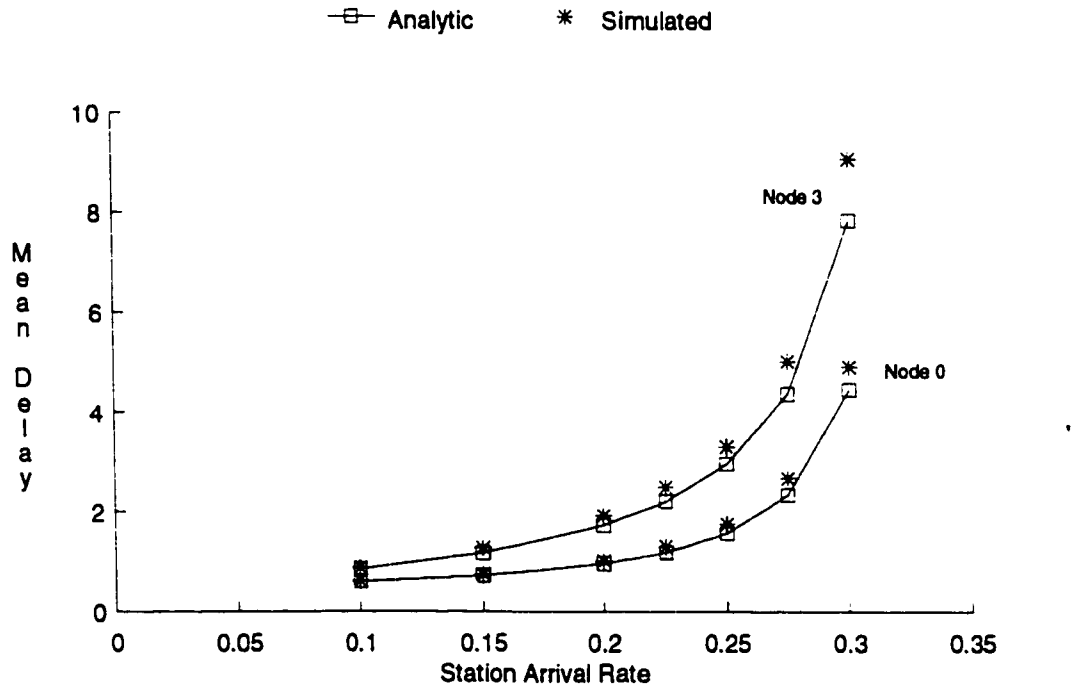


Figure 2.16 Mean Delay Nodes 0 and 3 (N=11)

Again, it can be seen that the analytic model gives mean delay predictions that are generally very good. Figure 2.17 shows the worst performance of the analytical model for all experiments performed. In this case, it can be seen that the model tends to overestimate the capacity available to node 5 as the network is loaded. Even though this is the case, the delay values obtained are still reasonable for design purposes, and are still within about 20% for delay values up to approximately 10. Slightly better

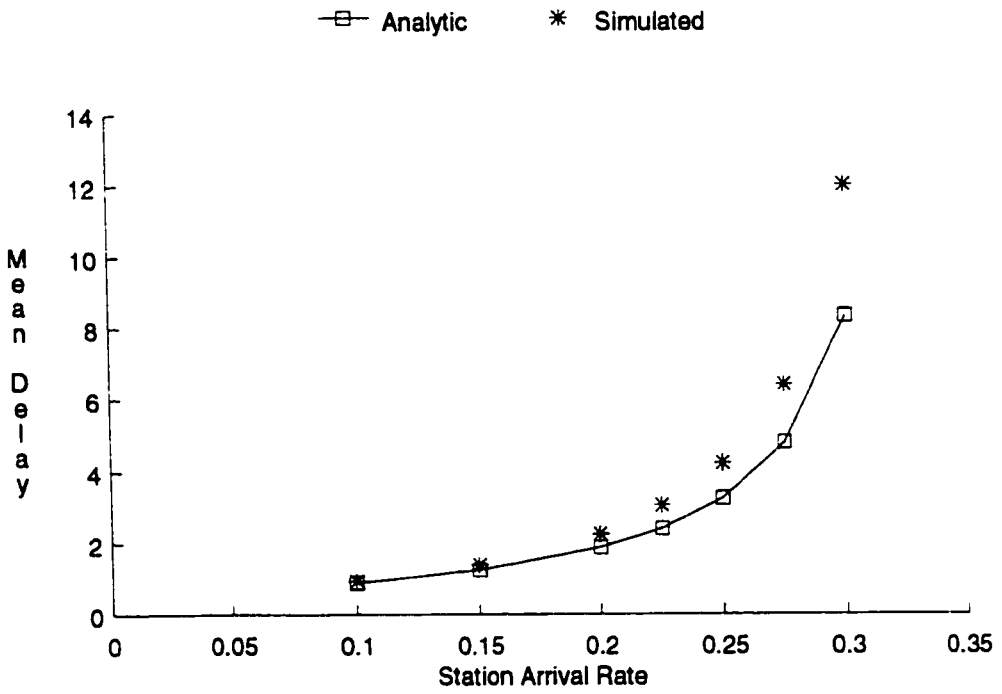


Figure 2.17 Mean Delay Node 5 (N=11)

results were obtained for node 6 in the 11-node network. Figure 2.18 shows the same system with curves for nodes 7, 8 and 9, including a curve showing the overall mean network queuing delay. In all cases, the analytic model gives very good results. Finally, in Figure 2.19 and Figure 2.20, results are shown for nodes 0 and 3 in a differentially loaded 11-node network. As before, the background load (BL) is held constant and the test node arrival rate is varied. Again, the analytic model does a very good job of predicting the simulation results. This is typical of that which was observed in other tests.

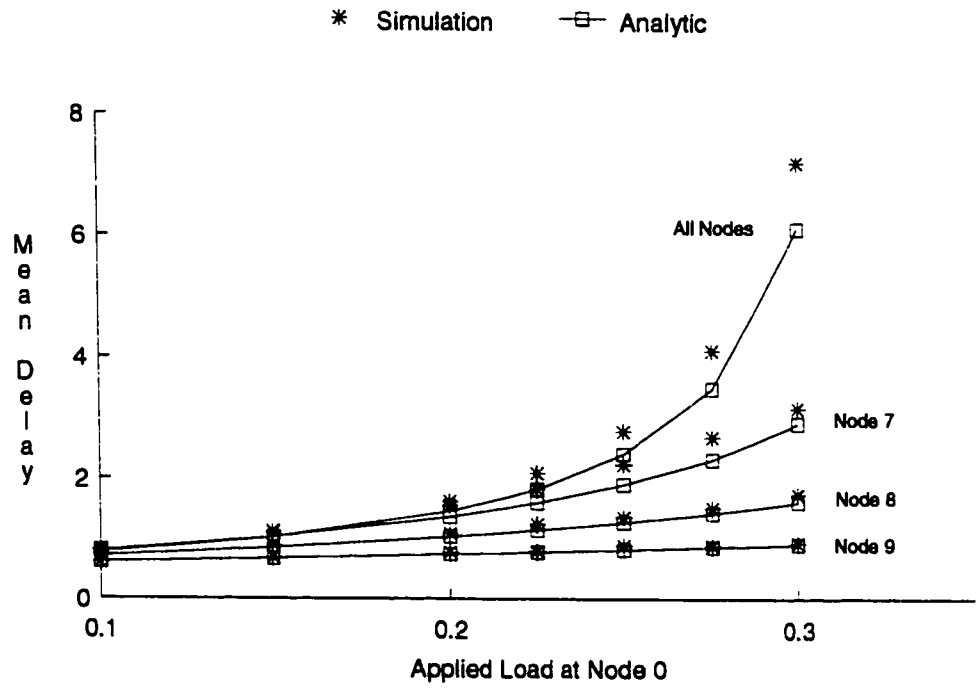


Figure 2.18 Mean Delay Nodes 7, 8 & 9 (N=11)

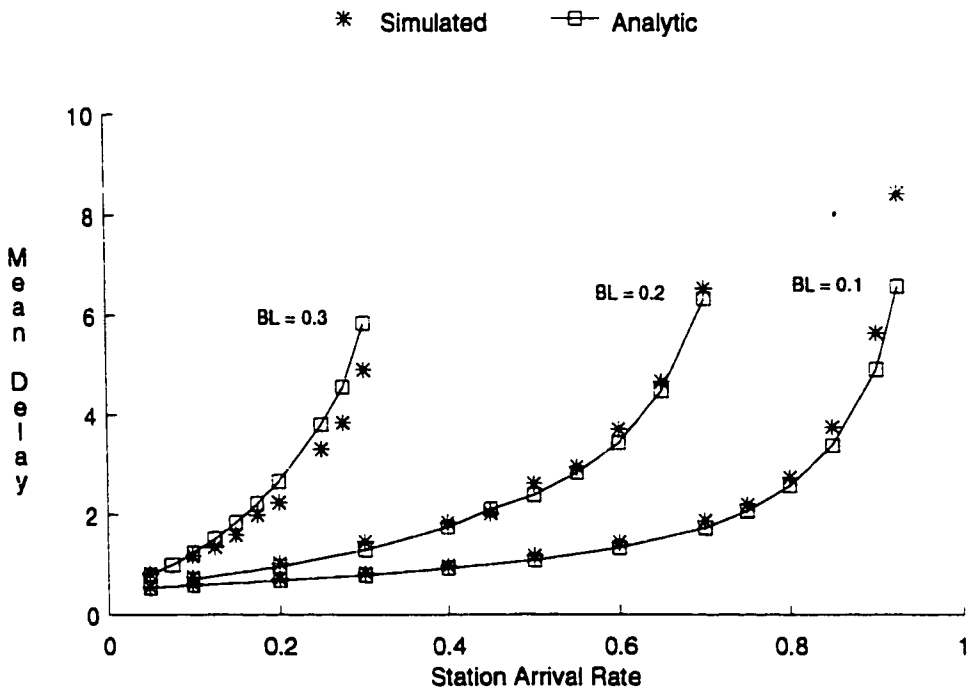


Figure 2.19 Mean Delay Node 0 (N=11)

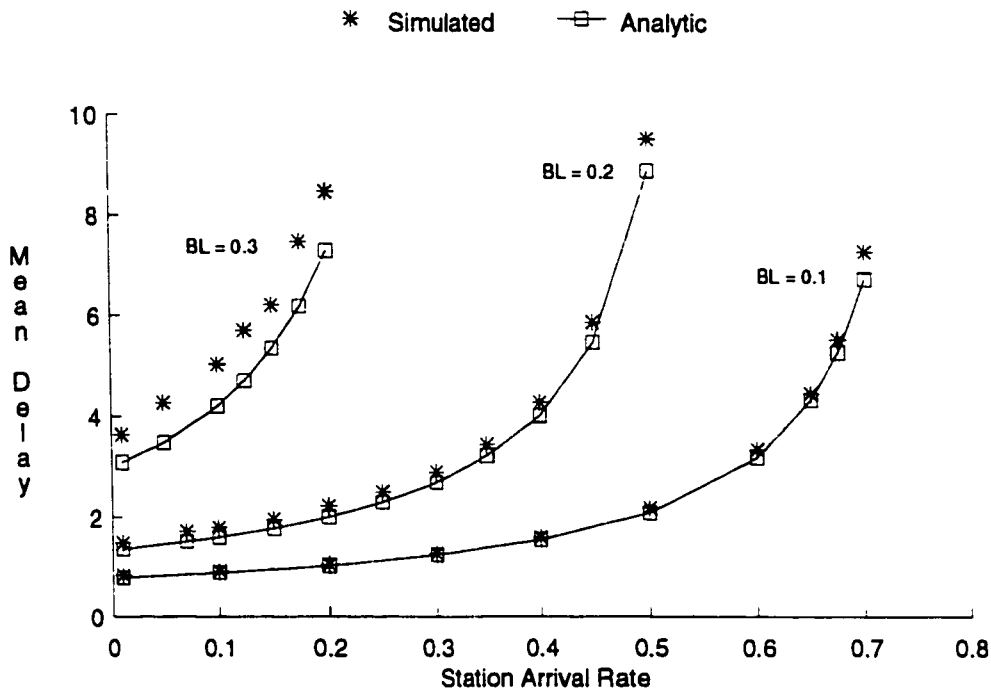


Figure 2.20 Mean Delay Node 3 (N=11)

2.5 Modifying the Delay Model for Receiver Allocation

The actual model developed in section 2.4 requires only simple modification in order to apply it to PBNets utilizing Receiver Allocation techniques. In an N station network, an NxN traffic flow matrix may be defined by the set of f_{ij} for i,j $\in$ {1,2,3 ... N}. Note that the stations are numbered from 1 to N, and each element f_{ij} corresponds to the average flow in full slots per slot period, sent from source station i to destination station j.

Two other NxN matrices are used to describe how receivers are allocated in these networks [Todd 94]. Each is a 'shadow' matrix for the forward (FS) and reverse (RS) buses. These matrices identify which stations act as shadow destinations for other stations. In these matrices, FS_{ij}=1 if, on the forward bus, station i is a shadow destination for station j. Otherwise, FS_{ij}=0. The diagonals of both RS and FS matrices are all 1's, since each station effectively acts as its own shadow destination.

The allocation of extra receivers modifies the traffic flow matrix. We call this modified matrix an effective traffic flow matrix. Consider the forward bus first. We modify the original flow matrix by adding to f_{ij} all traffic from station i that is destined for stations downstream of station j, for which station j is a shadow destination. We also subtract from f_{ij} all traffic destined for station j from station i that station j receives via its extra receiver. We therefore define a forward difference matrix (ΔF_{ij}) as

$$\Delta F_{ij} = \sum_{k=j+1}^{N} FS_{jk} \cdot f_{ik} - \sum_{k=i+1}^{j-1} f_{ij} \cdot FS_{kj} \qquad (2.5.1)$$

Similarly, for the reverse difference matrix (ΔR_{ij}), we get

$$\Delta R_{ij} = \sum_{k=1}^{j-1} RS_{jk} \cdot f_{ik} - \sum_{k=j+1}^{i-1} f_{ij} \cdot RS_{kj} \qquad (2.5.2)$$

The effective matrix entries, B_{ij}, are thus defined as

$$B_{ij} = f_{ij} + \Delta F_{ij} + \Delta R_{ij} \qquad (2.5.3)$$

This had modified the flow to account for the placement of additional receivers and allows us to directly apply the analysis of section 2.4 by setting d_{ik} in (2.4.1) equal to B_{ik} in (2.5.3). Otherwise, the analysis remains the same.

It has been shown [Todd 94] that this delay model produces good results when compared with exact simulations in many cases, including non-uniform traffic cases.

2.6 Summary

In this chapter, the design of slot-reuse protocols for photonic bus backbone networks (i.e. PBNets) were considered. Photonic bus networks are intended for use in an optical star configuration with multichannel fibre transmission. They are designed to have a low incremental channel cost, thus making them suitable for small, high bandwidth, multichannel backbone networks. The REQPASS Protocol was introduced to achieve fair allocation of bandwidth under overload conditions using the request mechanism introduced in [IEEE90]. It is simple, and achieves fairness in a wide variety of applications. The SELKILL protocol is more complicated, but provides fairness in all known cases. All cases were tested via an exact simulation of the networks. An analytic model was derived that permits an accurate calculation of individual station queuing delays by applying two simplifying assumptions. The model was shown to be accurate in a wide range of applications. A simple modification allows the model to be applied in situations where additional receivers are allocated. Once again, simulation verified the accuracy of the analytical model.

Slot reuse was introduced into the media access to allow for the investigation of several novel network design techniques based on the PBNet concepts. These techniques, such as Topological Design and Receiver Allocation, offered significant performance improvement over conventional networks (DQDB) in a wide range of test conditions.

A natural extension of the PBNet concept is to increase the degree of interconnectivity between the nodes. In fact, the proposed design technique of

Receiver Allocation is a first step in that direction. A generalization of PBNet concepts will be investigated in the next section.

3.0 PHOTONIC MESH NETWORKS (PMNets)

PBNets were designed as an extension of an emerging MAN network standard. A physical optical star network was used to create a dual bus 'virtual topology'. This, coupled with a multi-hop implementation, allowed for modification of standardized media access techniques, and created the opportunity to consider network design techniques that would not be possible in the original systems. The introduction of multi-hop implementation has been used in many different network design proposals [Hluc88, Acam87, Karo88, Eise88]. In this chapter, we will consider another way in which the combination of 'virtual topologies' and multi-hop implementations allow for the consideration of design techniques that are not possible in conventional networks. We begin by considering Receiver Allocation and PBNets in a different light.

As previously mentioned, the allocation of extra receivers in PBNet actually creates a lightly interconnected irregular mesh network. The media access technique applied in this case, however, limits the advantages of the mesh topology. Consider that when station j tunes an extra receiver to a channel (i-1, i), all data transmitted over this link can be received at station j at the same time as it is received at station i (stations are assumed to be on the forward bus). Station j checks each slot it receives via its extra receiver and absorbs data destined to itself. Any other destinations at this receiver are ignored. A station is thus only capable of using its extra receiver for slots destined to itself. In order to use the extra receiver so that other slots can take advantage of the mesh topology, the station would require store and forward buffering. Without the added complexity of such buffering, certain upstream stations are not able

to take advantage of shorter paths to destinations downstream of station j. To explore this situation further, consider Figure 2.5 (a copy is included below for convenience).

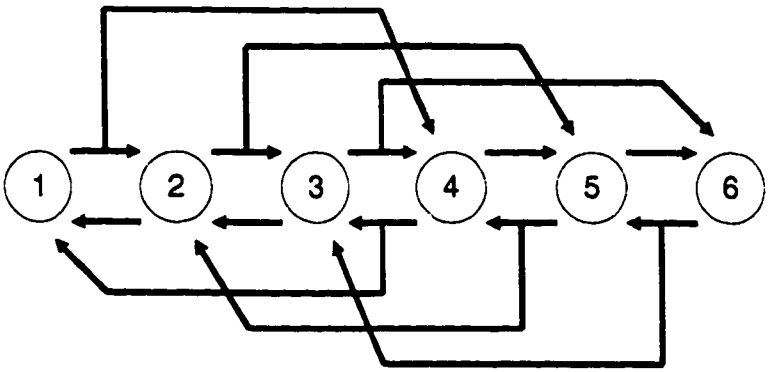


Figure 2.5 Receiver Allocation

In this case, a receiver is allocated to each node and tuned in a symmetric fashion. Note that this same network may be drawn as shown in Figure 3.0.1. The links drawn horizontally are 'dashed' in order to depict that they are not independent links, but are available due to the allocation of a receiver and the broadcast implementation of PBNet (there are 10 transmitters and 16 receivers in the network). It is interesting to note that a PBNet with allocated receivers may be viewed as an open chordal ring, where the flow on the chord is limited to last hop traffic. In this case, however, the added chords share the same transmitter at each source node. In addition, a chordal ring requires both routing and store-and-forward buffering, which implies a more complex node design. Receiver allocation can thus be thought of as a design

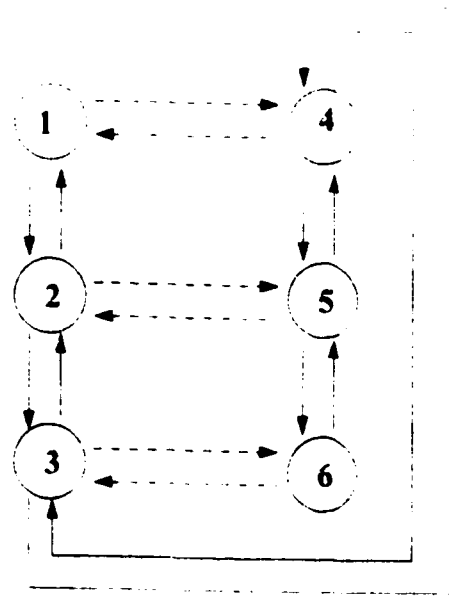


Figure 3.0.1 Receiver Allocation Redrawn

technique that simplifies node design by the careful location of extra receivers. The placement of these extra receivers actually predefines the routing decisions so that functionality is not required in the node itself. In fact, one can think of both the Receiver Allocation and Topological design techniques as methods of predetermining routing, in order to improve performance while keeping the node design simple. All of this is done within the context of a logical bus topology and its corresponding media access.

As an example, consider a slot sourced at node 1 destined to node 6. Define nodes 1, 2, 3 as column 1, and nodes 4, 5, 6 as column 2. Further, define nodes 1, 4 as row 1, etc. By using this approach, we have defined a mesh network of dimension 3x2. We started off with a bus topology. Through the allocation of one receiver per station, we have created a mesh network that uses fixed routing assignments. In

general, we can say that each slot follows the column until it reaches the row with the destination. It then follows the row to the destination. In the specific example cited, the slot would follow nodes 1, 2, 3 to arrive at node 6.

In general, any PBNet with N=mxn nodes can (using one receiver allocated to each node) form an mxn mesh network which, by design, will implement a column then row routing algorithm. The algorithm is not general, however. See Figure 3.0.2.

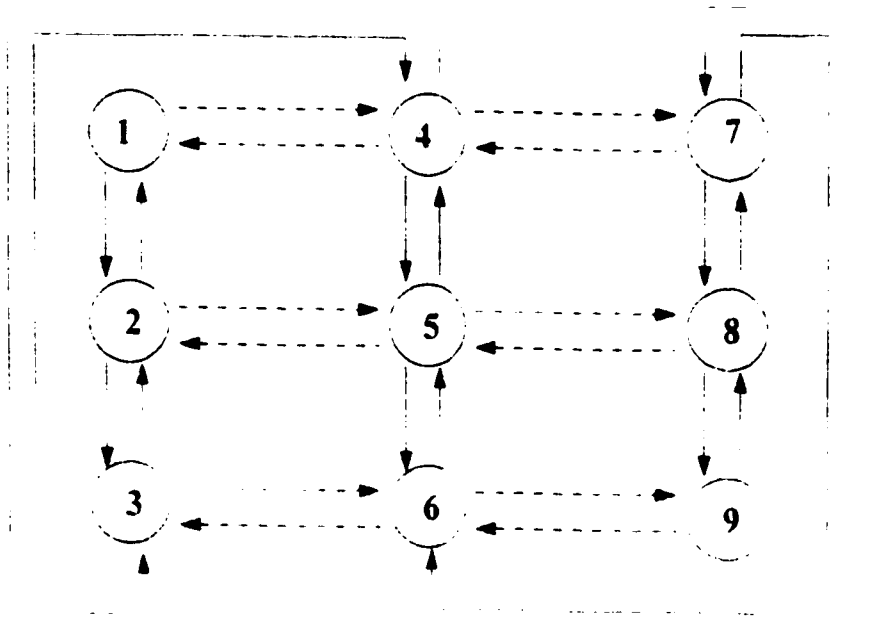


Figure 3.0.2 A 3x3 PBNet

In this 3x3 PBNet, consider node 1 as a source and node 9 as a destination. The slot would follow the column through nodes 1, 2 3 and then continue in the column direction through nodes 4, 5, 6 before following the row to node 9. We are now identifying some of the design trade-offs that occur when using PBNets with

Receiver Allocation techniques. On first inspection, the network of Figure 3.0.2 seems regular and heavily interconnected. There appears to actually be many possible shortest paths to follow between nodes 1 and 9. Due to our desire to reduce node complexity and take advantage of certain fibre-based technologies, we designed the network with a 'bus-based' media access on a broadcast medium with unequal numbers of transmitters and receivers. As a result, nodes do not need routing functions or store-and-forward buffers. The trade-off, however, is that slots do not necessarily follow the shortest path to their destinations, nor do they take advantage of the multiple paths to their destinations. These insights form the motivation for considering the design trade-offs and flexibilities in Photonic Mesh Networks (PMNets).

With PMNets, we have the same objective of simple node design. In particular, we wish to simplify the node design by taking advantage of abundant bandwidth on fibre links. Consideration of a mesh topology implies a routing function within the nodes in order to capitalize on the heavy interconnection inherent in the topology. In order to minimize the complexity of these routing functions, it was decided to use a regular mesh network. As an example, this would imply dropping the links between nodes 3, 4 and nodes 6, 7 in Figure 3.0.2. In addition, each of the row links should become independent by adding a transmitter for each allocated receiver. The result is a 3x3 regular mesh of nodes, interconnected by point-to-point fibre links. There does not need to be a transmissive star realization. Topologically, PMNets can be thought of as rows and columns of PBNets. In this case, however, we will redesign the media access and add limited routing functions.

PMNets were heavily motivated by Maxemchuk and his work on the Manhattan Street Network (MSN) [Maxe85, Maxe87a, Maxe89]. The system concepts that motivated the PMNet design are briefly introduced.

The majority of the research was targeted at a better understanding of a routing technique known as Deflection routing, and how it behaves in PMNet environments that are perfectly regular. The major application for this class of PMNet would be for small LAN backbones, where only non-isochronous traffic is considered. Since networks based on deflection routing are generally less well understood than many other networks, a significant effort in this research was to develop a meaningful framework within which to discuss such networks. This involved simulation of the network at a level of detail not published for other deflection routing proposals. In addition, the analytical modelling of PMNet was more general than other topologically homogeneous systems, such as the MSN.

3.1 Introduction and Architectural Overview

With PBNets, a system was proposed that:

(a) utilized a regular topology in order to simplify node design, especially the routing function,

(b) was positioned to take advantage of emerging fibre technology, particularly as regards bandwidth allocation,

(c) was a slotted system,

(d) utilized a relatively simple implicit media access technique.

In the end, this allowed for several novel design techniques to improve the performance over existing dual bus networks. We now wish to further improve performance by considering a two-dimensional form of PBNet called Photonic Mesh Networks. In this case, we will once again consider a slotted system, utilizing a regular topology with simple media access and the ability to flexibly allocate bandwidth. As previously discussed, this section of work was heavily motivated by Maxemchuk and his work on the Manhattan Street Network (MSN). The MSN was described in Section 1.1. It is reasonable to consider the MSN because it is a slotted, regular mesh topology network that uses simple media access. In addition, the MSN incorporates the novel deflection routing scheme. Some quick comparisons are warranted.

A number of important design concepts proposed by Maxemchuk are adopted in this work. For example, the topology in PMNet is regular, but the links consist of bi-directional lightwave trunks. It is intended that the design be capable of supporting

emerging (fine-grained) wavelength-division multiplexing (WDM) technology, which implies that the number of channels per trunk may not necessarily be the same for each node. This flexibility in design permits bandwidth allocation and network evolution (as was the case in PBNets), which is much easier to accomplish than in previous mesh network designs. In addition, a system may be engineered that consists of an arbitrary piecewise interconnection of regular topological segments. This feature also alleviates many of the problems associated with adding new nodes to the system, and relaxes some of the rigid topological constraints imposed by previous designs. In particular, the difficulty of mapping a logical MSN-type topology (where the wrap-around links must be treated as a unit hop) onto a real physical topology, with very high line speeds and metropolitan distances, is relieved. It does not appear that there are any nodal design penalties in relaxing the architecture in this way. As in [Maxe87a], the design of PMNet is tailored towards a significant reduction in the cost of the nodes themselves, while maintaining the advantages of a packet-switched network.

The architecture of PMNet is motivated by the requirement to mix a variety of different types of services in a single network supporting extremely high speed fibre links. Historically, both circuit and packet switching techniques have evolved that take advantage of decreasing processing costs to save expensive transmission costs through the deployment of processing power at switching points throughout the network. The deployment of this processing power, however, did not occur in an integrated environment. Circuit switching techniques are still used primarily for voice networks, and packet switching techniques are still used primarily for computer

communication networks. The technology to be used for the integrated environments of the future must perform well in the presence of traditional voice and data traffic, as well as a host of other traffic types that will be spawned by new services [IEEE87]. As pointed out in [Turn88], the flexibility of packet switching makes it an obvious candidate for networks designed for such diverse applications. The relatively poor speed, throughput and cost performance of the present packet switching systems, when compared with circuit switching systems, is not a result of the packet switching concept. Rather, it is the use of general purpose computers to implement packet switches that leads to the unfavourable comparison.

The fast-packet switching approach makes major strides in redesigning high-performance switching nodes to function using high speed fibre technology [Ades87, Abou88, Ahma88, Anid88, Arth88, Bern88, Cruz88, Eckb88, Eng88, Lee88a, Lee88b, Neum88, Neum88a]. The design of the entire network, however, tends to follow more classical techniques. Relatively large nodes (supporting hundreds or even thousands of lines) are strategically located in certain regions, with relatively distant loads being connected (possibly through concentrators) to these locations. In this way, large-scale residential and business services will eventually be accommodated using an ATM switching subnetwork.

In a Metropolitan Area Network, the approach tends to distribute the switching function closer to the load using nodes that are much simpler and more cost effective. This trend is a natural extension of the Local Area Network (LAN) technology (which tends to deploy bandwidth in a single dimension) initially deployed in the late 1970's.

In PMNet, for example, bandwidth is distributed in a topologically consistent fashion. The combination of this distributed switching, and the use of the connected bandwidth to reduce or eliminate buffer management in the nodes, may allow for the development of significantly less expensive nodes capable of supporting much higher line speeds. In view of emerging fibre optic technology that will permit the use of WDM, the traditional trade-off between switching costs and bandwidth may reverse. PMNet uses the availability of bandwidth (in the form of fibre optic links) to develop a network architecture that reduces the cost of the switching nodes. Such a system may be used to provide special, very high-speed data services in the metropolitan area and beyond. It should be noted that the use of routing methods under development in PMNet may be appropriate for certain traffic classes in a general purpose ATM switch.

A typical regular PMNet component is shown in Figure 3.1.1. It is seen to consist of the regular interconnection of switching nodes to create a square or rectangular grid. Topologically, the PMNet differs from the MSN in that it uses bi-directional links between nodes, and the edges of the network are not connected. As was the case with the MSN, the PMNet uses routing techniques (to be discussed later) that do not use buffers to store packets while they await access to their preferred output link. PMNet is a slotted system, where all packets are the same length. Incoming packets compete for the output links and, depending on the outcome, may actually take outgoing links that are not in a preferred direction. This is known as *deflection-routing*, and is a good example of the use of bandwidth to simplify the node design. Conceptually, the packets are being stored on the network (using the abundant

link bandwidth) rather than in buffers in the node. This also removes the need for flow control techniques to avoid buffer overflow in a node. Fortunately, the increased connectivity of the topology serves to reduce the number of deflections by offering a variety of different shortest paths to a given destination.

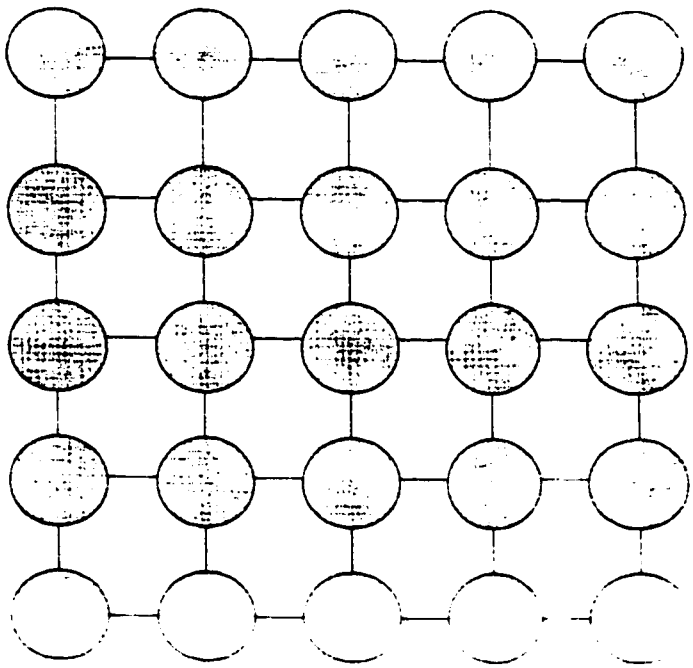


Figure 3.1.1 Regular PMNet Topology

The use of bi-directional links in a PMNet allows for the addition of incremental bandwidth on the network, without affecting the regularity of the structure. It is obvious from Figure 3.1.1 that the number of links into a node is the same as the number of links out of a node. Additional links may be added between any two nodes without affecting this input/output balance. Thus, as the network evolves, additional bandwidth may be allocated between any pair of nodes, provided that full-duplex fibre link pairs are added. It is easily seen that if this is the case, the number of incoming and outgoing links per node remains the same across the entire network. An example

of this capability is illustrated in Figure 3.1.2, where the additional lines between nodes indicates added full-duplex fibre links. It is easily seen that although the topology is unaffected, the number of input and output links is still equal for each node in the network. This provision permits considerable flexibility in tailoring the network bandwidth distribution to evolving user requirements.

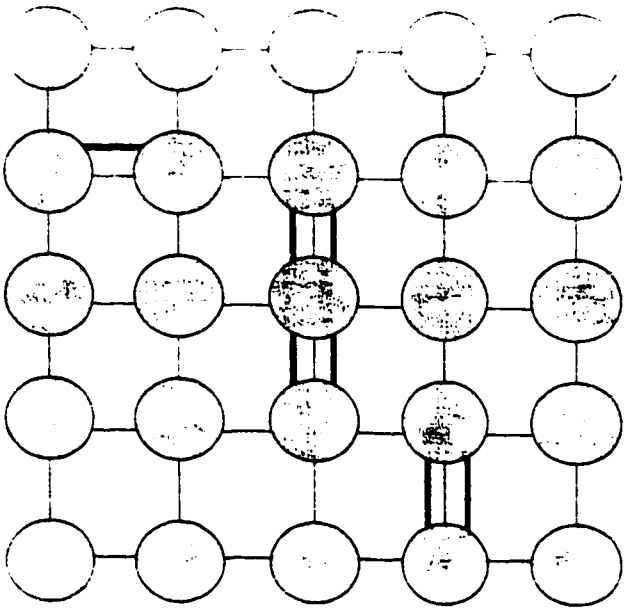


Figure 3.1.2 Bandwidth Allocation

We have already seen how the need for flow control to avoid node buffer overflow has been eliminated. The access schemes proposed for PMNet also tend to remove the need for flow control at the network access points. Packets may only enter the network when there is an empty slot on one of the outgoing links. Thus, as the load on the network increases, the number of empty slots available for entry traffic decreases. The fact that there is no buffering at the nodes leads to deflections which further restrict access to the network. This natural throttling effect removes the need

to control access to the network and ensures that the system throughput does not degrade as the system approaches capacity. The result is a further simplified network design.

The technique of deflecting packets offers another simplifying advantage. In most packet switched networks, a great deal of effort is expended to find a route for a packet which avoids congestion or problems on the network. Techniques involving dynamic routing complicate the design by requiring the availability of information which is not local to the node. If such a situation arises on PMNet, routing algorithms may be designed that inherently route packets around congested parts of the network. As the conditions on the network change, the routing changes to adapt to the new conditions.

A typical PMNet switching node is illustrated in Figure 3.1.3. It consists of an equal number of input and output fibre trunks. In this case, we have shown two links per trunk but, in general, there can be an arbitrary number, as long as the number of input and output trunks maintained are equal at a given node. Nodes at the edges of the network, for example, may only consist of two or three pairs of trunks. The trunks are organized into groups for each of the four directions needed to place the node in the PMNet topology. If a link on an input trunk fails, then a corresponding output trunk link must be disabled to maintain the balance at the node. The switching architecture of the node is thus designed to exploit the use of wavelength division multiplexing on the fibres [Okos87]. This offers the possibility of increasing aggregate trunk speeds to very high levels through the use of multi-link procedures,

and also allows for a convenient way to grow the network (in response to changing traffic patterns) using the existing physical trunks. Such techniques will be the subject of future research.

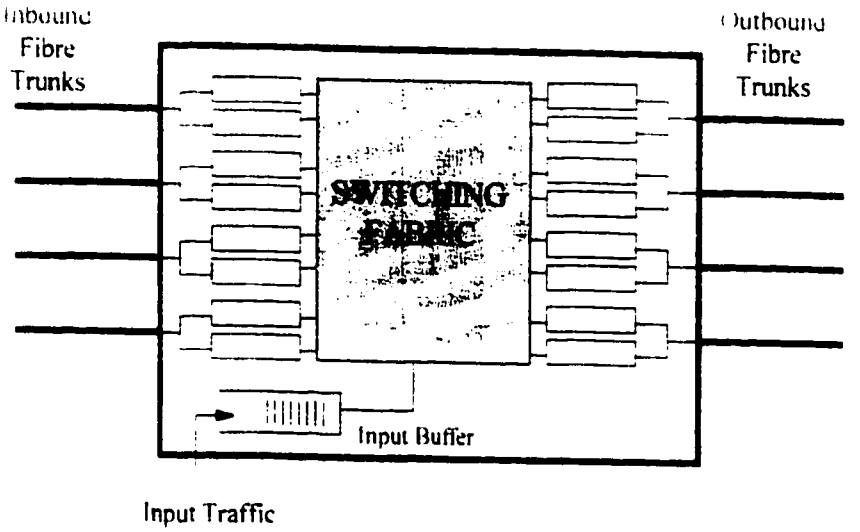


Figure 3.1.3 PMNet Switching Node

In Figure 3.1.3, assembly buffers are required on the input to receive incoming data and remove slot arrival skews. Once packets have been formed during the slot, they are presented to the switching fabric for assignment at the appropriate outgoing channel buffer. The actual switching fabric will be dependent upon the proposed routing. It is worth noting, however, that the small size of these distributed fabrics can make it possible, in certain circumstances, to design a completely non-blocking fabric. This would simplify the design by eliminating the buffers and associated flow control that are required in certain other fabrics (see [Turn88] for example). In the

figure, a single input buffer is shown for new entry traffic. In actual fact, depending upon the access strategy used, more buffers may be present.

The problems of non-uniform loads distributed over very wide areas can be addressed by a piecewise interconnection of the regular PMNet structures. An example is shown in Figure 3.1.4. In general, the routing within a regular subnetwork component is adaptive, as will be shown in subsequent sections. To accommodate topological irregularities, packets are routed to 'subdestinations', which provide an

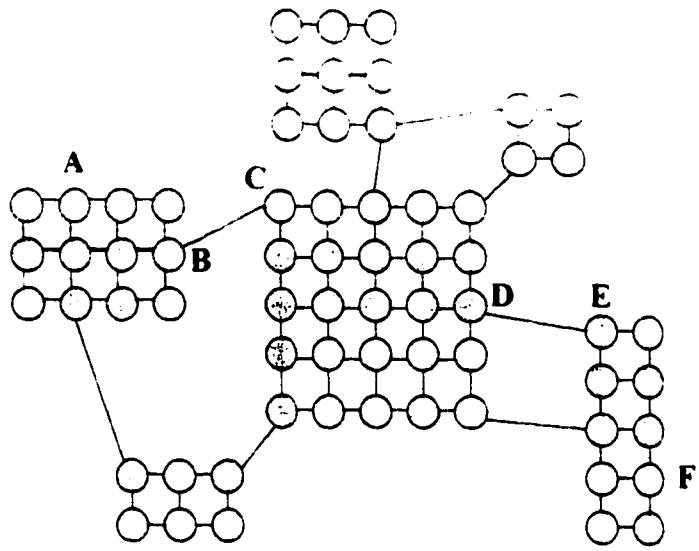


Figure 3.1.4 Piecewise Interconnection of Regular Structures

entry point to the next regular subnetwork. In Figure 3.1.4, a packet generated at node A, with node F as its destination, may be routed through subdestinations B, C, D and E. This may be accomplished using source routing techniques (where a portion of

the routing header is deleted at the end of each regular segment) or using a fast memory lookup at each node, based on globally unique addressing information. Thus, it can be seen that we have introduced a tradeoff between adaptivity in routing and the regularity of the network. This is true in that if the network is highly irregular, the routing tends to be less adaptive. Subdestination nodes include a simple boundary function, but otherwise perform in exactly the same way for normal transit traffic as do other nodes in the subnetwork. It is also worth noting that specific subnetworks and/or nodes could be implemented using different switching technologies. It is possible, for example, that specific nodes and/or segments be based on fast packet switches of the type introduced in [Turn88].

The PMNet node is simplified by the topologically regular distribution of the switching function and the use of high bandwidth links to remove buffering needs. The performance of such systems is very dependent upon the strategies for traffic processing at each node. Such strategies must relate intimately to the topology of the network to make maximum use of the system connectivity, and to appropriately administer the process of deflections. In the sections which follow, the basic routing functions of a PMNet node are presented and several algorithms are studied.

3.2 Traffic Processing in PMNet

In the previous section, an overview of the motivation behind PMNet was presented. In this section, we will investigate and present several possible routing algorithms. First, a framework within which different algorithms can be compared will be developed.

The specific PMNet we will be considering is a single, regular, square grid, consisting of nodes that have the same number of input and output links (see Figure 3.1.1). In the implementation of a PMNet, important attributes are the simplicity of the nodes and the ability to support extremely high speed lines. For these reasons, as well as others, the routing algorithm for a PMNet should satisfy several basic criteria, namely:

(a) It must be capable of supporting the high speed lines anticipated with the introduction of fibre optic technologies. In addition, it should be capable of conveniently supporting a multi-link procedure, in order to allow for the effective and incremental allocation of bandwidth on the network. This procedure is required in order to take advantage of emerging coherent lightwave techniques that may allow the spacing of many high speed channels on a single fibre. In addition, it may be possible to apply many of the design approaches considered with PBNets if flexible bandwidth allocation is supported.

(b) Due to the switching speeds required at the nodes, it is desirable to realize the algorithm in hardware using a ('wide-sense') non-blocking switch. The switching fabric itself may be implemented via a fast space division matrix as in [MacD88], or using a self-routing fabric as proposed in [Turn88]. Topological routing methods will be employed, thus permitting routing decisions within a single slot-time at lightwave speeds. The design of the fabric does not form part of this work. Since PMNet is a distributed architecture, it would be an advantage if all node hardware in the network was identical, regardless of the position of the node within the network. This attribute will be referred to as a distributed fabric, since it takes the concepts of a centralized, large switching fabric and distributes the fabric in a topologically regular way. The combination of the distributed fabric, the regular topology and the routing algorithm perform the function of the centralized fabric in a distributed fashion, over a wide area.

(c) In-keeping with the attempt to simplify the nodes, the requirement to queue transit traffic in the node should be limited, if not eliminated, as in [Maxe85, Maxe87a].

(d) One of the major advantages of distributing the switching function is the increased connectivity between loads. The routing algorithm should take into account this increased connectivity by selecting from the available routes to a destination in a rational manner.

The above criteria are not exhaustive. There are a host of other considerations in selecting a routing algorithm. Examples include ease of adding nodes, performance if a node fails and many others. Eventually, all such aspects must be considered, but for the purposes of this work the above, four points were used to focus the consideration of routing techniques.

Before describing the actual algorithms, there are strategies for traffic processing in PMNet that are common to all algorithms. These strategies will be defined in the next section.

3.2.1 Strategies for Traffic Processing in PMNet

Traffic processing in a PMNet is closely coupled to the topology of the network. From any location within the network, a source node can determine its location relative to a destination node, in terms of the number of row and column links that a message must traverse in order to reach that destination. In general, this information may be derived from, and/or used to determine, the header addressing data needed to route the packet through the intermediate nodes. A PMNet is a slotted system, so that once every slot period, a routing decision is made at every node. In the examples that we will be describing, this means that there can be, at most, n transit packets (one for each incoming link) that must be routed per node every slot. As in the MSN [Maxe87a], new entry traffic may only be considered when incoming trunk slots are empty, or if incoming packets are absorbed by a destination. Depending upon the access discipline followed, the access packets may enter the network before, or after, the routing decisions are made. Slots in a network are synchronized using a technique described in [Maxe88].

All traffic processing in a PMNet must be distributed in nature. This implies that the routing decision at each node will be made based on locally available information. The various routing methods under investigation belong to the class of topological routing algorithms, such that the decisions made are dependent only upon the current location of the packets relative to their destinations. In regular topological structures, a number of such algorithms were originally proposed in [Maxe87a]. In the examples we will be describing, the strategies of each algorithm are the same at each

node. In fact, there are three basic strategies to traffic processing in any of the routing algorithms of a PMNet.

The first strategy, referred to as the *access strategy*, relates to the manner in which new traffic is allowed to enter the network, and whether or not such entry occurs before or after the routing decisions are made for transit traffic. Two strategies are considered. The Post-Routing Access (P) method uses a single queue, which is formed for new traffic at each node in the network. New traffic is allowed to enter the network, and subsequently compete with the transit traffic in the routing decision, whenever any empty transit slots exist. Accordingly, when there are w_n packets in the input queue, and r_n available outgoing slots at the beginning of slot n, a total of $\min(w_n,r_n)$ packets are drawn from the input queue. Thus, if the number of incoming and outgoing links at the node is N_L, $N_L-r_n+\min(w_n,r_n)$ packets are considered for the routing decision in that slot interval. Figure 3.2.1 further illustrates the post-routing access algorithm. In a given slot, destination packets are first identified. This is followed by an empty slot-fill operation, which draws $\min(w_n,r_n)$ packets from the single input queue. Once this is accomplished, routing and allocation are performed, which consist of dynamically mapping the slots onto outgoing links.

By direct comparison, we can introduce the second Access strategy, Pre-Routing Access (R). With Pre-Routing Access (once again see Figure 3.2.1), destination packets are first identified, and then transit packets are routed and allocated. The empty slot-fill operation follows the routing and allocation. In this

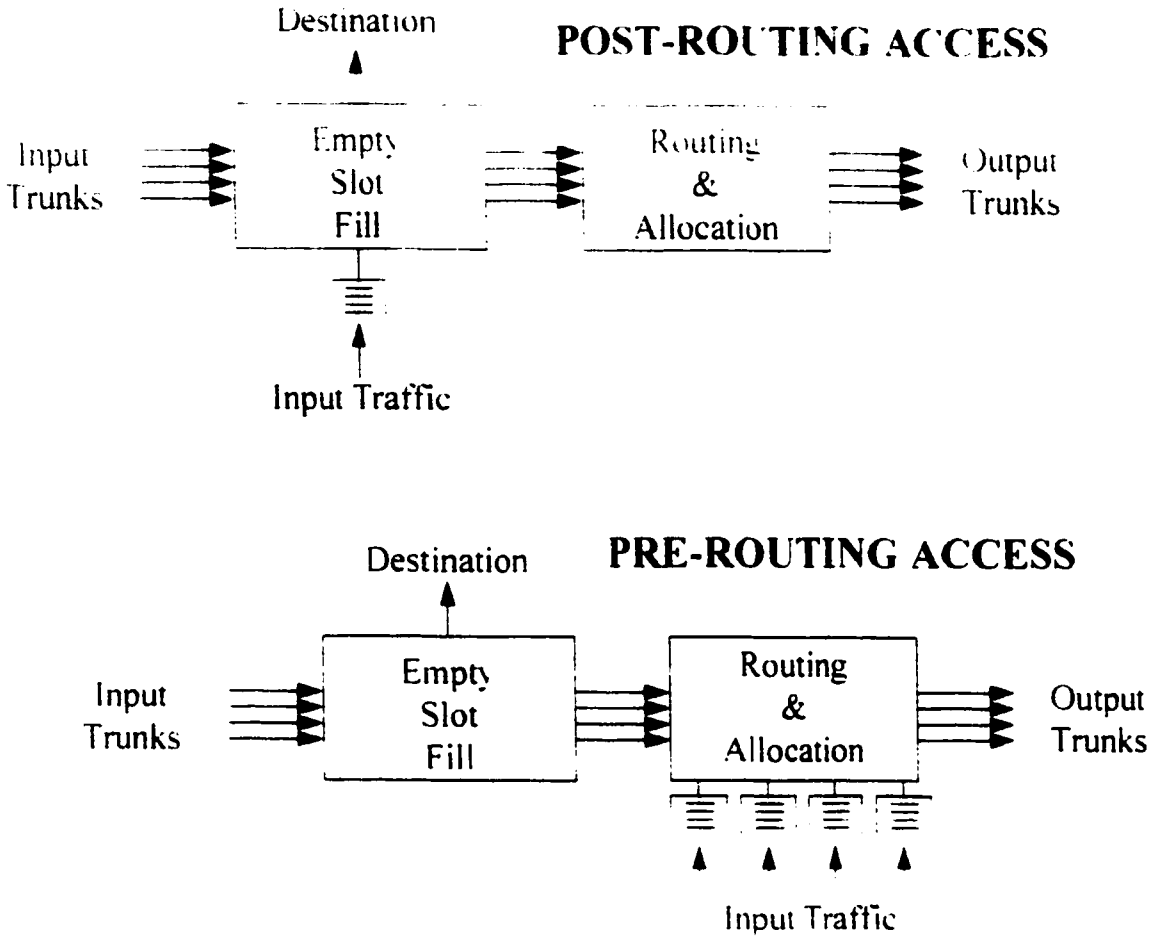


Figure 3.2.1 Pre and Post-Routing Access

case, note that a separate input queue is formed at the node for each trunk direction. A single new packet is allowed to enter the network from each input queue if there is an empty slot after the transit packets have been processed. Since the new packets select the trunk direction in which they will attempt to enter the network, they are said to be 'pre-routed', thus the term Pre-Routing Access.

We can see from the above description that Pre-Routing Access refers to the access strategy wherein access packets are assigned to specific output links in an *a priori* fashion. Thus, access packets are only permitted to enter the network after the routing decision for the transit packets has been completed. In this case, the entry packets do not compete with the transit packets and are allocated by some other means to the available output slots. Typically, this would be implemented by maintaining four separate queues, one for each output direction at each node. In such a case, new packets may only be allowed to enter the network if a slot on the output link corresponding to their queue is available. Hence, in Pre-Routing Access, the routing decision for entry traffic must be performed before the packets are assigned to a specific queue. This is in contrast to Post-Routing Access, where the transit and newly entering packets compete as peers in the routing decision. This fact is important, since the manner in which packets enter the network can be different between the two methods, and it will be shown that this can have a significant impact on the performance of the various algorithms.

The second strategy of traffic processing in a PMNet node is the *routing strategy* itself. This is the strategy that actually determines which packets will be assigned to specific output links. The routing strategy must be distributed, yet simple enough to allow for fast combinatorial implementation. Many schemes are possible; two generic types are considered here. The first algorithm tends to route packets along a (step-wise) diagonal from source to destination and will be called diagonal routing. The second technique routes in the row direction first, followed by the column direction, and will thus be referred to as orthogonal routing. It should be

noted that a variant of orthogonal routing was first suggested for use in HR4net [Borg87].

When routing is actually performed, a mapping must be made between the available packets and the output links in question. We refer to this process, which is the third strategy, as *link allocation*. For example, in the case of Pre-Routing Access, the allocation strategy may execute in two stages. The first stage would be applied to the transit packets, and the second stage would consider the entry traffic. The point to note is that the allocation strategy is used to resolve contention for the output links. In this work, we consider three allocations methods. The first two are sorting techniques that allocate the channels based on the results of a sorting procedure performed on topological information. The third allocation strategy is called a random strategy, since it resolves contention by randomly selecting the order in which to allocate channels.

Using this classification scheme allows for an abbreviated statement of an algorithm. For example, an algorithm employing Pre-Routing Access, Diagonal routing (to be described later) and Sorted allocation strategies will be called an RDS algorithm. Similarly, a POR algorithm implies Post-Routing access, Orthogonal routing and Random allocation. This terminology will be used in the following sections.

An integral part of each algorithm is the use of preference vectors. In the traffic processing algorithms investigated, preference vectors are used to define the

desired route for a given packet. At every node, a packet has the choice of any one of the four outgoing links. A preference vector states, in order of decreasing preference, the preferred links for the packet. The allocation strategy uses the preference vectors to determine the link allocation assignments. As will be seen, preference vectors are a convenient way of expressing the allocation strategy. In general, packets will be routed to the link of highest preference that is currently available. There will be more discussion of preference vectors in the sections describing each of the algorithms.

The above three strategies provide a framework in which to classify the various possible routing algorithms. Before discussing these algorithms, it is worth investigating some other properties of PMNets.

An important aspect of any algorithm is how well it is able to route traffic along one of the shortest paths between source and destination. Another important attribute of a routing algorithm is its ability to select the route that best matches the conditions existing on the network at the time of the routing decision. This second point will be the subject of future discussions. At this time, we will define a closed form expression for the average distance from any one node in the network to any other node. This will then be extended to find the average distance between source and destination on a PMNet. All this information will be used when comparing the performance of the algorithms.

Consider a regular PMNet subnetwork component, consisting of m rows numbered 0 to m-1 and n columns numbered 0 to n-1. In total, there are m*n nodes

in the network. To begin with, we focus our attention on the bottom left corner node, node (0,0). The sum of the distances from this node to all other nodes in column 0 is given as a straight-forward sum of an index; say i, as i varies from 0 to m-1. The distance from node (0,0) to all of the nodes in column 1 is given by the same summation, only each node is one link further away. At this point, we are assuming that each link has been normalized to one unit in length. By continuing this summing process for all columns, manipulating the results and dividing the global sum by m*n-1, we calculate the average shortest path (SP_{avg}) from node (0,0) to all other nodes in the network.

$$N_{00} = \frac{mn(m + n - 2)}{2} \tag{3.2.1}$$

$$SP_{avg} = \frac{N_{00}}{mn - 1} \tag{3.2.2}$$

If we now consider node (0,1), then we have moved 1 link closer to m(n-1) nodes and one link further from m nodes. Similarly, if we move to node (0,2), then we are 2 links closer to m(n-2) nodes and 2 links further from m nodes. Recognizing this fact allows us to modify N_{oo} for each of the source node locations. The result is that the average shortest path distance from any node (i,j) to all other nodes can be defined as follows

$$N_{ij} = N_{00} - in(m - i - 1) - jm(n - j - 1) \tag{3.2.3}$$

$$SP_{avg}(i,j) = \frac{N_{ij}}{mn - 1} \tag{3.2.4}$$

The global average shortest path is then found by summing $SP_{avg}(i,j)$ over all i (denoted as S_i) and all j (denoted as S_j) and dividing this sum by the total number of nodes in the network (m n).

$$SP_{avg} = \frac{mnN_{00} - S_iS_j[in(m - i - 1) + jm(n - j - 1)]}{nm(nm - 1)} \qquad (3.2.5)$$

Note that SPavg is not a function of the routing algorithm. It is a fundamental attribute of a PMNet, and will be used to judge the effectiveness of the routing techniques. We are now in a position to present the various algorithms.

3.2.2 Traffic Processing Algorithms

Each of the algorithms may be described using the classification scheme of Section 3.2.1. In order to gain an appreciation of the motivating factors underlying the design of the algorithms, two algorithms will be described in detail. These will be the RDS and ROR algorithms, and will involve pre-routing access, diagonal routing, orthogonal routing, one form of sorted allocation and random allocation. By combining each of the access, routing and allocation strategies, the various algorithms are created. In total, there are two access strategies, two routing strategies and three allocation strategies, providing a total of 12 algorithms. After the detailed RDS and ROR descriptions, the remaining access and allocation strategies will be presented. Each of the algorithms uses the concepts of orientation and preference vectors, which will be discussed shortly.

RDS - Pre-Routing Access/Diagonal Routing/Sorted Allocation

The technique used to remove the need for transit buffers in the PMNet is to deflect packets on the network. Deflections occur when, at a given node, a packet is not allowed to use a link that will move it one step closer to its destination. This occurs when the desired outgoing links have been allocated to other packets, and the affected packet is said to be deflected onto the network in a direction that takes if further away from its destination. The RDS algorithm was designed to allow transit traffic to perform exceptionally well in this type of environment. The access, routing and allocation strategies were devised to reduce the number of deflections. This behaviour was obtained by:

(a) Restricting the access strategy so that packets can only enter the network in their most preferred direction (i.e. pre-routing access).

(b) Selecting a routing strategy that tends to maximize the number of deflection-free alternatives for a packet in future steps towards its destination (i.e. diagonal routing).

(c) Applying an allocation strategy that assigns the outgoing links in a manner that attempts to maximize the number of future deflection-free alternatives (i.e. secondary counter sorting).

As has already been pointed out, the algorithm must be closely coupled with the PMNet topology. In Figure 3.2.3, we define the framework within which we can explain the algorithm. In general, a node has four input and four output links. Each link is in one of the four directions stated. Nodes at the edge of the network will have either two or three links in the appropriate directions. If we consider (i,j) to be the source node, then the destination may lie in any one of four quadrants defined by the links emanating from node (i,j). The packet that leaves the source node on route to the destination is said to have an orientation depending on the quadrant in which the destination lies. Provided that the packet is not deflected beyond the destination or into another quadrant, the orientation will be the same at all intermediate nodes between source and destination. This implies that the preferred outgoing link or links will be known immediately from the orientation.

The shortest distance between source and destination is given by the number of rows plus the number of columns separating them. In our case, we want to follow the shortest path in such a way that we make maximum use of the connected bandwidth to circumvent points of congestion or output link contention. In order to do this, we define a column counter (N_c) and a row counter (N_r). Note that this information may not be explicitly carried in the packet headers themselves, but may be derived from them. Hence, we may view the column counter as being decremented each time a link is traversed in the column direction towards the packets destination. An analogous procedure applies to the row counter. When both counters are zero, the destination has been reached. Deflections, on the other hand, will increment the appropriate counter. The next step is to utilize these counters to obtain the desired algorithm.

For a given packet, the distance between a transit node and the destination is given by

$$N_t = N_c + N_r \tag{3.2.6}$$

The total number of shortest paths between this and the destination node is given by all of the possible ways of selecting Nr rows from Nt choices.

$$N_{sp} = \frac{N_t!}{N_r!\ N_c!} \tag{3.2.7}$$

It would be considered an advantage if the routing algorithm were to maximize N_{sp}, since this would allow the packet the greatest freedom to select future routes in the presence of congestion and still follow the shortest path to the destination. Inspection of Equation (3.2.7) reveals that N_{sp} is maximized by keeping the values of N_c and N_r

as close to equal as possible. This may easily be shown as follows. Consider the isolated decision as to whether to route a single packet on a row or column link. If a row link were used, then the number of shortest paths available at the next node would be given by $N_{sp} = N_t!/[N_c!(N_r - 1)!]$; and if a column link is used, the expression is given by $N_{sp} = N_t!/[(N_c - 1)!N_r!]$. It can be seen (by taking the log of the above two equations) that when $N_r > N_c$, routing along a row maximizes the number of shortest paths that the packet has at the subsequent node. This is the equivalent of selecting the outgoing link that will decrement the larger of N_c or N_r, and translates into following a diagonal route across the network towards the destination. The same principles are applied to deflections. If a deflection is necessary, the route that maximizes the next value of N_{sp} should be taken.

In general, we must consider that any packet arriving at a node (regardless of its incoming link) may want access to any outgoing link. Define the larger of N_c or N_r as the primary counter (N_p) and the smaller of the two as the secondary counter (N_s). In the event that contention occurs and it is necessary for a packet to select other than its most preferred route, priority will be given to the packets, in succession, based on the smallest value of secondary counter. This allocation strategy tends to give preference to packets that are closest to their destination, and thus removes them from the network. Note that if packets were assigned priority based on the lowest values of both N_s then N_p, this would be the equivalent of giving preference to packets closest to their destination. This technique will be described in a later section.

The algorithm, and in fact each of the algorithms in this work, is executed by using preference vectors. Preference vectors are tables that define the most preferred to the least preferred outgoing link for a given packet. It is via preference vectors that the routing and allocation strategies are merged. Depending upon the orientation and the relative values of the row and column counters, a given packet will select the outgoing link that is highest in its preference vector and which has not been previously allocated. Table 3-1 summarizes the preference vectors for the RDS algorithm.

Table 3-1 Preference Vectors For RDS

Orientation	$N_c = N_p$	$N_r = N_p$
0	0 1 3 2	1 0 2 3
1	2 1 3 0	1 2 0 3
2	2 3 1 0	3 2 0 1
3	0 3 1 2	3 0 2 1

This simple statement of preference vectors embodies the routing function. As an example, consider a packet with orientation 2 received at any node. Upon inspection of the row and column counters, it is determined which of the counters is the primary counter. In our example, assume that it is the row counter. From Table 3-1, we can see that this packet would prefer to use outgoing link 3 if it is available. If link 3 is not available, it would prefer to use links 2, then 0, then 1 in order of preference. The allocation strategy will determine when this packet will make its decision and thereby resolve any contention for the outgoing links.

A second example will now be considered, to allow for the development of expressions stating the distribution of the most preferred routes within a PMNet. Assume the source node is (0,0) and the destination node is (4,2) (see Figure 3.2.2). In this case, the original counters would be N_c=4 and N_r=2. From Figure 3.2.3, we can see that the packet orientation is 0. Since N_p=N_c, using Table 3-1, we determine the most preferred route to be direction 0. This same decision will prevail until we reach node (2,0), called the corner node. At this point, the two counters are equal, and either direction 1 or direction 0 may be taken. The corner node is so called since it defines one corner of a square whose opposite corner is the destination node. The most preferred route from a corner node to the destination node always follows a diagonal path, thus the name diagonal routing.

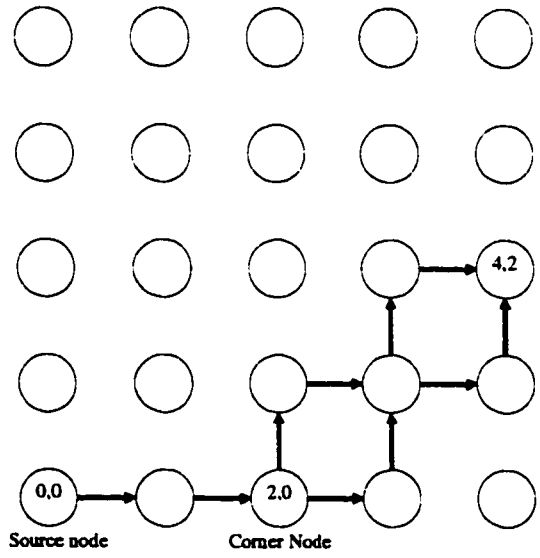


Figure 3.2.2 Diagonal Routing

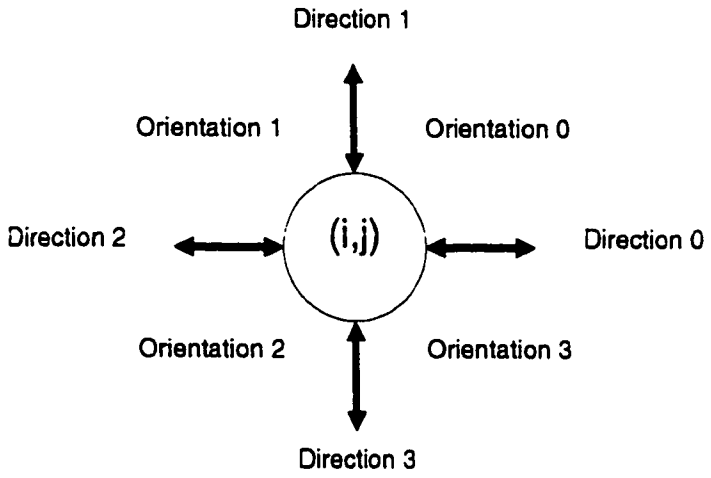


Figure 3.2.3 Node Reference Model

Given the above formulation, we will now investigate the distribution of most preferred routes using the RDS algorithm. Define the link of node (i,j) in direction k (k element of {0,1,2,3}) as link(i,j,k). Any most preferred route given consists of up to two segments. The first segment consists of links connecting the source node and the corner node. We will allocate any links used in this segment with a weight of 1.0 since there exists only one most preferred link at each step. The second segment consists of links joining the corner node and the destination node. In this case, the most preferred route consists of the upper or lower diagonal links or any combination of the two. Since either of two links can be selected at each step, these links will be allocated a weight of 0.5. By selecting all possible source/combination pairs, and summing the weightings for each link, we will obtain an expression for the relative frequency of use of each link when using the RDS algorithm. The analysis will only be valid when every packet has the luxury of selecting its most preferred route. Although in practice this is not reasonable, the results obtained give good insight into

the behaviour of the algorithm and will, in fact, form the basis for suggesting the PDS algorithm.

Definitions

Cd = destination column

Cs = source column

Ccn = corner node column

Rd = destination row

Rs = source row

Rcn = corner node row

By inspecting the algorithm described thus far, we can determine the total link weightings by considering each possible source/destination pair, and modifying the link weight each time according to the tests stated below.

if {|Cd-Cs| > |Rd-Rs|} and {(Cd-Cs) > 0}

then a node (i,j) will lie between the source node and the corner node if

{j = Rd} and {i < Ccn}

where Ccn = Cd-(Rd-Rs).

For all nodes (i,j) satisfying the above conditions, the weighting of link(i,j,0) should be incremented by 1.0. Define this as LinkWeight(i,j,0). Since the weight of link 0 is incremented, we say that this path has a search direction 0. From the corner node to the destination node, two sets of conditions apply to the weighting functions. All nodes which lie within the square containing the corner node and the destination node satisfy the following conditions.

{Rd > Rs} and {Rs ≤ j ≤ Rd} and {Cd-(Rd-Rs) ≤ i ≤ Cd}

The nodes that lie on the diagonal, or are one step away from the diagonal, will have links that lie along the most preferred path.

if {i-Ccn = j-Rcn} increment LinkWeight(i,j,0) and LinkWeight(i,j,1) by 0.5

if {1+i-Ccn = j-Rcn} increment LinkWeight(i,j,1) by 0.5

if {i-Ccn-1 = j-Rcn} increment Linkweight(i,j,0) by 0.5

The method described above takes into account all source/destination pairs where the original search direction from the source node to the corner node is direction 0 and the orientation of the packet is 0. Similar expressions exist for the other possible source/destination pairs. As an example, all weightings appropriate for search direction 1 are summarized below.

Search Direction 1 {|Rd-Rs| > |Cd-Cs|} and {Rd-Rs > 0}

if {[i=Cd] and [j<Rcn]} increment LinkWeight(i,j,1) by 1.0

if {[Cd>Cs] and [Cs<=i<=Cd] and [Rcn<=j<=Rd]}

if{i-Ccn=j-Rcn} increment LinkWeight(i,j,0) and LinkWeight(i,j,1) by 0.5

if{1+i-Ccn=j-Rcn} increment LinkWeight(i,j,1) by 0.5

if{i-Ccn-1=j-Rcn} increment LinkWeight(i,j,0) by 0.5

if {[Cd<Cs] and [Cd<=i<=Cs] and [Rcn<=j<Rd]}

if{i-Ccn=j-Rcn} increment LinkWeight(i,j,1) and LinkWeight(i,j,2) by 0.5

if{1+i-Ccn=j-Rcn} increment LinkWeight(i,j,2) by 0.5

if{i-Ccn-1=j-Rcn} increment LinkWeight(i,j,1) by 0.5

Similar expressions apply for all four search directions, but will not be included here.

The results of this analysis are shown in Figure 3.2.4, where the weightings for each of the nodes in a 5X5 PMNet are illustrated. Note that, as expected, the majority of the most preferred routes follow the diagonals of the network with nodes closest to the centre of the network having the highest weighting. Nodes at the corners of the network have the lowest weightings for two reasons. Firstly, they are at the extremes

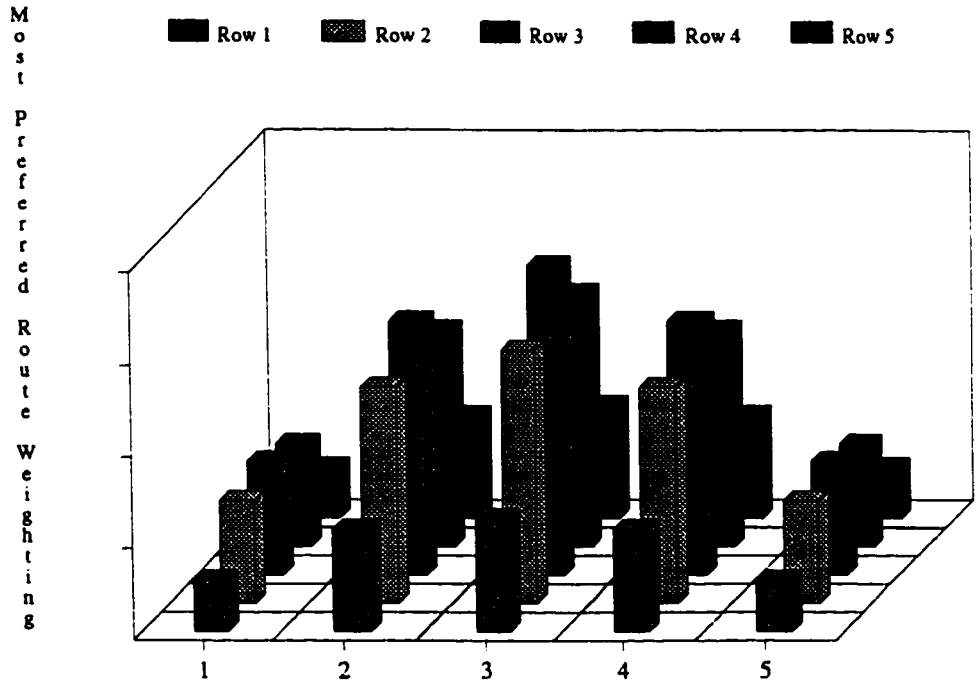


Figure 3.2.4 Preferred Routes - RDS

of the network and consequently are not as likely to handle transit traffic. Secondly, these nodes have the lowest number of outgoing links (there are just two) and, therefore, support a correspondingly lower number of most preferred routes. These

same two factors affect the weighting of the nodes on the edge of the network, but to a lesser degree. The centre node has the highest weighting and there is a great deal of symmetry across the entire network.

Other rectangular architectures [Maxe85, Maxe87a, Borg87] intentionally join the edges of the network with a link in order to provide topological homogeneity. With such networks, an analogous analysis would show even weighting across all nodes. This observation becomes apparent when one notes that in such networks any node can consider itself the relative centre of the network. In general, it would be considered an advantage to have even-weighting since it implies a fairness for all users once they access the network. In practice, however, such topological homogeneity may not be possible, or even desirable. Consider the following points.

(a) It is expected that with networks of these types, communities of interest will arise and that significant amounts of traffic will be confined to the community members. The advantage of direct links joining the edges of the network will therefore have a correspondingly smaller effect.

(b) As the number of nodes in a network grows, the effect of these single return links becomes more pronounced. This statement is based on the fact that it is possible to reach the other side of the network via one hop. In a realistic network, such a situation could not exist unless there were assumptions limiting the speed of the links. For example, if link speeds were 1 Gbps, slots on the system were 1000 bits long and a network were 100 km wide, the effective

length of the return link would be 330 slots long. This could be prohibitively long, compared to other links internal to the network and should, therefore, be taken into account in the routing decision. The routing decision to be made at each node would not be the same for all packets if this were the case, and the implementation would be more difficult. The value of the return links therefore become questionable, as the link speeds increase. In fact, such links introduce a discontinuity that ruins the homogeneity and make it more difficult to design a distributed routing algorithm. In a PMNet, a packet determines an orientation and, in some cases, sees a more homogeneous network from source to destination.

The problem posed by the results of Figure 3.2.4 is that of fairness among users. Users connected to nodes near the centre of the network may be denied access to the network because, on average, their node is forced to support more transit traffic from other nodes. This phenomenon will become obvious when the simulation results are presented. Possible ways of overcoming this problem are to allocate higher bandwidth links in the middle of the network, or to alter the network access strategy.

As a final point, it should be mentioned that Figure 3.2.4 applies only when the most preferred route is followed by every packet. In actual systems, other shortest paths will be taken in order to resolve contention, and this will have the effect of more uniformly spreading the transit traffic throughout the network without increasing the transit delay of a packet. A second-order effect will occur as packets are deflected.

ROR - Pre-Routing Access/Orthogonal Routing/Random Allocation

The ROR algorithm was considered, in light of the potential congestion problems of the RDS algorithm predicted in Figure 3.2.4. The algorithm is one of several that could be designed, to more uniformly distribute the most preferred routes across all nodes. Since the purpose was to study the effect of the routing strategy, the access method remained exactly the same as with RDS, except the preference vectors were altered to be consistent with the routing strategy. Unless the column counter is 0, a preference vector is always selected that has link 0 or 2 as the most preferred direction. The actual preference vectors are shown in Table 3-2. Note that in the table, the current column position is given by i, and C_d denotes the destination column for a particular packet.

Table 3-2 Preference Vectors For ROR

Orientation	$i \neq C_d$	$i = C_d$
0	0 1 3 2	1 0 2 3
1	2 1 3 0	1 2 0 3
2	2 3 1 0	3 2 0 1
3	0 3 1 2	3 0 2 1

The routing strategy selected follows one of the shortest paths defined in RDS as the most preferred path. The RDS algorithm was designed to perform well in the presence of heavier loads since the routing strategy employed tends to maximize the future number of deflection-free decisions for a specific packet. Since the ROR

algorithm follows only one of the possible preferred paths of the RDS algorithm, it is not expected to perform as well as the RDS algorithm for transit traffic.

The routing strategy is easily explained by comparing the routing of ROR with that of RDS. In RDS, packets follow a diagonal path from corner node to destination node. With orthogonal routing, as shown in Figure 3.2.5, there is no corner node. Packets attempt to follow along a row until they arrive at the same column as the destination node. At that point, they follow the column to the destination node. It is apparent that, in this case, fewer preferred paths pass through the centre of the network.

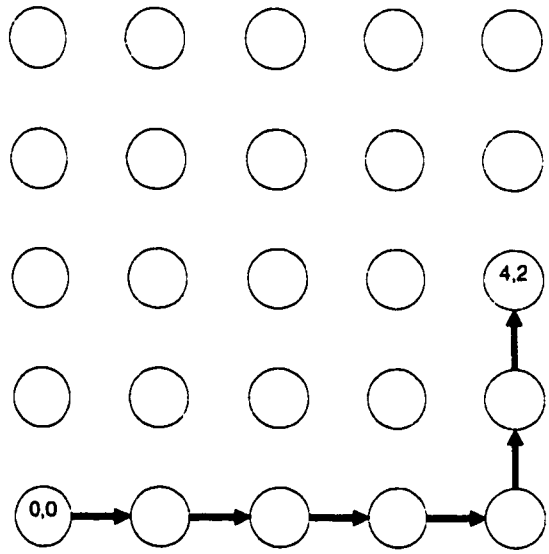


Figure 3.2.5 Orthogonal Routing

For purposes of comparison, we now derive the distribution of the most preferred routes with orthogonal routing using the same notation as was developed for diagonal routing. By inspection we can see that:

if{Cd>Cs}	increment LinkWeight(i,Rs,0) for i=Cs -> Cd-1
if{Cd<Cs}	increment LinkWeight(i,Rs,2) for i=Cd-1 -> Cs
if{Rd>Rs}	increment LinkWeight(Cd,j,1) for j=Rs -> Rd-1
if{Rd<Rs}	increment LinkWeight(Cd,j,3) for j=Rd-1 -> Rs

Figure 3.2.6 was developed using the above analysis, and can be directly compared with Figure 3.2.4. Note that the desired effect has been achieved. The most preferred routes are more uniformly distributed across all nodes. This is particularly true when one notes that the nodes at the edges of the network have fewer links than nodes in the middle of the network.

The allocation strategy in ROR is random. That is, packets which arrive at a node are awarded their desired outgoing link in a random order. Contention is resolved using the same preference vector technique as in RDS, except that preference vectors for rows are always used unless the destination and source nodes are in the same column. This is another factor that should lead to poorer performance than RDS as loads increase.

In summary, to design the ROR algorithm, a tradeoff was made. When compared with RDS, the ROR algorithm is expected to perform more poorly for transit traffic due to:

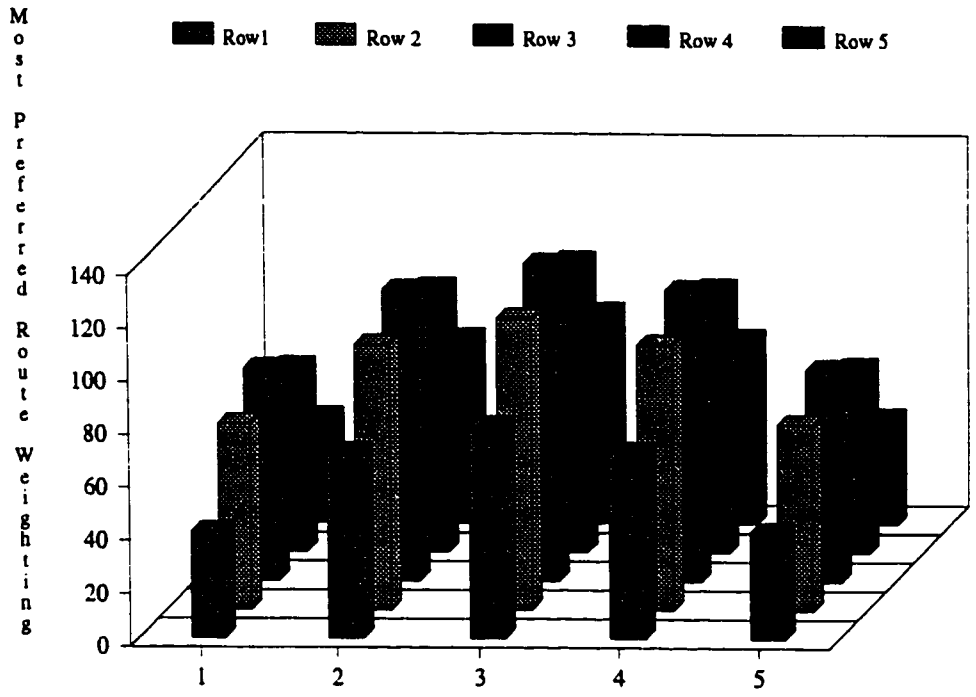


Figure 3.2.6 Preferred Routes - ROR

(a) A routing strategy that does not maximize future deflection-free decisions.

(b) Allocating outgoing links in random order, rather than using a sorting technique. This does not take advantage of the relative position of the packet to its destination.

(c) Using preference vectors that select rows as the most preferred links. This has the second-order affect of deflecting packets in a manner that does not maximize future deflection-free decisions.

The algorithm should, however, perform better than RDS for traffic attempting to enter the network since the distribution of most preferred routes is more uniform.

Post-Routing Access Strategies

In this section, the second access strategy is presented. The ROR algorithm was developed in an attempt to improve the access performance of the network by modifying the routing strategy. The Post-Routing Access algorithms will attempt to improve access performance by directly modifying the access strategy.

With pre-routing access algorithms, entry traffic to the network is placed into the queue associated with its most preferred outgoing link according to its orientation and the routing algorithm. In order for a packet to enter the network, an empty slot must exist on the outgoing link associated with its input queue. The concern is that this does not take full advantage of the bandwidth available to enter the network. A packet may be forced to wait in an input queue when there are empty slots available at that node. The post-routing access algorithms were developed to investigate the tradeoffs associated with allowing packets to enter the network in a non-preferred direction versus waiting to enter in the most preferred direction.

All of the post-routing access algorithms allow packets to enter the system whenever there is an empty slot received on any incoming links. The newly entered packets are then treated as if they were transit packets, and compete for the outgoing links according to the routing strategy. Rather than having an input queue for each outgoing link, the post-routing access algorithms select packets at a node from a single

queue using a first-come-first-serve (FCFS) discipline. Direct comparisons can be made between the pre and post-routing algorithms in order to study the affects of the access strategy on the total performance of the traffic processing function.

Allocation Strategies

In this section, algorithms are developed in order to investigate the sensitivity of the overall traffic processing to the particular allocation scheme used. Recall that the allocation scheme is used to resolve contention when several packets are competing for the same set of outgoing links. The resolution of contention might well imply that certain packets are deflected. It is the allocation scheme that attempts to administer the deflection process in an intelligent fashion, thereby attempting to improve overall performance.

Two allocation strategies have been described thus far. The first strategy sorted the competing packets according to the size of their secondary counters. By awarding the highest priority to the packet with the lowest secondary counter, this strategy attempted to avoid deflection at all costs. The packet with the lowest secondary counter was the closest to approaching a situation of imminent deflection. While this may not be true for all possible combinations of packets, it is certainly true that if the value of the secondary counter goes to zero and that packet is not allowed its first choice, deflection will occur. The secondary counter-sorting strategy assumes that it is globally better to reduce the future number of *don't care* decisions of all the other packets at the node than it is to deflect a single packet. Deflecting a packet obviously increases the delay incurred, which is viewed by this allocation strategy as a

last resort only. The quality of the decision, however, is limited to the local information available at the time of the decision.

The second allocation strategy resolves contention by randomly allocating the priority of selection. In this case, there is little value added by the decision since there is no attempt to use the local information to improve the quality of the decision. For this reason, the random allocation technique serves as a good baseline with which to compare other allocation techniques. Since it has already been described in the discussion of the ROR algorithm, it will not be discussed further at this time.

The third technique to be described in this section is similar in nature to the secondary counter-sorting technique. In this case, however, the parameter of concern will not be the secondary counter but the sum of both secondary and primary counters. It is clear that the sum of the two counters is the distance to the destination for the packet. The packet with the lowest value of this sum is given the highest priority in the selection process. This scheme attempts to remove the packets from the network that are closest to their destinations. The rationale for such a strategy is two-fold. First, the packets closest to their destination represent, on average, the largest investment of network resources and should, therefore, be given the highest priority. Second, packets that are close to their destination offer the best opportunity to free up network resources quickly. It can been seen that, when compared with the secondary counter-sorting technique, this distance-sorting technique places less direct emphasis on deflections in the network.

In summary, the twelve algorithms considered in this section, and simulated in the next section, are:

1) Pre-routing Access/Diagonal Routing/Secondary Counter Sort Allocation (RDS)
2) Pre-Routing Access/Diagonal Routing/Distance Allocation (RDD)
3) Pre-Routing Access/Diagonal Routing/Random Allocation (RDR)
4) Pre-Routing Access/Orthogonal Routing/Secondary Counter Sort Allocation (ROS)
5) Pre-Routing Access/Orthogonal Routing/Distance Allocation (ROD)
6) Pre-Routing Access/Orthogonal Routing/Random Allocation (RDR)
7) Post-Routing Access/Diagonal Routing/Secondary Counter Sort Allocation (PDS)
8) Post-Routing Access/Diagonal Routing/Distance Allocation (PDD)
9) Post-Routing Access/Diagonal Routing/Random Allocation (FDR)
10) Post-Routing Access/Orthogonal Routing/Secondary Counter Sort Allocation (POS)
11) Post-Routing Access/Orthogonal Routing/Distance Allocation (POD)
12) Post-Routing Access/Orthogonal Routing/Random Allocation (POR)

3.3 Simulation Results

As stated earlier, the behaviour of deflection routing networks is less understood than many other networks. For this reason, it was decided to perform an extensive number of simulation experiments for each of the twelve algorithms defined in the previous section. By studying and comparing the results, it was possible to gain greater insight into the operation of such networks. The total simulation work was too exhaustive to include here; however, in order that these insights not be lost, a summary of the simulation results (with some of the more interesting findings) is included in this section of the thesis. The insights are illustrated using results from 25-node networks, however, they are also valid for the regular networks from 9 to 81 nodes that were simulated.

The terminology used in this section will be defined as follows. The general manner in which traffic is handled at a node will be described as traffic processing. In order to facilitate a study of traffic processing, it has been broken down into three components. These components will be called strategies, leading to the definition of Access, Routing and Allocation strategies. The combination of particular strategies will be called algorithms as, for example, in the RDR algorithm.

In order to study the algorithms, an exact simulation of each algorithm was executed. The simulations included input queuing delays that, complete with transit delays, measured total delays for each packet generated. Most of the work in the literature prior to PMNet did not model the input queuing to the network, and consequently did not lead to the insights found in this work. Source and destination

nodes were uniformly distributed across the network, and a Poisson process was used to model single slot arrivals to the network. Each link in the system was assumed to be normalized to unit length (as in [Maxe87a]) so that the delay associated with moving from one node to an adjacent node was unity.

Three components of delay were measured for each simulation. Total average delay was calculated using all the packets from all the nodes in the network, and included the delay from packet creation until the packet exited the network. Average access delay was measured from packet creation until the packet accessed the network. The difference between average total delay and average access delay is the average transit delay. Average transit delay is a measure of the time that the packet actually spent on the network itself. From the equations developed in the preceding section, the average shortest path length for each node in the network was calculated. This is the transit delay which would be expected if all packets followed the shortest path to the destination. The difference between the average transit delay and the average shortest path is a component of delay attributed to deflection. All measurements were made on an individual-node basis and a complete-network basis.

Input loads to the network were varied from values near zero up to values where system capacity was approached. At each point on the curve, multiple runs were completed, with enough entry packets to produce results where 90% confidence intervals were less than 2% of the average value of delay. The exception to this occurred as the capacity of the system was approached, and confidence intervals

increased to 15-20% of the average values, but this was limited to the last few points on each curve.

The results will be presented in one, or both, of two forms. A set of curves showing each component of delay for the complete network versus the message arrival rate may be shown, or a three dimensional bar graph may be used to represent the average delay experienced by each individual node in the network at a specific arrival rate. Both forms of results will be related back to the original descriptions of the algorithms.

As was the case in Section 3.2, there will be a detailed discussion of the RDS and ROR algorithms in order to permit direct reference to the corresponding subsections of Section 3.2. This will be followed by results relating to each of the three strategies of the traffic processing function.

3.3.1 Results for the RDS Algorithm

Figure 3.3.1 shows the average total delay experienced by a packet versus the applied load. Note that as the load approaches zero, the delay approaches 3.83 slots. This is consistent with the analysis of Section 3.2, which would predict a transit delay of 3.33 slots when one of the shortest paths is always followed. In addition to the transit delay, one would expect that, on average, a packet would experience an access delay of 0.5 slots since packets are only allowed to enter the network at slot boundaries. This results in a total delay of 3.83 slots.

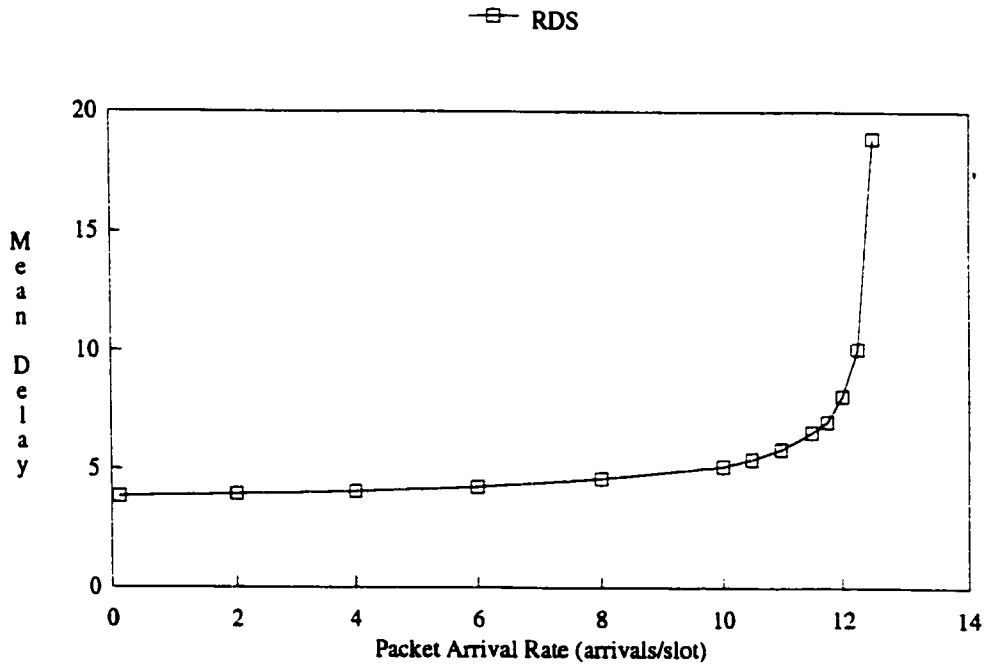


Figure 3.3.1 Mean Delay - RDS Algorithm

The delay increases slowly as the load is increased until the load reaches about 11 arrivals per slot. At this point, the delay increases quickly for very small increases in load, and would indicate that the network is approaching its capacity. In order to interpret this data, we should consider the following. A 5X5 PMNet contains 80 links. In fact, it can easily be shown that the total number of links (N_{links}) in an m*n PMNet is given by

$$N_{links} = 4mn - 2(m + n) \tag{3.3.1}$$

An ideal upper bound for network capacity can be derived by using equations (3.3.1) and (3.2.5), the equations for the average shortest path between nodes m and n. If there are 80 links on the network and the best average transit delay one can expect is 3.33 slots then, by applying Little's theorem [Haye84], the highest arrival rate one could support would be 24 arrivals per slot. Such a limit could never be reached in practice, of course, since average transit delays will always be greater than 3.33 slots due to deflection. In general, the maximum arrival rate (λ_{max}) that could be supported is given by

$$\lambda_{max} = \frac{N_{links}}{SP_{avg}} \tag{3.3.2}$$

Since the simulation actually shows the system capacity to be at arrival rates greater than 12.5 arrivals per slot, the RDS algorithm allows the system to operate up to approximately 52% of ideal capacity.

Further insight into the delay characteristic of Figure 3.3.1 can be gained by considering the individual components of delay shown in Figure 3.3.2. It is

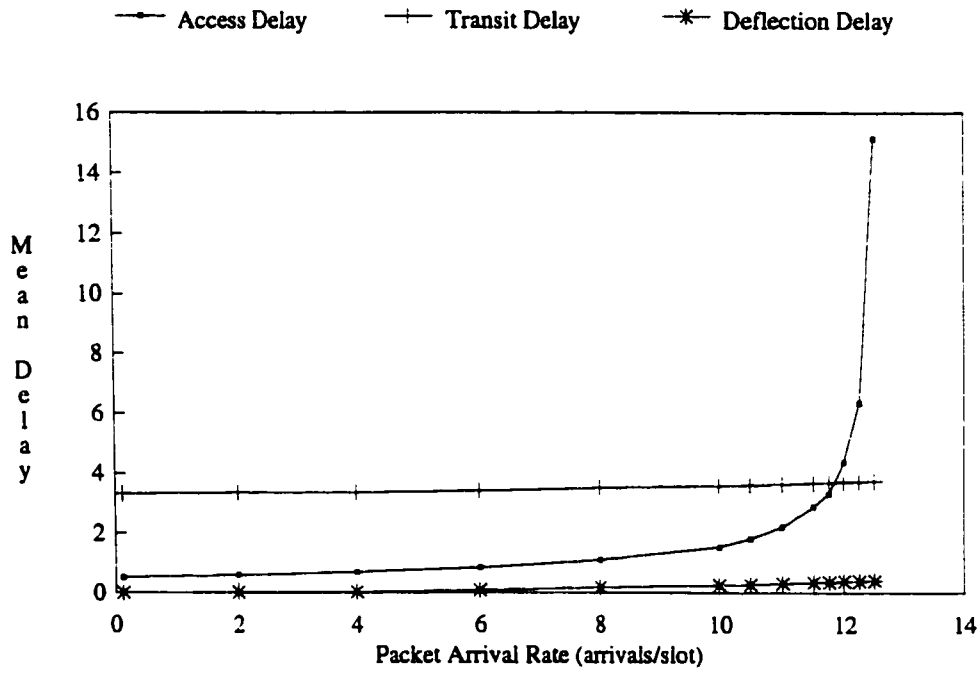
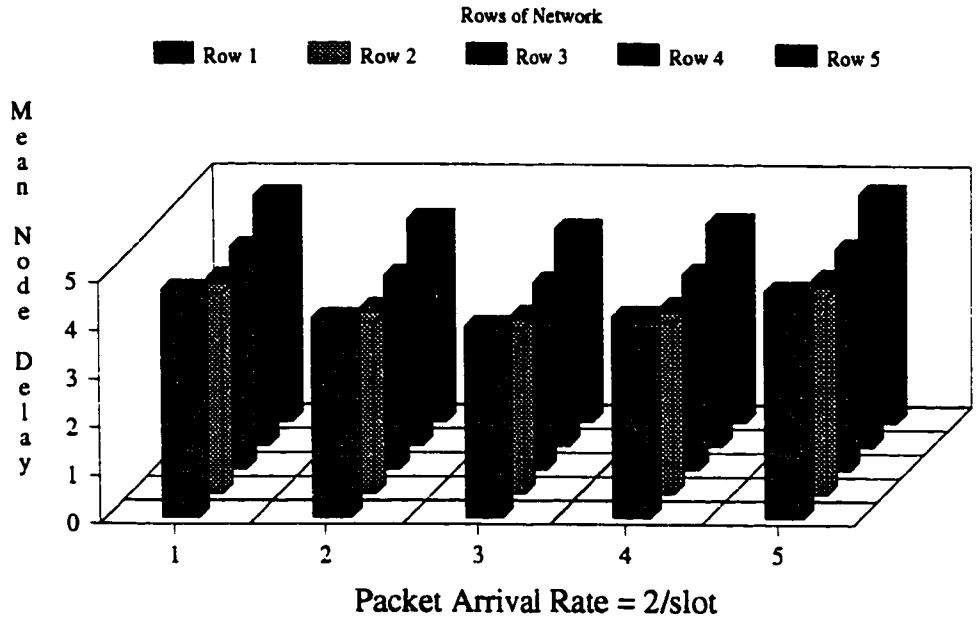


Figure 3.3.2 Delay Components - RDS

immediately obvious that the most critical component of total delay is access delay, and that as the system approaches capacity, it is the access delay that dominates. This is further support for the concern (expressed in Section 3.2) regarding access delay. Recall that the RDS algorithm was designed to perform well at higher loads. This performance is substantiated by the transit delays of Figure 3.3.2. Transit delay is always less than 3.8 slots, and it can be seen that the deflection delay appears to vary almost linearly with load. At all times, the delay due to deflection is a small portion of the total delay. The strong performance of the routing strategy is further support for techniques that address the access delay by placing a greater burden on the routing strategy.

Figure 3.3.3 is a three-dimensional representation of the delays experienced by traffic entering each node. In this figure, the message arrival rate is relatively low (2 messages/slot), and the delay at each node is seen to relate to the actual positioning of the node in the network. Nodes near the centre of the network are, on average,



Node Delays - RDS

Figure 3.3.3 Node Delays - RDS (2/slot)

closer to all nodes than nodes at the edges of the network. As expected, a great deal of symmetry exists, and the results are very consistent with those derived in Equation (3.2.4). As the load increases, access delay begins to play an increasing role, and the effect of the routing strategy becomes apparent. In Figure 3.2.4, we derived the most preferred route weighting function for each node. Comparison of this figure with

Figure 3.3.4 leads to some interesting conclusions. Figure 3.3.4 is identical to Figure 3.3.3 except that the load has been increased to 12 arrivals per slot. Note that, as might have been predicted from the most preferred routes, nodes at the centre of the network display much higher delays than others. This is due to the congestion caused in the centre of the network by the routing strategy that attempts to select diagonal paths. It is also obvious that, in this case, the capacity of the network is determined by the centre node, a problem which could be relieved if one could incrementally add more bandwidth to the centre of the network. Bandwidth allocation of this type is easily accomplished in a PMNet.

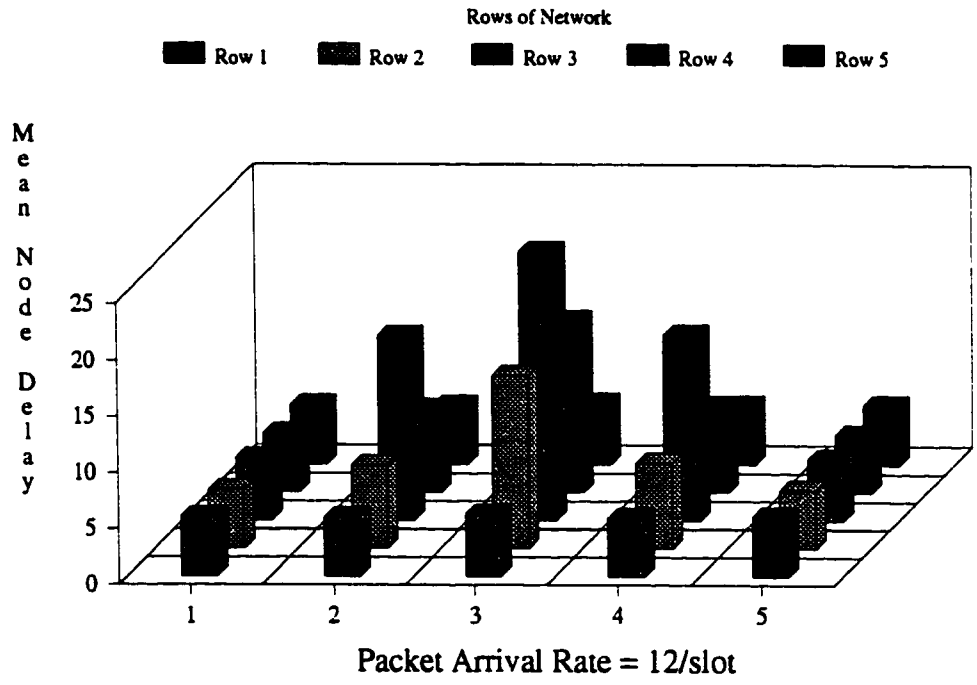


Figure 3.3.4 Node Delays - RDS (12/slot)

It is seen that access delay plays an important role in the performance of the network. It appears that we can identify three possible ways of addressing access delay in the design of the network. The first is to modify the routing strategy, and this will be investigated in the next section. The second technique is to modify the actual access strategy, and this will be investigated in our consideration of post-routing access techniques. The third technique involves incrementally adding bandwidth to the network in the appropriate places. This one is of particular interest since it relies on one of the fundamental properties of the PMNet. That is, that bandwidth may be added anywhere in the network, in the form of additional links, without affecting other parts of the network or the distributed routing algorithm. Other network architectures require that bandwidth be added uniformly across the network, or portions of the network, to address such a situation. This same capability would likely prove useful in addressing the requirements of non-uniform loads on the network.

3.3.2 Results for the ROR Algorithm

Figure 3.3.5 shows the various components of delay for the ROR algorithm. The main motivation behind this algorithm was to improve access delay by modifying the distribution of the most preferred paths selected by the routing strategy. The technique improved the performance so that the system remained stable for arrival rates up to about 12.85 arrivals per slot. This translates into approximately 54% of the ideal maximum capacity, and is marginally better than that of RDS. Once again, deflection delays were a small portion of the total delay.

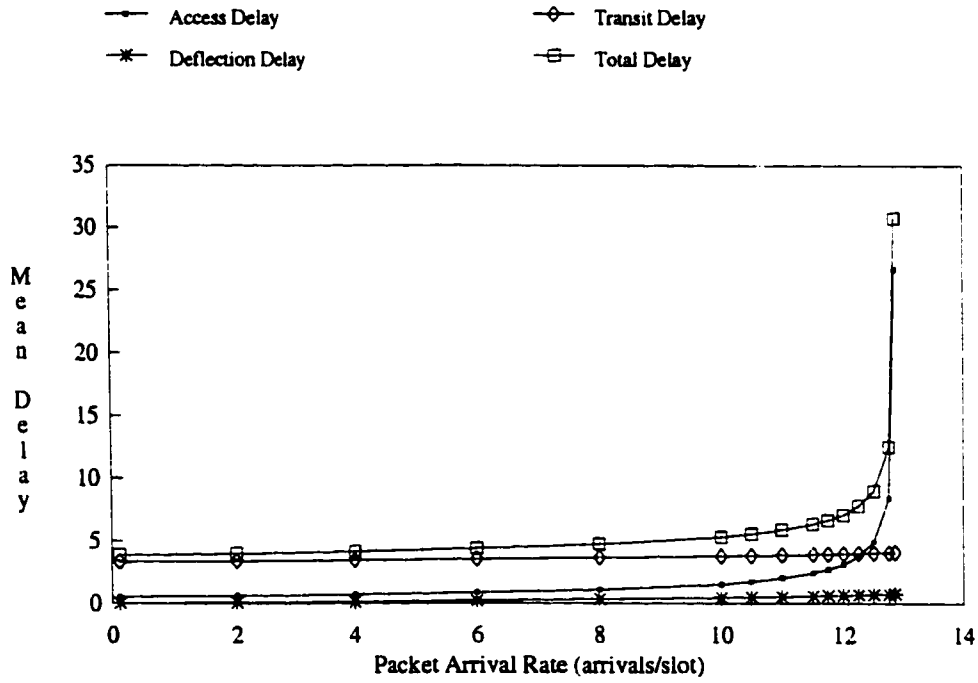


Figure 3.3.5 Delay Components - ROR

Figure 3.3.6 and Figure 3.3.7 can be directly compared with their related RDS figures. Such a comparison shows that both algorithms perform very similarly at low loads but differ substantially at higher loads. With ROR, the desired spreading of delays across all nodes has been achieved, and is directly attributable to a routing strategy that altered the resulting access delays at each node. As was the case with RDS, the opportunity appears to exist to improve performance by the appropriate allocation of additional bandwidth.

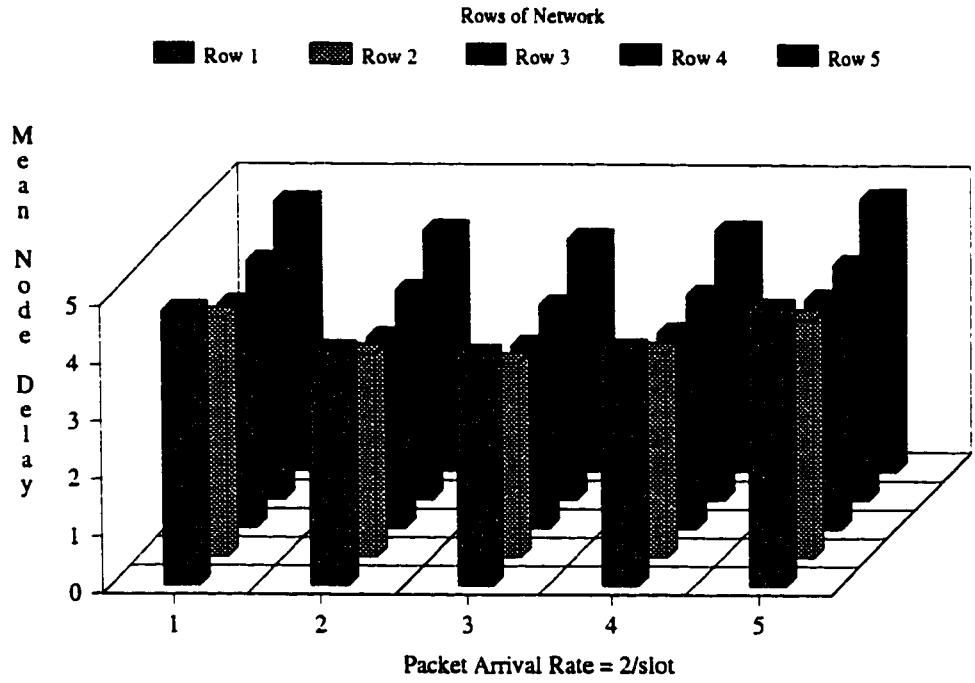


Figure 3.3.6 Node Delays - ROR (2/slot)

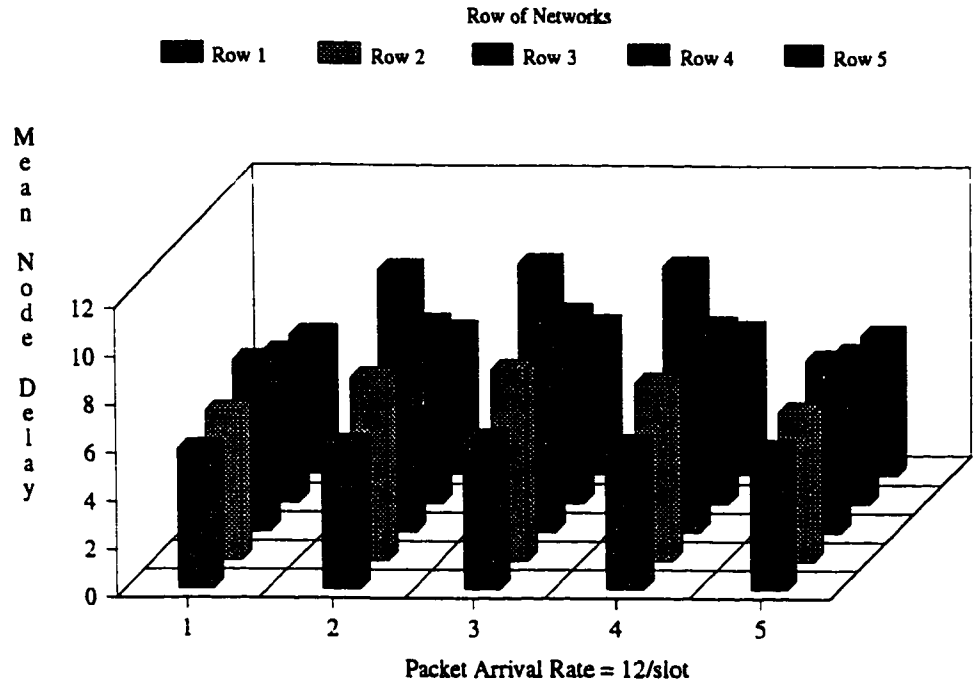


Figure 3.3.7 Node Delays - ROR (12/slot)

3.3.3 The Role of the Access Strategy

In the preceding sections, we have gained insights into the pre-routing access algorithm. The post-routing access technique was developed in order to study the effects of reducing access delay by potentially allowing packets to enter the network whenever any empty slot exists. The speculation was that this would more evenly distribute the delay across all nodes in the network, and thus improve performance. As an example, Figure 3.3.8, Figure 3.3.9 and Figure 3.3.10 apply to the POR algorithm. It can be seen in this case that, while the access delay is lower than with ROR, the net effect is to increase the average delay and to reduce the network capacity. The benefits of reducing the access delay are outweighed by the increased deflection delay. Also note that the deflection delay is not as linear as has been the case with the previous two algorithms. In Figure 3.3.10, it can be seen that the delays are nearly uniformly distributed across the network using POR. The behaviour of the algorithm can be explained as follows.

ROR uses a 'weaker' routing strategy than RDS, in an attempt to improve access delay. The use of post-routing access in conjunction with this routing strategy creates a situation where packets are allowed to enter the network in a non-optimal direction. The ROR routing algorithm is not able to cope with this situation without significantly penalizing transit traffic. The incremental improvements in access delay due to post-routing access are neutralized by the increased delays for transit traffic. The net effect is to provide poorer performance than ROR. This points out that uniform delays are not always consistent with the best performance.

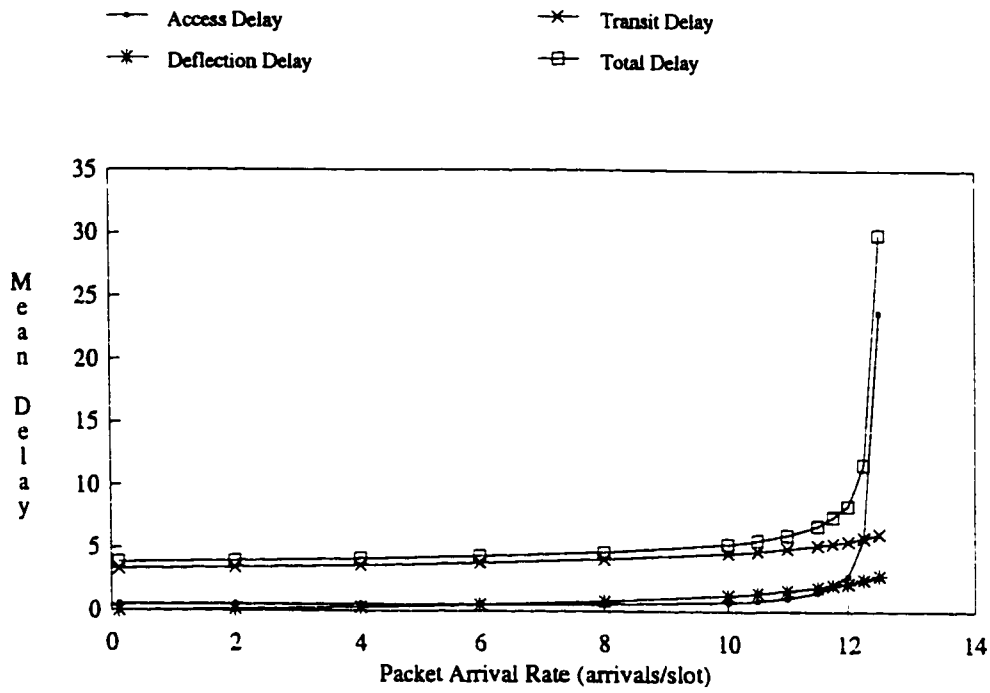


Figure 3.3.8 Delay Components - POR

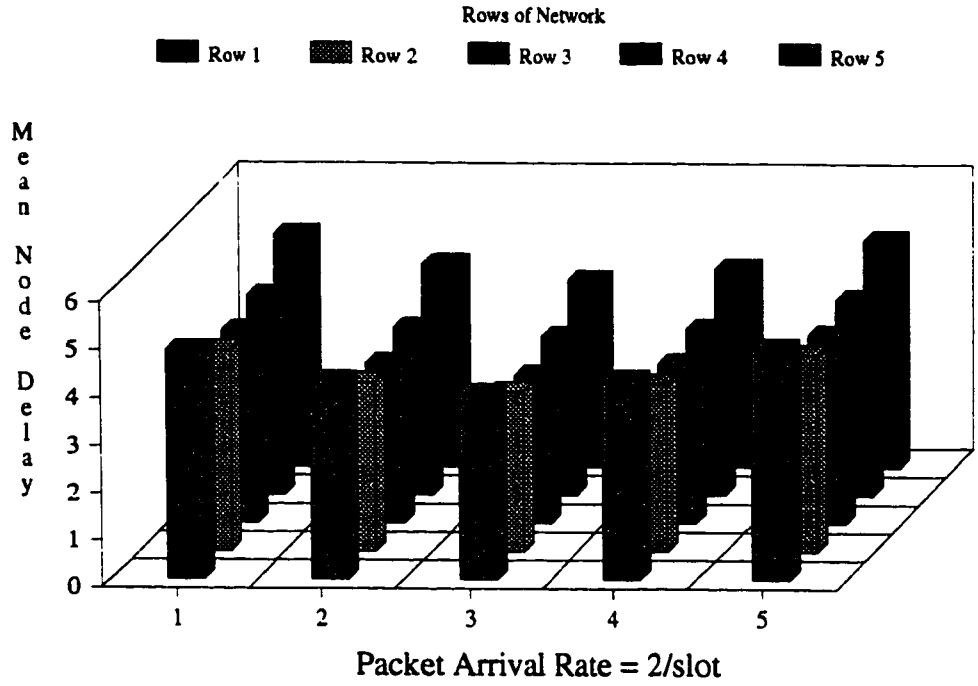


Figure 3.3.9 Node Delays - POR (2/slot)

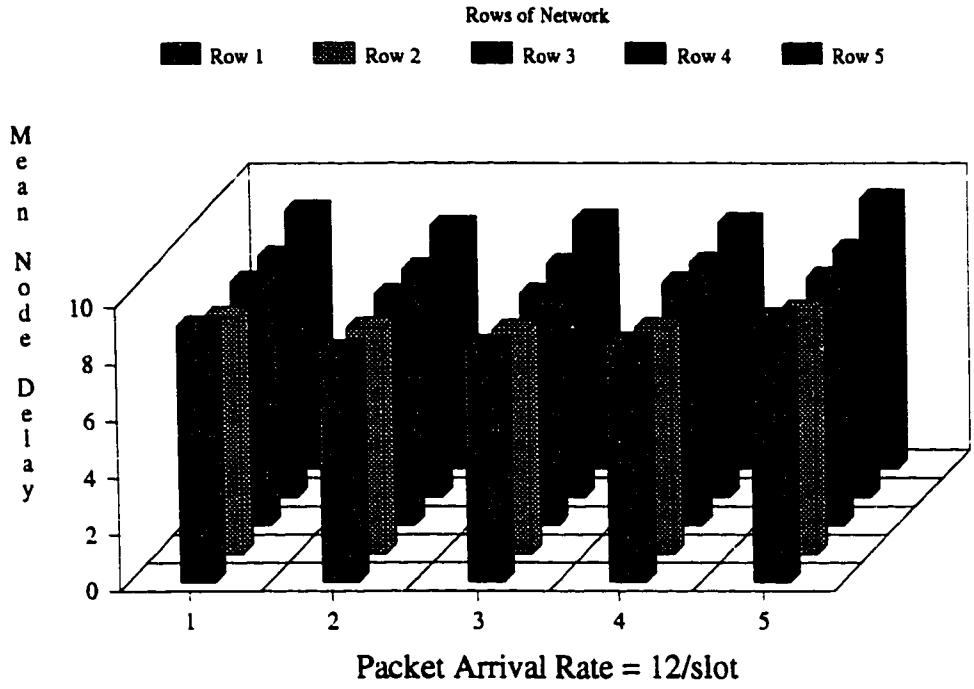


Figure 3.3.10 Node Delays - POR (12/slot)

Figure 3.3.11, Figure 3.3.12 and Figure 3.3.13 apply to the PDS algorithm. In this case, performance using post-routing access is substantially improved. Stable results were obtained for arrival rates below 14.5 arrivals per slot, which translates into about 60% of the maximum capacity and results in a 20% improvement over the other techniques. As with POR, the access delays remain low; only this time, the PDS routing strategy keeps transit delays low enough so that the net effect of post-routing access is favourable. Figure 3.3.13 shows the improvement in the distribution of delays across the network.

Until this point, it has not seemed that deflections have played a very important role in the performance of the network. The delay due to deflections has only been a small component of the total delay, and has been largely linear; that changed with post-routing access. With pre-routing access the increase in access delay was not associated with any significant trend in deflection delay. Close inspection of Figure 3.3.11 shows an increase in access delay that is accompanied by an increase in deflection delay. This seems to suggest that once improvements in access delay are made, the role of deflections in the network becomes more critical, and it is the matching of these two strategies that provides the system with better performance. Considering them independently may not be adequate. For example, poor deflection performance with POR produced a transit delay of 5 slots for an arrival rate of 11. With PDS, the same delay was not incurred until the arrival rate was about 13. A transit delay of 5, compared with an ideal delay of 3.33, implies that, on average, each packet on the network utilizes 1.67 slots more due to deflections and this throttles access to the network. This, in turn, increases access delays and affects system

capacity. With the PDS algorithm, it is worth increasing the deflection delay in order to reduce the access delay. The net effect is improved performance. Other phenomena related to access strategy will be discussed in the section that compares all algorithms. In the meantime, suffice it to say that the access algorithm plays a very important role in the delay and throughput performance of PMNet.

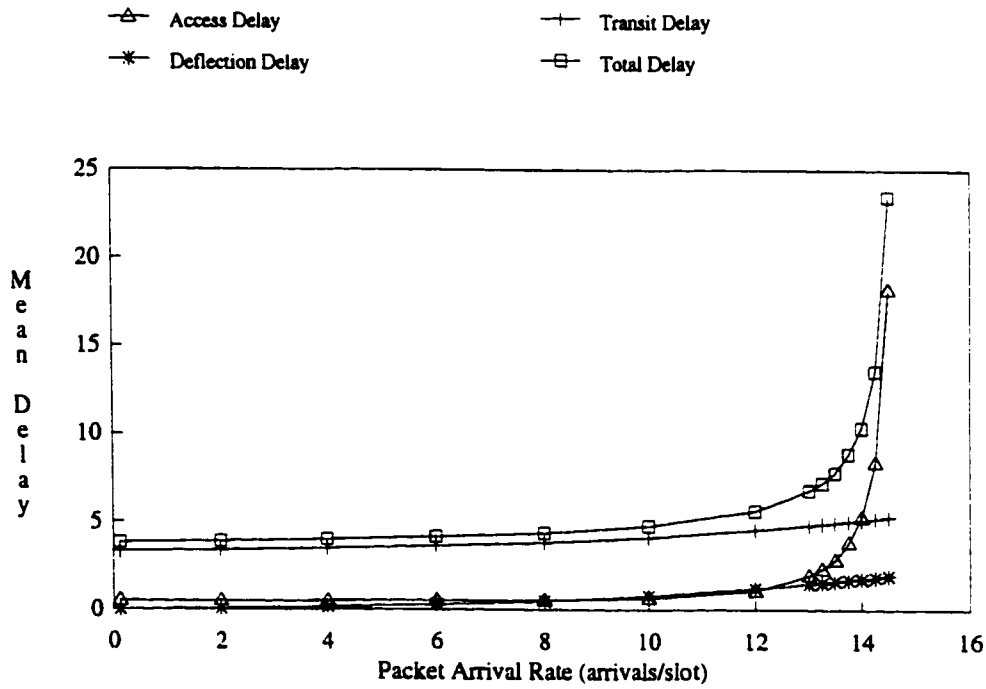


Figure 3.3.11 Delay Components - PDS

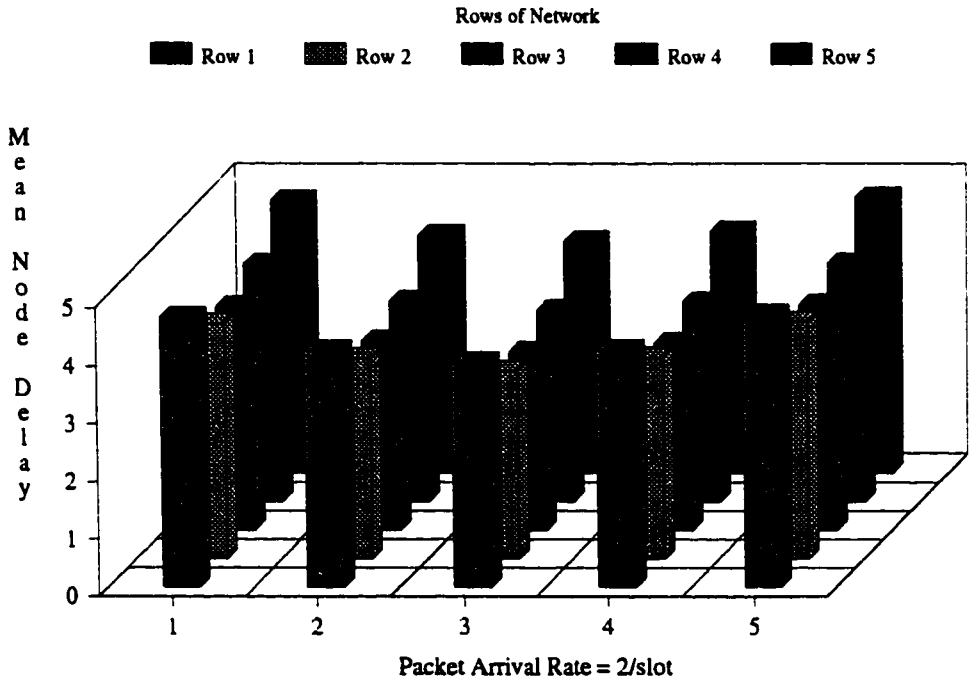


Figure 3.3.12 Node Delays - PDS (2/slot)

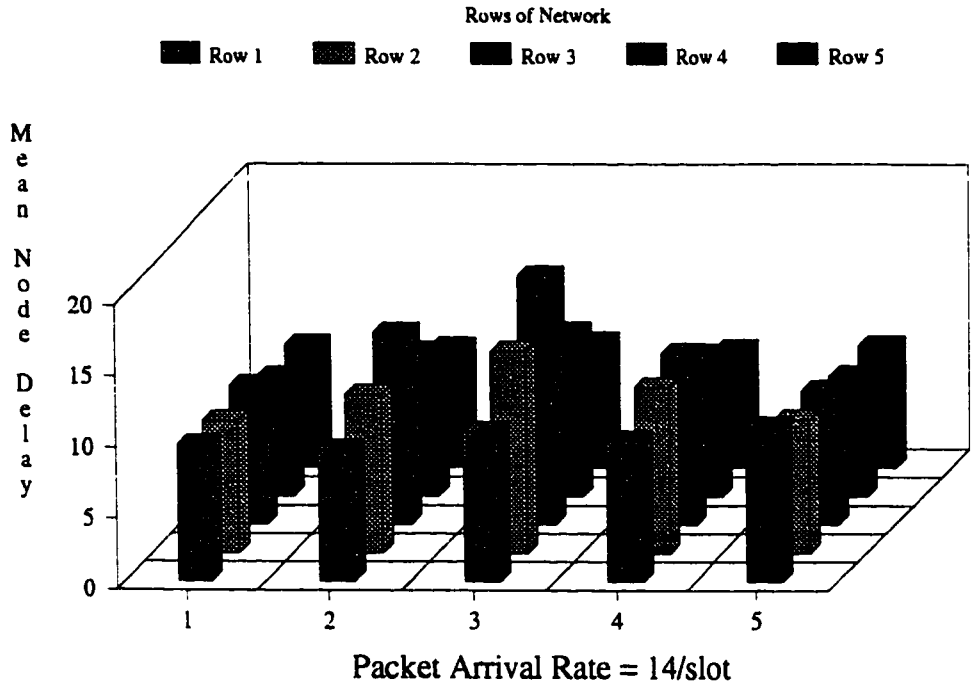


Figure 3.3.13 Node Delays - PDS (14/slot)

3.3.4 The Role of the Allocation Strategy

In total, three allocation strategies were investigated. All of the algorithms discussed in this section use the diagonal routing strategy. The nature of the other algorithms will be discussed in the section that compares all of the algorithms. As expected, the RDR algorithm produced the poorest results (see Figure 3.3.14). This is due to the negative impact of restricting the access to the network, as well as making no attempt to improve the quality of the decision in the allocation process. The

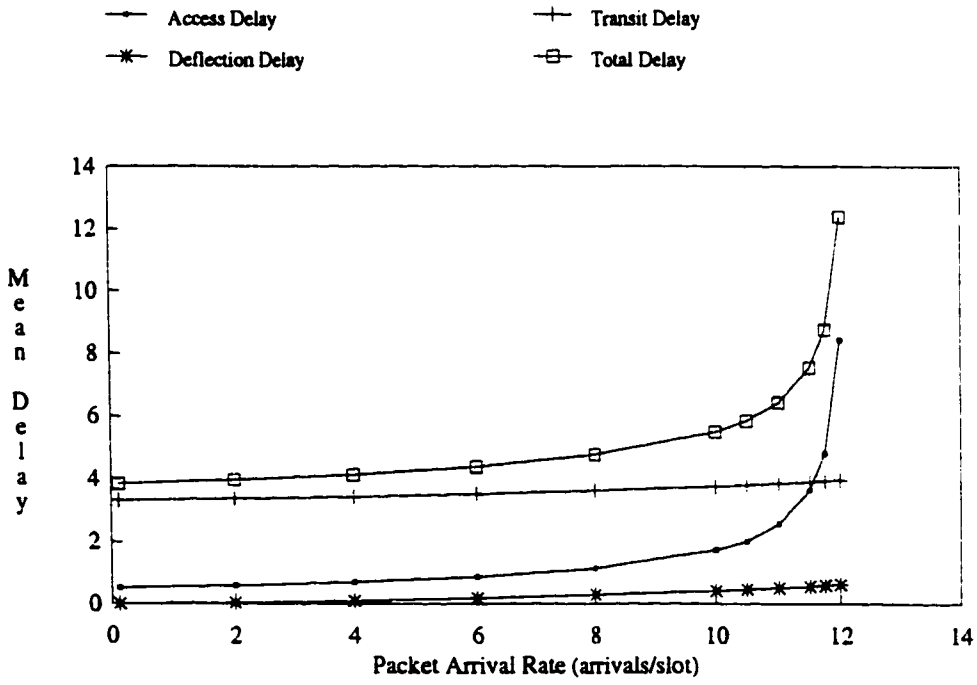


Figure 3.3.14 Delay Components - RDR

deflection component of delay is modest, although it is slightly higher than the deflection delay for the distance and secondary counter strategies, and is responsible

for the poorer performance. The system seems to reach capacity between 12.0 and 12.25 arrivals per slot, which translates into approximately 51% of the ideal capacity.

Figure 3.3.15 illustrates the large disparity of individual node delays, for an arrival rate of 12 messages per slot, which are typical of Pre-Routing access techniques. The fact that the centre node delay is so large compared to the others is indicative of the onset of instability, and is the limiting factor in determining the capacity of the network.

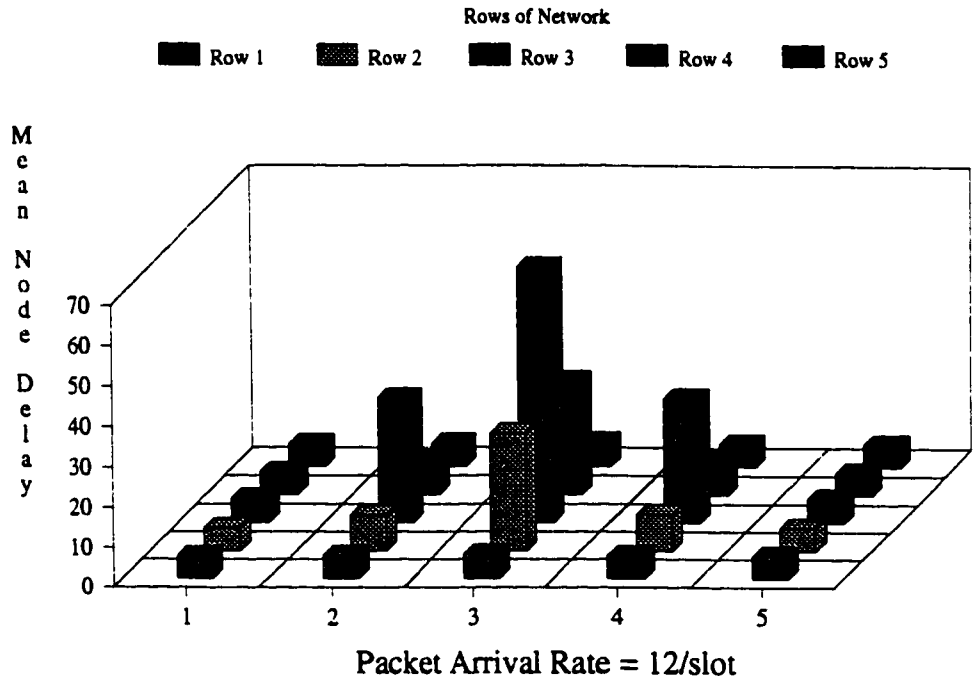


Figure 3.3.15 Node Delays - RDR (12/slot)

When comparing RDS and PDS, large gains were made by freeing up access to the network. This was due to the strength of the routing and allocation strategies, which allowed a favourable tradeoff between reducing access delays and increasing

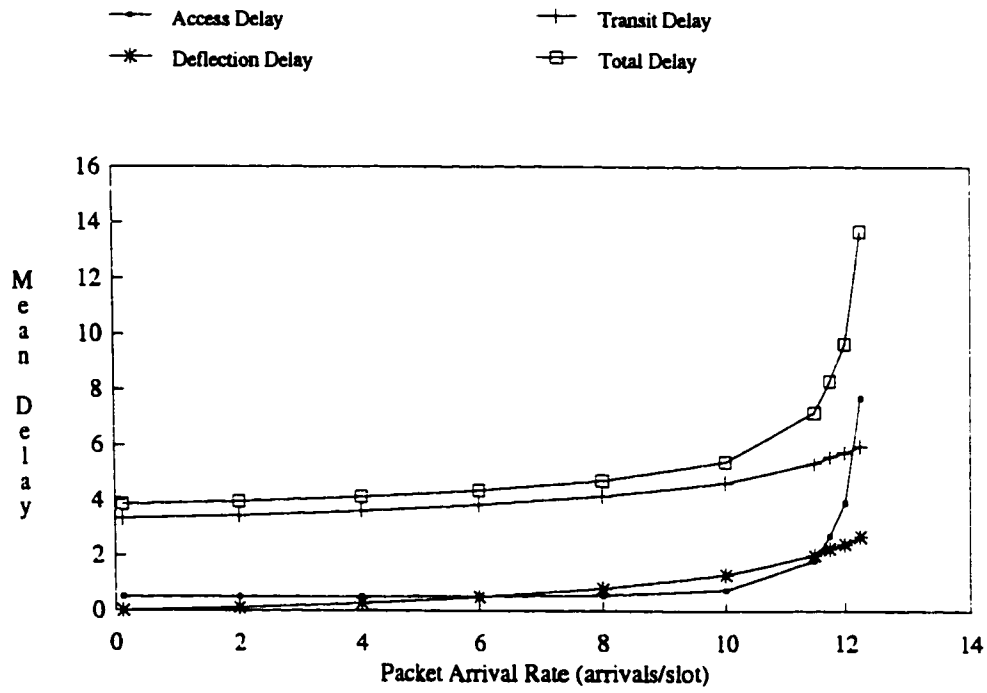


Figure 3.3.16 Delay Components - PDR

transit delays. Transit delays increase due to deflections, so it was the strength of the secondary counter-sorting strategy that made this tradeoff favourable. We have already pointed out the reasons for the random strategy not performing as well in the presence of deflections. Figure 3.3.16 confirms this point, showing only a very slight improvement in performance for PDR over RDR. In this case, note a more significant increase in delays due to deflections. Two factors are responsible for this trend. Post-Routing access allows packets to enter the network in a non-preferred direction which, in itself, is a deflection. The effect of this component becomes more pronounced at higher loads, and is responsible for the higher values of the deflection delay curve. The second factor is the same poorer deflection delay observed in the restricted access

curve. The net effect is only a slight improvement in performance over the Pre-Routing access technique. Figure 3.3.17 shows the more uniform individual node delays expected with a Post-Routing access technique. Note that, due to the non-complimentary nature of the access and allocation techniques, the more evenly distributed node delays result in very little improvement of overall performance.

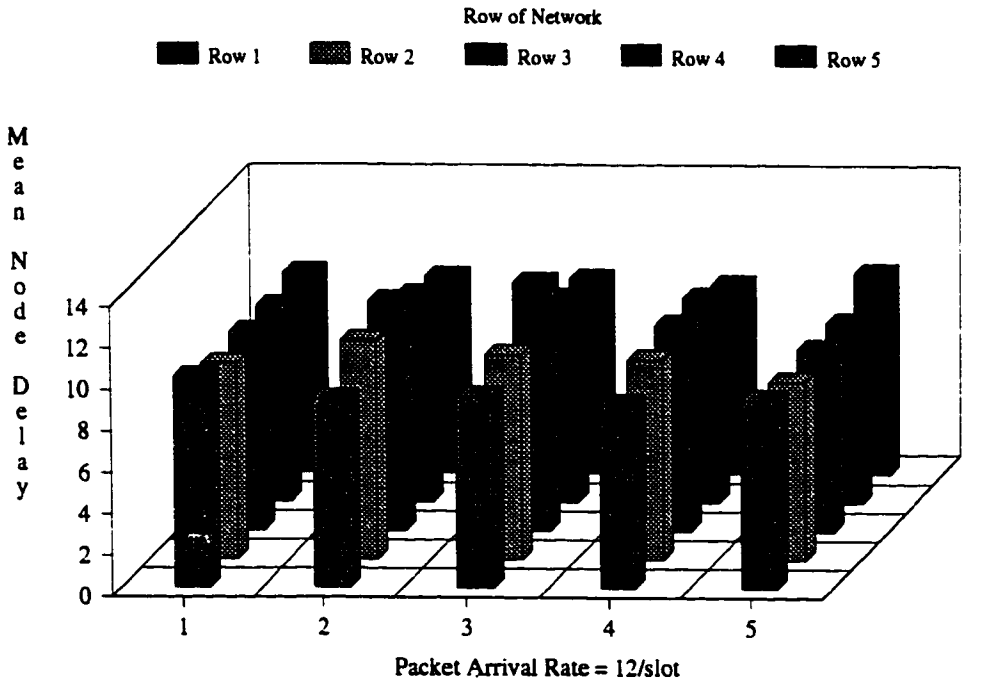


Figure 3.3.17 Node Delays - PDR (12/slot)

The allocation strategy of sorting on the distance to the destination was expected to perform better than the random strategy since it attempts to use locally available information to improve the quality of the decision. Figure 3.3.18 illustrates that, with Pre-Routing access, the improvement in performance of RDD over the random technique is very small. This is a direct result of the dominance of the access

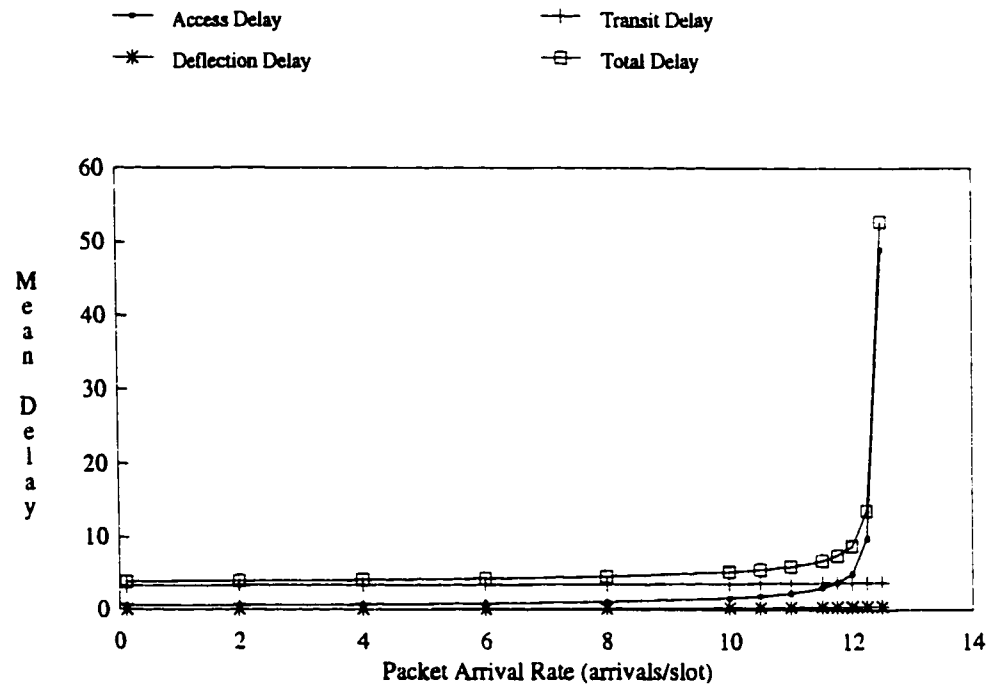


Figure 3.3.18 Delay Components - RDD

delay. With Pre-Routing access, the allocation strategy can have little effect since the performance of the network is almost completely determined by the access strategy. The deflection delay is seen to be a very small factor in the total delay, which indicates that (once on the network) the distance strategy performs well at high loads. Figure 3.3.19 shows the dominant effect of the access strategy at the centre node.

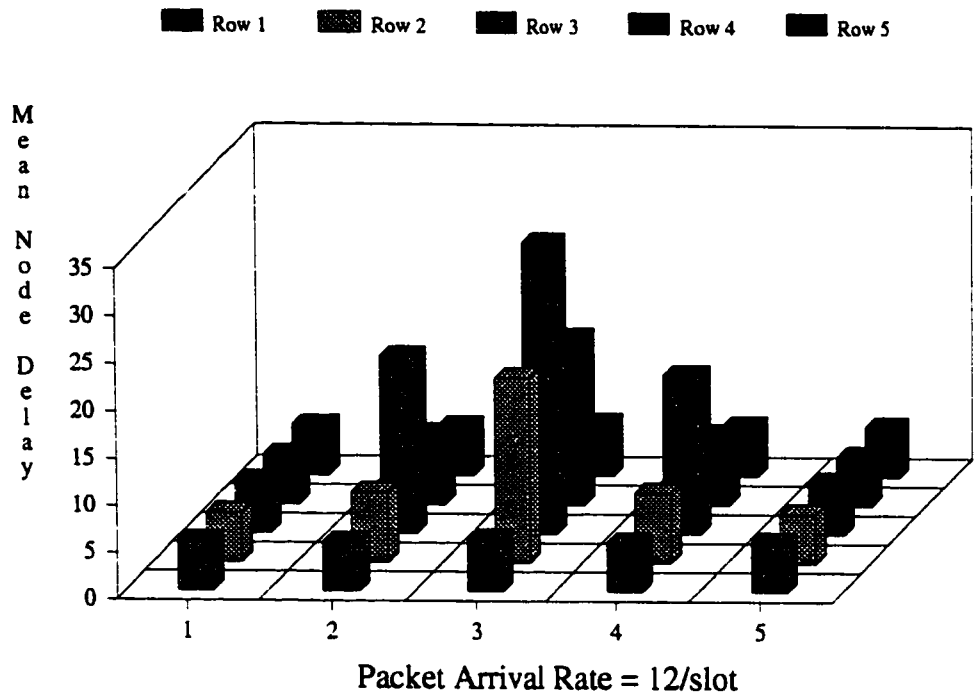


Figure 3.3.19 Node Delays - RDD (12/slot)

Unlike the random allocation strategy, the distance allocation strategy was expected to be strong enough to allow for a favourable tradeoff with access delay. This expectation was confirmed by the simulation results. Figure 3.3.20 shows the improved performance, very similar to that obtained with PDS over RDS. In fact, the PDD performance will be seen to be comparable with that of PDS. Figure 3.3.21

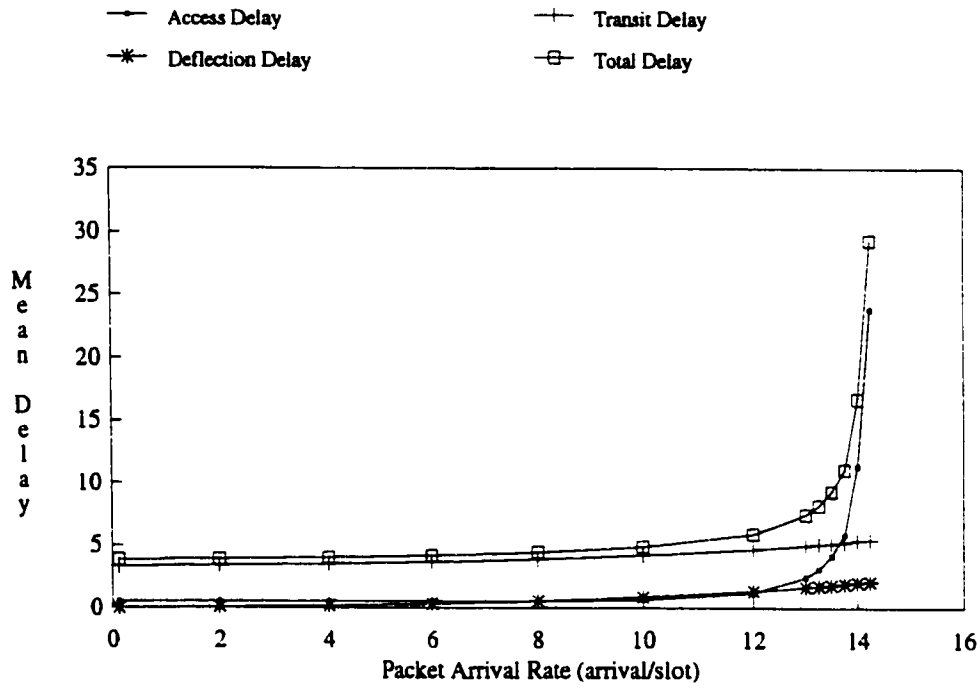


Figure 3.3.20 Delay Components - PDD

shows the improved distribution of node delays achieved with the free access technique.

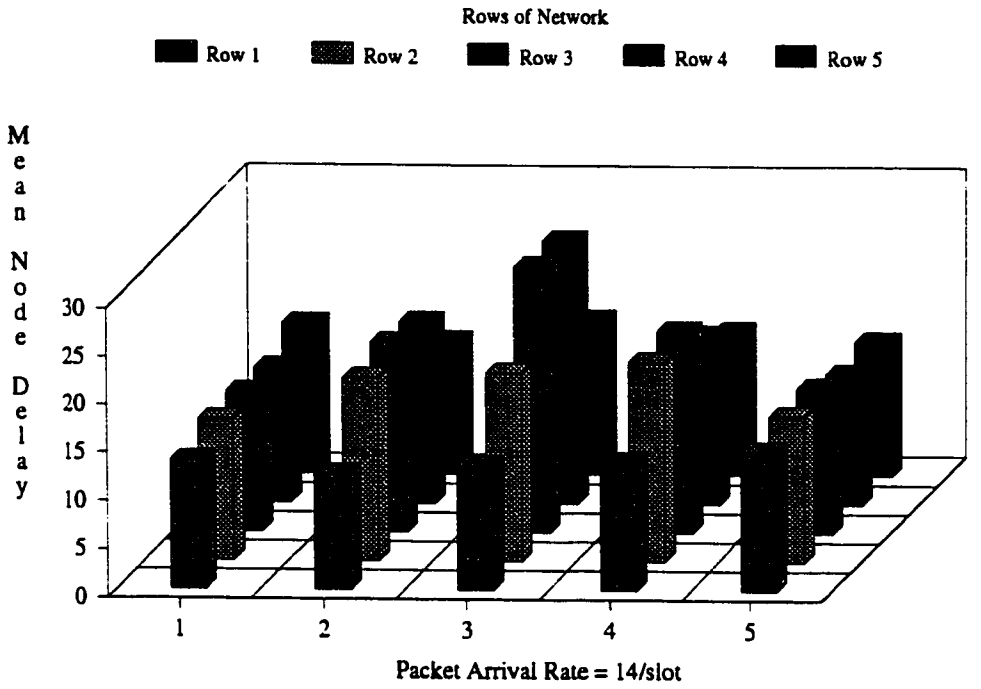


Figure 3.3.21 Node Delays - PDD (14/slot)

3.3.5 The Role of the Routing Strategy

In the preceding sections, the role of the access and allocation strategies have been investigated. The role of the routing strategy will now be considered. Specific comparisons between RDS and ROS, as well as PDS and POS, will be used to highlight certain aspects of the role that the routing strategy plays in delay and throughput performance. One might expect that the routing strategy would play the most important role of all. After all, it is the routing strategy that, to a large extent, determines the distribution of transit traffic in the network, and consequently impacts the availability of slots to access the network at the nodes. The routing can also be designed to avoid situations where deflections are necessary (as in the case of diagonal routing), and thereby reduce the requirement for channel allocation. The allocation strategy actually relates to the quality of the decision at a node, given that there is contention for certain of the outgoing links. A routing strategy can reduce the need for such decisions by avoiding contention. Given the apparent underlying role that the routing plays, it would seem reasonable that the algorithm would strongly impact the overall system performance. As we will see shortly, the actual dependency on routing is not as strong as originally anticipated.

Both diagonal and orthogonal routing have the important attribute of selecting the shortest path to the destination, given that deflections do not occur. This has the effect of freeing up network resources as quickly as possible, thus preventing entry traffic from being denied access to the network. The capacity of these networks is seen to be determined by the first input queue to lose stability and this, in turn, is determined by the distribution of transit traffic (or conversely, of vacant slots) on the

network. Although it has not been investigated in this work, a tradeoff might be considered between path length and the uniform distribution of transit traffic. Is it more important to uniformly distribute transit traffic and evenly build input queues even if longer path lengths are experienced, or should the shortest path on the network be taken for all of the reasons stated thus far? Our reasons for selecting only shortest path algorithms are as follows.

(a) One major motivation for this work is the simplification of the node design. It is clear that if a beneficial tradeoff exists between path length and performance, the desired path length must lie between upper and lower bounds. It must be longer than the shortest path, and must certainly be shorter than some maximum path lengths. As an example, consider Figure 3.3.22, where the deflection delays for the algorithms with the best performance are neither the shortest or longest delays. This implies that the routing algorithm must select paths with average lengths lying within certain limits, and which evenly distribute the availability of vacant slots on the system. The concern is that such an algorithm would complicate the node design by requiring that nodes perform differently depending upon their actual location in the network and possibly the size of the network. The distribution of destinations, as seen from any one node in the network, given the four possible outgoing directions, will vary from node to node. The routing at any node would therefore be specifically tailored for its position in the network. The exception to this lies in the grid networks which directly connect the edges of the networks, and thereby create a topologically homogeneous environment [Maxe85,Borg87].

Regardless of which node you select, the distribution of destinations is uniformly distributed across all outgoing links in such networks. The motivation for considering PMNet have been reviewed in the architectural overview and will not be repeated here.

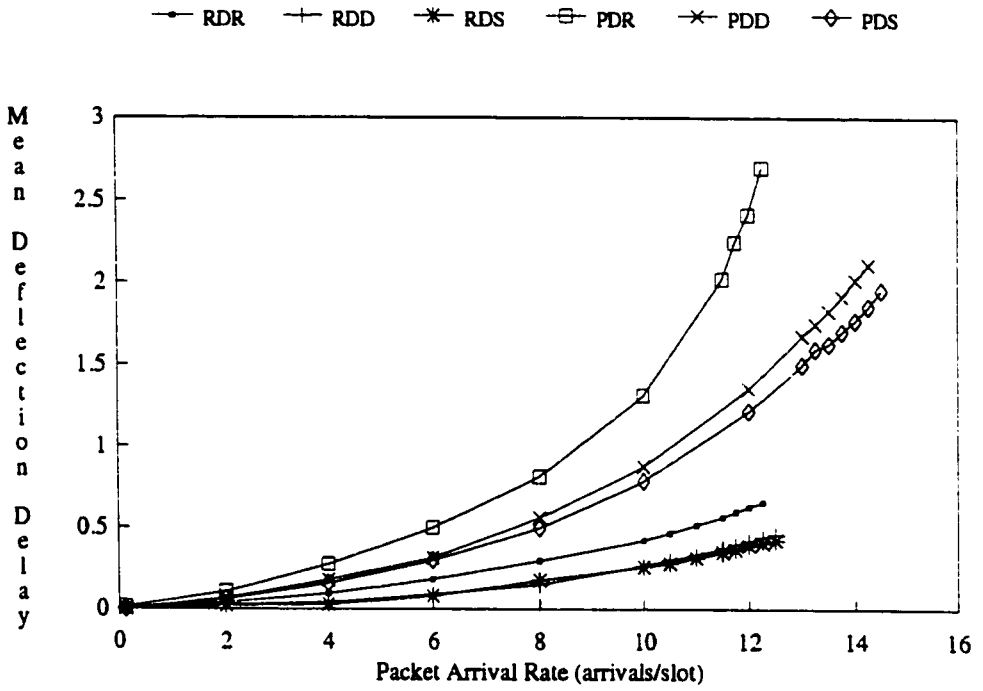


Figure 3.3.22 Allocation Comparison

(b) It is generally accepted that uniform traffic distribution across an entire PMNet is not likely to occur in practise, and that communities of interest will form. Such communities of interest will reduce the value of a tradeoff between path length and access delay since traffic will be relatively contained to a smaller portion of the network, thus reducing the tendency of certain nodes to support disproportionate amounts of transit traffic. The existence of communities of

interest will further complicate routing, which attempts to uniformly distribute slots across the network by introducing another degree of variability between nodes. Furthermore, communities of interest will reduce the impact of network architectures that attempt to derive benefit from directly connecting the edges of the network.

(c) The studies performed to-date apply to non-isochronous traffic. Isochronous traffic will likely require the use of a non-isochronous call setup packet procedure. The route taken by the call setup packet may determine the route taken by the balance of the isochronous traffic for the duration of the call. If shortest path algorithms are used, a call setup priority scheme can be included to ensure that the resources used for such traffic (which would be many times that of non-isochronous traffic) would be minimized by selecting a shortest path across the network. Routing strategies that intentionally take longer paths would not do this.

(d) The final point in this regard is concerned with the number of shortest paths that exist on the network. Given that so many shortest paths exist, there are opportunities to select a routing strategy that tends to evenly distribute the availability of vacant slots without adversely affecting the path length.

Further investigations may improve our appreciation of the viability of a tradeoff between path length and access delay, and may, in fact, lead to an algorithm that does not select the shortest path. Such algorithms are not, however, considered

here. The balance of this section will discuss the results obtained using the diagonal and orthogonal strategies.

By reviewing the simulation results, we found that the results for the secondary counter-sort allocation strategy best illustrated the important aspects of the role of the routing strategy. We begin our discussion with a comparison of diagonal and orthogonal routing using a pre-routing access strategy. Refer to Figure 3.3.23

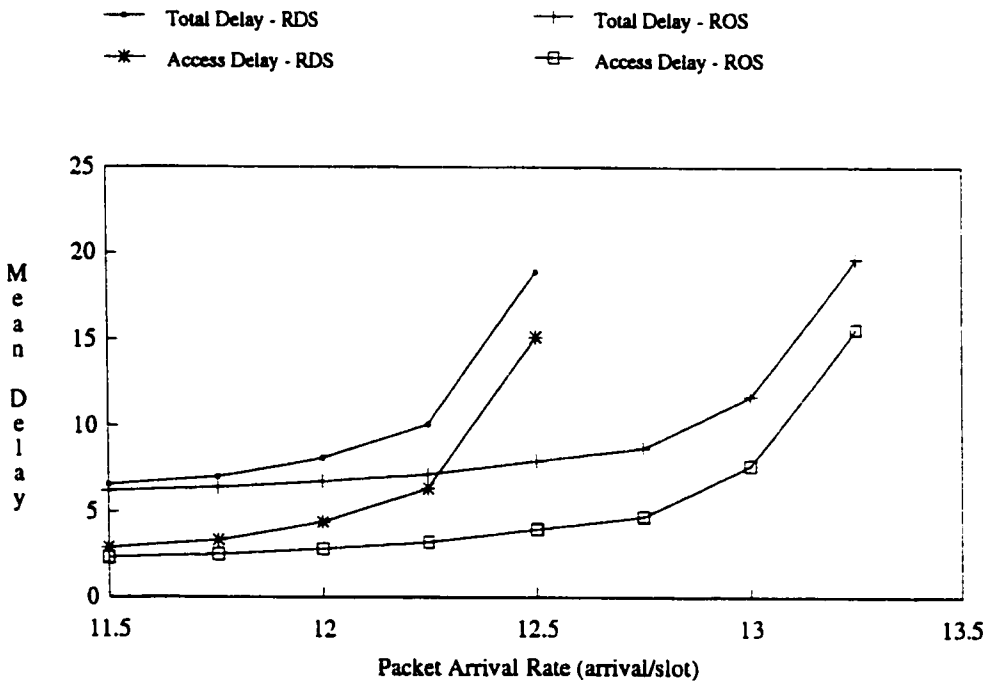


Figure 3.3.23 RDS and ROS Algorithms

Figure 3.3.23 show the average total and access delays for an expanded scale message arrival rate. We will use this to highlight an aspect of the relationship between the access and routing strategies when pre-routing strategies are used. Recall

that the pre-routing access strategy required that packets queue for entry to the network according to the most preferred route to the destination as selected by the routing algorithm. Differences in the routing algorithm may therefore affect the access performance indirectly by affecting the distribution of vacant slots on the network and by affecting the direction or directions in which packets are allowed to enter the network. This second point needs further clarification.

The orthogonal routing strategy always selects the row direction as the most preferred direction unless the destination lies directly along the present column. This has the effect of further restricting access to the network since the majority of destinations will require the selection of one of two rows. Access to the network then becomes dependent upon the vacant slot availability in two directions. The diagonal routing strategy tends to distribute the new arrivals more evenly to all input queues, and (one would think) take better advantage of the available vacant slots to enter the network. This should have a positive impact on performance which would favour the diagonal routing scheme. However, the simulation results show that the orthogonal routing strategy has lower access delays and better overall performance. This is attributed to the fact that the relative impact of uniformly distributing vacant slots across the network, which orthogonal routing does better than diagonal routing, is greater than the further restriction of network access experienced by the orthogonal routing strategy. In any case, the superior access performance of the orthogonal routing must be due, in a global sense, to better correlation between the distribution of vacant slots across the four links of a node and the distribution of new arrivals to the four input queues at that node. This results in fewer 'missed' vacant slots at a node,

and therefore smaller average access delays and a larger throughput. It is obvious from this, that an access strategy could be selected that matched the availability of vacant slots at the node. This, however, leads to concerns of node complexity and fairness to users. One access method that addresses both of these concerns is the post-routing access algorithm.

Another aspect of the routing strategy can be seen in Figure 3.3.24, which illustrates the delay due to deflection for both algorithms. As would be expected from the original design of the routing strategies, the relative performance of diagonal routing for transit traffic is superior. The reasons for this have already been explained. Routing has a direct impact on transit delay performance.

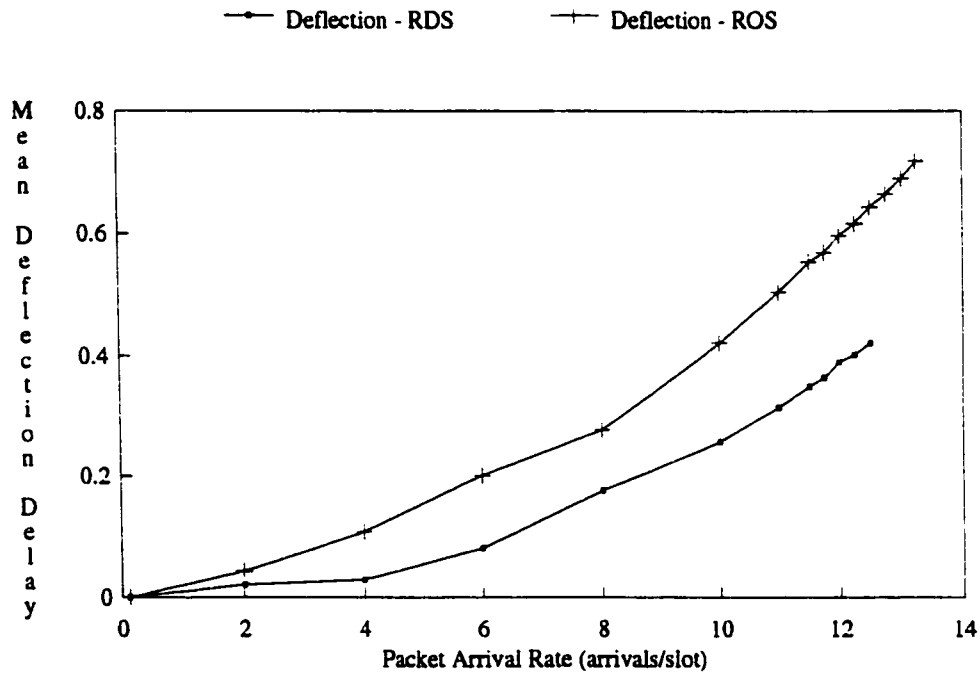


Figure 3.3.24 RDS and ROS Deflection Delay

In summary, the routing strategy plays three roles when used with pre-routing access techniques. It affects access delay by determining the availability of vacant slots at a node and by affecting the distribution of new arrivals to input queues. It also directly affects the transit delay by appropriately impacting delays due to deflections. The net effect of this is that the routing strategy seems to play a significant role in determining network performance when using pre-routing access techniques. This is supported by the differences in the RDS and ROS curves.

Post-routing access partially decouples the routing and access techniques. Obviously, the routing itself still impacts the access performance through the distribution of vacant slots, but to a lesser degree. It does not at all impact the distribution of new arrival access directions since a single queue is maintained at each node. If a non-zero queue exists at a node, then every vacant slot at a node will be used in an attempt to empty that queue. Packets are considered on a first-come-first-serve (FCFS) basis. We have already seen that, in general, this results in a significant improvement in the delay and throughput performance. We will now discuss the role that routing plays in such improvements.

Figure 3.3.25 compares the POS and PDS algorithms. The most important thing to note is that the performance of the two algorithms are comparable. The immediate observation is that the difference in routing strategies has very little effect on the results. This is consistent with the observations of the pre-routing access techniques. The performance of the network is dominated by access delays, and the impact of the routing function on access delays has been lessened by the use of post-

routing access. The decoupling of access delays and routing strategies leaves only the transit delay performance to differentiate between the two routing strategies.

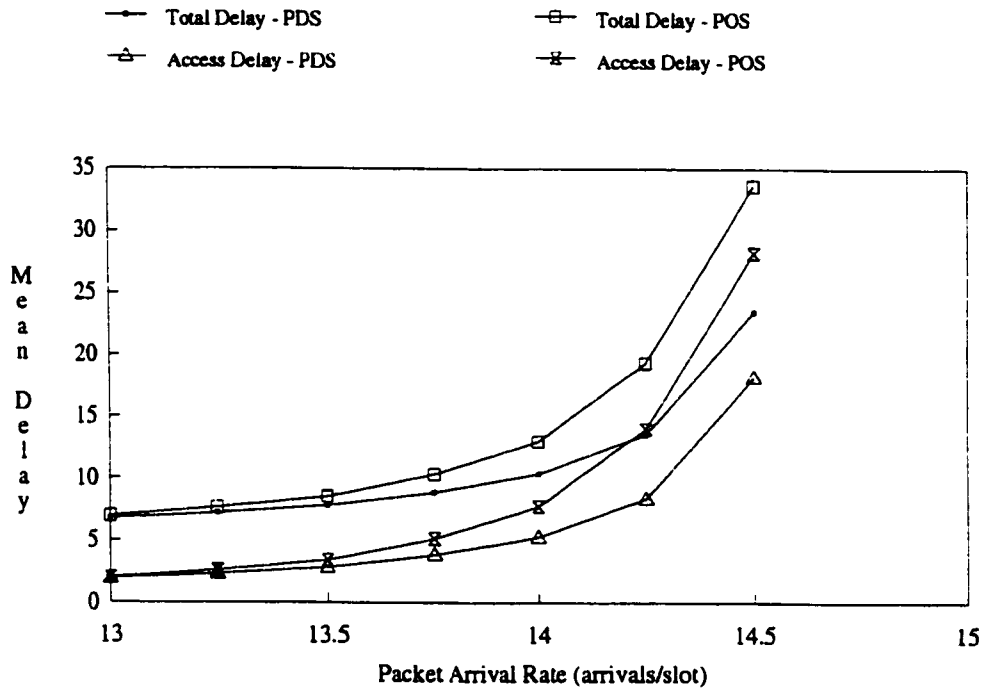


Figure 3.3.25 Comparison of PDS and POS

Figure 3.3.26 compares the deflection delay of the two algorithms, and it is seen that the relative differences are not as pronounced as they were with pre-routing access. In the case of post-routing access, the comparison is confused by an additional factor introduced by the access method. With post-routing access, deflection may occur at the access point to the network. Such deflections would not occur with pre-routing access. When deflections are allowed to occur on access to the network, they distort the normal perception of transit traffic deflections due to an increased probability of deflection. Any packet entering the network has, at most, two directions

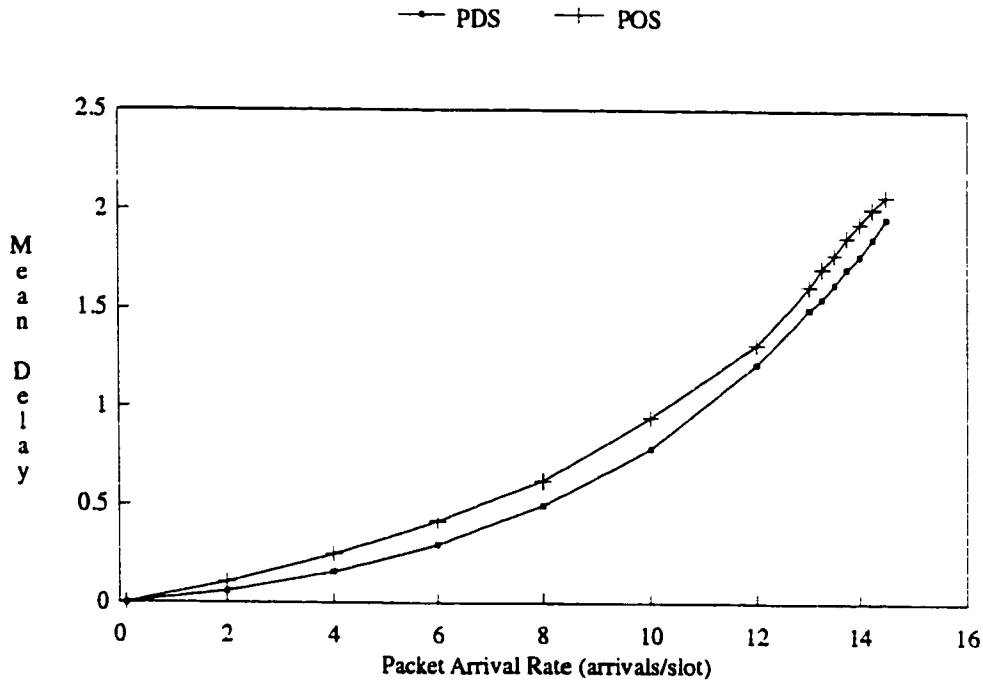


Figure 3.3.26 Deflection Delay Comparison

that it can go to, to get closer to its destination. The ability of a packet to take that direction will be dependent upon the other packets present at the node and the allocation strategy used. The secondary counter-sort strategy will tend to give preference to packets which have fewer deflection-free decisions left and, in some sense, are closer to their destinations. Newly entered packets will, on average, have the most deflection-free decisions left and be farthest from their destinations. This will result in a larger secondary counter, and poorer allocation than most transit traffic. The result is that the probability of deflection is higher for newly entered traffic. The deflection delays shown in Figure 3.3.26 are dominated by deflections associated with packet entry, which is roughly the same for both routing algorithms. The differences

due to deflections while in transit are, therefore, less pronounced with post-routing access. The result is only slightly better performance for the diagonal routing strategy.

In closing this section, we can say that the role of the routing strategy appears to be significantly diminished in cases where post-routing access methods are applied. The reason for this is that the routing and access techniques become more decoupled and, since the performance of the system is still dominated by the access delay, the routing strategy has less impact on the network performance. Such a phenomenon could play an important role in selecting an easily implementable algorithm if the effect on performance is not significant. This could become a valuable tradeoff in the design of a node.

3.3.6 Comparison of Algorithms

In this section, we shall briefly compare the simulation results for all of the algorithms, reconfirming several of the points already made. Previous figures have selected various algorithms for comparison, in order to highlight certain points. This section will show that the behaviour illustrated in the previous sections is consistent across all of the algorithms.

Figure 3.3.27 shows the total delay for all of the pre-routing access algorithms in a 5X5 PMNet. Only the pre-routing access curves are included since inclusion of all of the algorithms makes the figure too cluttered for presentation. A separate figure

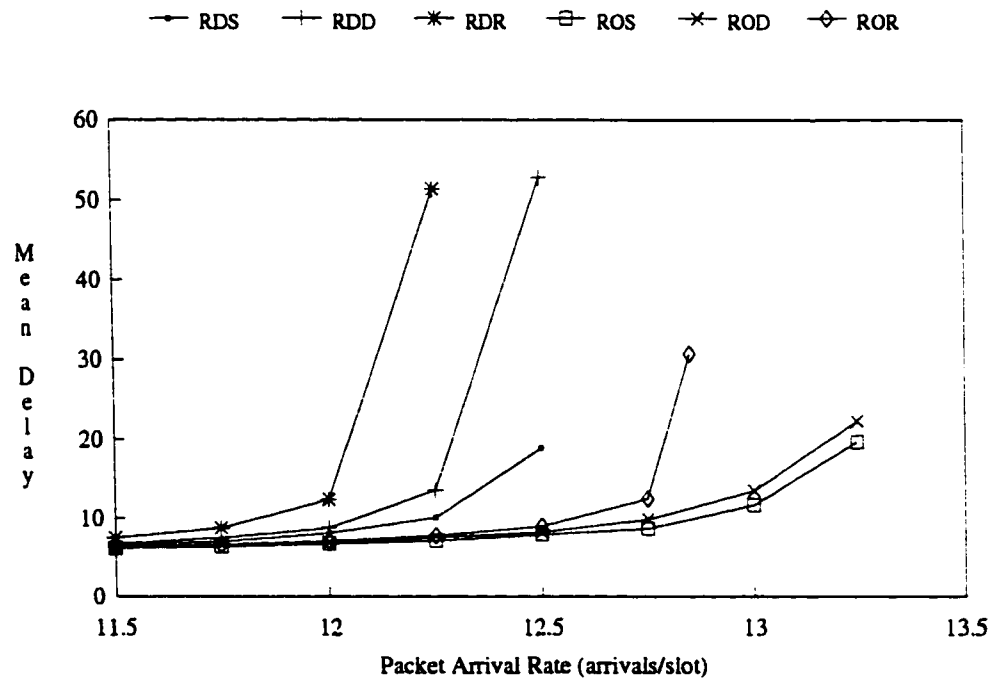


Figure 3.3.27 Pre-Routing Access Algorithms

will be used for the post-routing access curves. Note that the message arrival rate scale is expanded since, at low loads, all of the algorithms produce nearly identical results. This highlights the fact that all of the algorithms considered here are shortest path algorithms. Given that there is a light load on the system, there will be no deflections and the access to the system will not be adversely affected by transit traffic. This means that there is very little differentiation between the curves at light loads. As the load increases, the different effects of the strategies can be seen. Note that the orthogonal routing strategies are better than the diagonal routing strategies in every case. This is exactly as stated in the preceding section, where routing strategies were compared. With pre-routing access, the routing strategy greatly affects the access delays and, since orthogonal routing more evenly distributes transit traffic, the performance is better. For pre-routing access, the routing strategy is seen to be the largest differentiator between algorithms. Within the group of orthogonal routing algorithms, the algorithms are differentiated by their allocation strategy. In fact, for both the orthogonal and diagonal routing strategies, the dependence upon the allocation strategy is the same. In each case, the secondary counter-sort strategy produced the best results, followed fairly closely by the distance-sort strategy, with the random allocation strategy being significantly worse than the other two. The comparisons of all of the pre-access strategies is seen to be consistent with the observations of the preceding sections.

Figure 3.3.28 compares the post-routing access algorithms. This confirms our previous assertion that the routing strategy, due to the decoupling affect of the post-routing access strategy, does not affect performance as significantly as the allocation

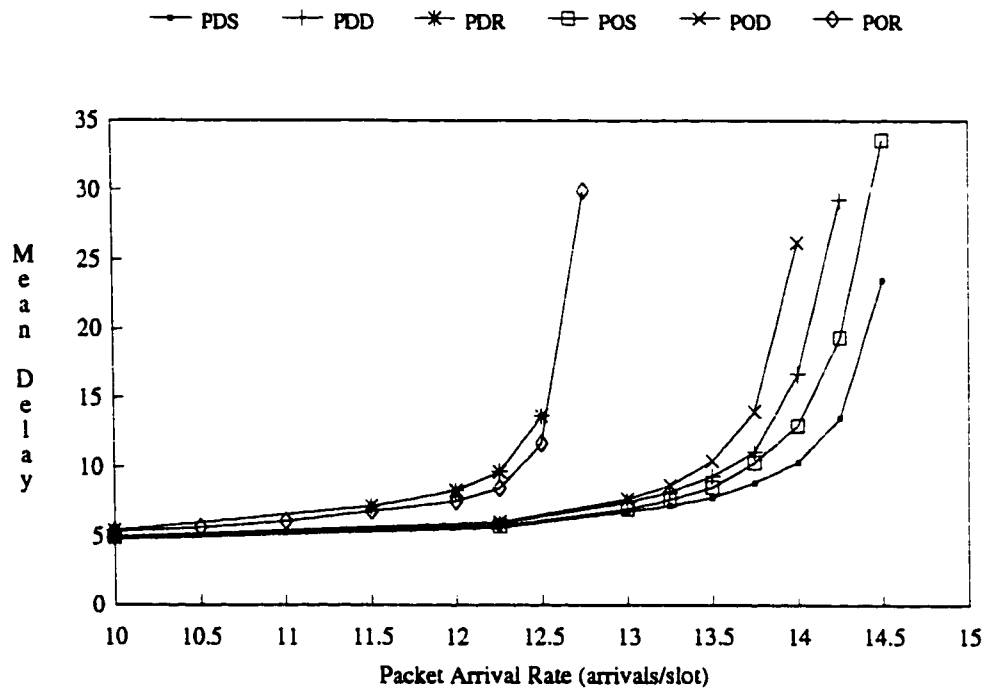


Figure 3.3.28 Post-Routing Access

and access strategies. The figure can be considered as three pairs of curves, depending upon the allocation strategy used. The first pair, the two curves with the best results, is associated with the secondary counter-sort strategy. The best curve uses diagonal routing, but the difference between that curve and the orthogonal routing curve is not substantial. The second pair of curves represents the distance-sort allocation strategy with, once again, a small difference between the curves due to the routing strategy. These two pairs of curves are grouped closely together when compared with the third pair of curves. This is due largely to the similarity in the two allocation strategies used, and the dissimilarity with the random allocation strategy of the third pair of curves. The third pair of curves is grouped very closely together. It

is interesting to note that, in this case, the orthogonal routing strategy produces slightly better results than the diagonal routing strategy. This seems inconsistent with the other post-routing access curves. The improved performance of the orthogonal routing strategy is due to the access delay since its deflection delay is actually longer than that of the diagonal strategy. It appears that the allocation strategy can play a favourable role in reducing the access delays, which is noticeable when post-routing access is used. The reasoning for this is as follows.

The allocation strategy determines the quality of the decision at a node when contention exists. This directly affects the deflection delay performance of the network, which, in turn, affects the average path length of a packet on the network. Increasing the average path length, directly affects access delay by increasing the number of transit packets in the system. Post-routing access lessened the impact of the routing strategy on the access delay to the point where the stronger diagonal routing strategy, when coupled with a reasonable allocation strategy, produced better performance than the orthogonal strategy. (This was not the case for pre-routing access.) As the load increases, the performance of any algorithm is largely affected by the allocation strategy since contention occurs so frequently. If the allocation strategy is rendered useless, by randomizing it, the advantages of one routing strategy over another are less apparent. The diagonal and orthogonal routing strategies are relatively comparable, as witnessed by the similarity in the other post-routing access curves. This being the case, randomizing channel allocation has the affect of lessening the deflection delay advantages of the diagonal routing strategy. With this advantage gone, the improved transit traffic distribution of the orthogonal routing strategy, which

still exists with post-routing access, gives the orthogonal routing algorithm slightly better performance.

The final comparison between Figure 3.3.27 and Figure 3.3.28 compares the access strategies that have the strongest affect on performance. In every case, the delays are dominated by the access delay component, and the reasons for this have been discussed at length. It is important to note, however, that pre-routing access algorithms may perform better than post-routing access algorithms, depending upon the allocation strategy. As an example, consider the pre-routing access, orthogonal routing algorithms as compared with the post-routing access, random allocation algorithms. The advantages of the post-routing access strategy are outweighed by the negative impact of random channel allocation. Random channel allocation has the affect of increasing deflections, and thereby increasing access delay, through an increase in the average path length of a packet. In the example we have selected here, the increased component of access delay due to the increased path length attributed to the allocation strategy is larger than the decreased component due to the access strategy.

As a final point, note that the best performance is obtained by algorithms with post-routing access and a reasonable allocation strategy. The least significant of the strategies, in this case, is the routing strategy. The relative importance of the routing strategy increases when pre-routing access is used and, to a lesser extent, when poor allocation strategies are used.

3.4 A Generalized Flow Rate Model

In this section, we develop a model which may be used to accurately approximate the detailed link flow-rates for the deflection routing algorithms defined above. Recall that, in general, each algorithm consists of three distinct strategies (access/routing/allocation) and that the manner in which these components interact to affect the performance of the system is dependent upon the algorithm. An iterative approach to the problem will be employed where the detailed link flow-rates from one iteration are used to update the input flow of new packets in the next iteration which, in turn, is used to update the detailed link-flows. The process is continued until acceptable accuracy is achieved. In this analysis, a memoryless source model is assumed. The resulting calculated flow-rates are found to closely approximate those which are established under an infinite buffer assumption.

The common starting point for the analysis is the individual link flow at a node. Define F_{kl}^{u} as the steady state flow from node k destined to node u out link l. Note that the only state information necessary for a routing decision is the destination of each of the packets at the time the routing decision is made. These individual flow-rates are then used to define the initial conditions on the input lines for any routing decision. In the case of pre-routing access, this is the only input to the routing decision. For post-routing, we must also consider new packets entering at the node since they will also take part in the routing decision. The manner in which these flow-rates are used to condition the probability expressions associated with the routing decision is thus dependent upon the algorithm. First, define $P_{routing}$ as the steady state

probability of a given set of output flows from a node, conditioned on the inputs to the routing decision. This probability will be defined by the particular routing and allocation strategies employed by the algorithm. Also define P_{init} as the steady state probability of a given set of initial conditions on the input lines to the node in question.

The input to the routing decision with post-routing access is determined by the packets arriving over the input links to the node and by the new packets entering at the node. Define a set of input conditions as $\zeta=(u,v,w,x)$, where u,v,w,x represent the destinations of packets received on each input line of direction 0,1,2 or 3. The condition of no packet arriving is the null condition. Define $\varepsilon=(i,j,k,l)$ as a set of destinations for new packets which enter the network at the node from the input queue. Finally, define $\xi=(q,r,s,t)$ as the set of destinations of packets mapped to the outgoing directions 0,1,2 and 3. In general, for post-routing access, we may now define the probabilities of interest.

$$P_{init} = P_r(\zeta) \tag{3.4.1}$$

$$P_{entry} = P_r(\varepsilon/\zeta) \tag{3.4.2}$$

$$P_{routing} = P_r(\xi/\zeta,\varepsilon) \tag{3.4.3}$$

The conditioning of the problem is different for pre-routing access. In this case we have

$$P_{init} = P_r(\zeta) \tag{3.4.4}$$

$$P_{routing} = P_r(\xi/\zeta) \tag{3.4.5}$$

$$P_{entry} = P_r(\varepsilon/\xi) \tag{3.4.6}$$

Since the conditioning of the expressions is different in the two cases, the resulting equations for steady state mean link flows must also be different. In the following equations, the probability of flows out of a node are defined, in general terms, by summing over all possible input link conditions and over all entry traffic conditions. For post-routing, the expression becomes

$$P_r(output\ flows) = \sum_{all\ \zeta} \sum_{all\ \varepsilon} P_{routing} P_{entry} P_{init} \tag{3.4.7}$$

and for pre-routing the expression is

$$P_r(output\ flows) = \sum_{all\ \zeta} \sum_{all\ \varepsilon} (1+P_{entry}) P_{routing} P_{init} \tag{3.4.8}$$

Individual link flows F_{kl}^{u} have already been defined. A transformation is now introduced that defines link flows into a node in terms of the output flows from adjacent nodes. Note that the analysis requires the flows to each destination on each link out of each node in the network. In actual fact, due to the symmetry of the network, it is only necessary to find the flows at the topologically unique nodes, since flows at other nodes may be directly deduced from these. The number of topologically unique nodes is dependent upon the deflection algorithm used. In order to define the transformation, Φ, let the coordinates of node k be denoted by its

column and row address (i,j). Φ maps each input link l at node (i,j) into the output link at the adjacent node, which is connected to l.

$$\Phi(i,j,l) = (i + [\delta(l) - \delta(l-2)], j + [\delta(l-1) - \delta(l-3)], mod4(l+2)) \quad (3.4.9)$$

For simplicity of notation, we will denote $\Phi(i,j,l)$ as $\Phi(k,l)$. We are now in a position to present the analytic model more rigorously. The post routing algorithms will be discussed first.

3.4.1 Post-Routing Access

The mean number of busy slots into node k, $\overline{F_k}$, may be determined by summing the input link flows over all input links to all destinations, excluding those destined to node k itself. Assume that there are l_k links at node k and that the network is a grid of N*M nodes.

$$\overline{F_k} = \sum_{l=0}^{l_k} \sum_{i=0}^{NM-1} F^{i}_{\Phi(kl)} - \sum_{l=0}^{l_k} F^{k}_{\Phi(kl)} \tag{3.4.10}$$

The mean number of empty input slots will be used to constrain the number of packets allowed to enter the network. We now define $P^{u}_{E_k}$ as the probability of a packet entering from the input queue at node k with destination u. Note that this probability will vary from node to node since it is dependent upon the transit traffic at the node. We assume that the arrival rate of new packets to node k is defined as λ_k and that the destination addresses are uniformly distributed among all possible NM-1 destinations.

$$P^{u}_{E_k} = \frac{\lambda_k}{(NM-1)(1-\overline{F_k})} \tag{3.4.11}$$

The next necessary probability is related to the input link conditions. In general, there may be up to l_k packets received at node k in any slot. To formulate the problem, assume that l_k=4 and that those inputs have packets destined to nodes (u,v,w,x). For convenience, assume that when u,v,w or x equals k, it represents both

an empty slot condition and a packet destined to node k condition. This simplifies the analysis since (u,v,w,x) now completely specifies the conditions on the input links at the node. Now define $P^{u}_{input_u}$ as the probability that input link l at node k has a packet destined for node u. In addition, define $P_{input_l}(u,v,w,x)$ as the probability that, in the same slot, link 0 at node k has a packet destined for node u, link 1 has a packet destined for node v, and so on. The input condition at any one input link at node k may then be defined. Finally, consider that any condition where an input link slot is empty, or contains a packet destined to k, is an opportunity for a new packet to enter the network. The probability of entering the network conditioned upon the availability of an empty slot is given by (3.4.11). The situation of neither an arrival on an input link nor a new entering packet is defined for u=k. The contribution of one input link towards the total input to the routing decision may written as

$$P^{u}_{input_u} = \begin{cases} F^{u}_{\Phi(kJ)} + (1-\overline{F_k})P^{u}_{E_l} & u \neq k \\ (1-\overline{F_k})(1-\sum_{i=0}^{NM-1} P^{i}_{E_l}) & u = k \end{cases} \qquad (3.4.12)$$

Using (3.4.12), all the input conditions to the routing decision at the node may be defined as

$$P_{input_l}(u,v,w,x) = P^{u}_{input_{l_0}} P^{v}_{input_{l_1}} P^{w}_{input_{l_2}} P^{x}_{input_{l_3}} \qquad (3.4.13)$$

We have now completely specified $P_{input_l}(u,v,w,x)=P_{entry}P_{init}$ in (3.4.7).

In order to solve for the output flows from a node, we must now consider the routing algorithm and the manner in which it maps its inputs to the output ports. As was pointed out earlier, this mapping function actually consists of both the routing and allocation algorithms and is embodied in the preference vectors for the node processing. In general terms, the routing algorithm may be described as consisting of two basic operations. The first operation is to differentiate the input packets according to some attribute. There may be an arbitrary number of levels of differentiation, depending upon the required characteristics of the routing function. This differentiation operation allows each packet, or group of packets, to be considered individually in the second operation, which is called the conditioning operation. After being appropriately differentiated, packets are subjected to a set of conditions to determine the output link to which they will be mapped. The combination of differentiating packets, and then processing them according to a set of conditions, allows the routing decision to be tailored to the desired characteristics.

As an example of the above, the preference vectors for PMNet are seen to consist of two levels of differentiation and three conditions. The first level of differentiation is defined by the allocation strategy, which attempts to order the input packets according to some attribute (Secondary counter, Distance to destination or Random). This mechanism provides selective preference in the routing decision, and thus tailors its characteristics. For example, Random allocation offers no differentiation, while Secondary counter-sorting is designed to improve the deflection performance of the system. The allocation strategy also resolves the contention problems that would otherwise result in a deflection routing switch. The second level

of differentiation is determined by the orientation of the input, and is used to improve the routing decision based upon the destination of the packet relative to its present position in the network. After differentiation, three conditions determine which output link a packet may use. The first condition is given by the state of the links (already allocated or unallocated). The final two conditions are given by the state of the row and column counters for the packet. Changing these conditions changes the actual preferred paths of the packets from source to destination in the network.

For the purposes of our analysis, we consider the conditioning stages to be deterministic. That is to say, given a differentiated set of inputs, the same mapping to output links will occur with probability 1. Given a set of inputs to the routing decision, however, there may be several possible differentiation decisions. The net result is that any one set of inputs to the routing decision may be mapped in several ways to outputs, and the output flows must take this into account. The number of decisions possible is dependent upon the allocation algorithm. If we assume that there are n packets to route, then for Random allocation, these packets may be selected in any order and the number of possible decisions is given by

$$N_{decisions_R} = n! \tag{3.4.14}$$

For Secondary counter-sorting allocation, consider that there are n packets to route and that these n packets consist of k groups of n_k packets each, where the packets in each group have the same value of Secondary counter. In this case, the number of possible decisions is

$$N_{decisions_S} = \prod_{i=1}^{k} n_i! \tag{3.4.15}$$

For Distance allocation, consider that there are n packets to route and, as above, there are k groups of n_k packets, each with the same distance to their destinations. The number of decisions is given by (3.4.15).

We can now define $P_{routing}$ in (3.4.7). The probability of selecting any one decision from the number of possible decisions is uniformly distributed. Define $P_r(R_l=u/i,j,m,n)$ as the probability that the routing algorithm makes a decision where a packet destined for node u is routed out link l, given that the input to the routing algorithm consisted of packets i,j,m and n. Also define $P^u_{PV_l}$ as the probability that, given the differentiated input packets, the routing decision will map a packet destined for u onto output link l. Recall that this probability will depend upon the preference vectors, and can only take on the values of 0 or 1. We may therefore write

$$P_r(R_l=u/i,j,m,n) = \frac{P^u_{PV_l}}{N_{decisions}} \tag{3.4.16}$$

The output link flows for Post-routing access may thus be determined from

$$F^u_{kl} = \sum_{i=0}^{NM-1}\sum_{j=0}^{NM-1}\sum_{m=0}^{NM-1}\sum_{n=0}^{NM-1} P_r(R_l=u/i,j,m,n)P_{input_k}(i,j,m,n) \tag{3.4.17}$$

3.4.2 Pre-Routing Access

With pre-routing access, the input to the routing decision does not include new packets entering at the node. The equation equivalent to (3.4.12) becomes

$$P^{u}_{input_u} = \begin{cases} F^{u}_{\Phi(k,l)} & u \neq k \\ 1 - \sum_{i=0}^{NM-1} F^{i}_{\Phi(k,l)} + F^{k}_{\Phi(k,l)} & u = k \end{cases} \tag{3.4.18}$$

Equation (3.4.13) remains the same. The probability of new packets entering at a node must be considered differently for pre-routing access. Start by dividing the arrival process to node k into an arrival process for each queue at node k. Define λ^{u}_{kl} as the mean arrival rate of packets to queue l at node k destined for node u. The total throughput of the queue may be written as

$$\lambda_{kl} = \sum_{u=0}^{NM-1} \lambda^{u}_{kl} \tag{3.4.19}$$

From the previous iteration of the analytic model, the output flow on link l is known. We may therefore define the output flow seen by the new packet queue on link l in terms of the total output flow on the link, corrected for the throughput of the new packet queue on the same link. This defines the mean number of slots available for new packets to enter the network, and can be used to constrain the throughput of arrivals at the queue in question. Define $P^{u}_{E_u}$ as the probability of a new packet

entering the network at node k on link l destined for node u, given that there is an empty slot.

$$P_{E_u}^{u} = \frac{\lambda_{kl}^{u}}{1 - \sum_{i=0}^{NM-1} F_{kl}^{i} + \lambda_{kl}} \tag{3.4.20}$$

The routing decision itself is independent of the access algorithm, thus $P_r(R_l=u/i,j,m,n)$ is as defined in (3.4.16). The access algorithm only affects the packets that are presented to the routing decision. There are two terms in the summation on the righthand side of (3.4.8). The first term applies to the output flows from the routing decision. The second term applies to the throughput associated with new packets entering the network after the routing decision. By noting this fact, we may write the pre-routing access equivalent to (3.4.17) as

$$F_{kl}^{u} = \sum_{i=0}^{NM-1} \sum_{j=0}^{NM-1} \sum_{m=0}^{NM-1} \sum_{n=0}^{NM-1} P_r(R_l = u/i,j,m,n) P_{input_l}(i,j,m,n) + \lambda_{kl}^{u} \tag{3.4.21}$$

3.4.3 Delay Analysis

We are considering a network where the links connecting nodes are assumed to be one slot long, although generalization to different length links is straight forward. In this case, the sum of the elemental link flows gives the mean utilization of a link, and the sum of these utilizations gives the mean number of packets expected to be in the network ($\overline{N}$). Assume that the mean arrival rate of packets to the network is given by λ_T. From application of Little's theorem, the mean transit delay ($\overline{D}$) experienced on the network can be found as follows

$$\overline{N} = \sum_{k=0}^{NM-1} \sum_{l=0}^{l_k} \sum_{i=0}^{NM-1} F_{kl}^{i} \tag{3.4.22}$$

$$\overline{D} = \frac{\overline{N}}{\lambda_T} \tag{3.4.23}$$

The analytical node delay model used to calculate access delays to the network requires, as input to the model, the mean probability, q_{kl}, that a link at a node is available for new packets to enter. In the case of pre-routing access this probability is defined as

$$q_{kl} = 1 - \sum_{i=0}^{NM-1} F_{kl}^{i} + \lambda_{kl} \tag{3.4.24}$$

For post-routing, we have

$$q_{kl} = 1 - \sum_{i=0}^{NM-1} F_{\Phi(k,l)} + F^{k}_{\Phi(k,l)} \qquad (3.4.25)$$

It is also possible to approximate the distribution of transit delay on the network using the elemental link flows and signal flow graph techniques [Zade63]. The technique involves constructing a signal flow graph based upon the flows out of each node, starting with a source node and destined for a single destination. If each such link flow is weighted by z^{-1}, to represent a unit delay, Mason's rule may be applied to determine the z transform (G(z)) of the transit delay for packets between that source node and the destination node. From this, the distribution of delay may be determined by inverting the z transform. The inversion process would involve evaluating G(z) at uniformly spaced points on the unit circle $|z|=1.0$ and performing an inverse discrete Fourier transform to arrive at the probability distribution. Although results of this process are not included in this paper, it was found that the analytical model, when compared with simulation results, produced only moderately good results. The process was performed on traffic between node pairs and was used as a preliminary investigation into the problem of misordering packets in a deflection routing network.

3.4.4 Results

In order to investigate the performance of the analytic model, the PDS, ROS and RDR algorithms were selected since they offer a good cross-section of the possible mix for the three components of node processing. The results consist of a comparisons between exact simulations and the analytical model. In the simulations, it was assumed that the arrival of slots to the system was Poisson with uniformly distributed destinations, and that the input buffering was infinite. Each point of a simulated curve corresponds to the mean value of multiple runs, such that the confidence intervals were 95%, except when the system approached capacity. In the figures which follow, the link availability at each topologically unique node, as well as the overall system transit delay, are compared. The link availability is needed for input to the node delay model of the next section. The combination of the transit delay of this section, and the access delay of the next section, provides a complete delay model for the algorithms considered.

In Figure 3.4.1, Figure 3.4.2 and Figure 3.4.3, since we are considering post-routing access, the total link availability at a node is considered. In this case, we introduce the parameter α_n, which will be used in the node delay model and is defined as the mean probability that there are n links available at a node in a slot. In each figure, we plot the link availability against the total applied load to the network for each topologically unique node in a 9-node network. It is also clear from the graphs that the proposed algorithm yields analytic results that compare very closely to the actual values. Note, however, that the worst performance is at the corner node. The

modelling inaccuracy is attributed to the spatial independence assumption employed to calculate the joint event probabilities at the node. This will be discussed further, when considering the node delay model in the next section.

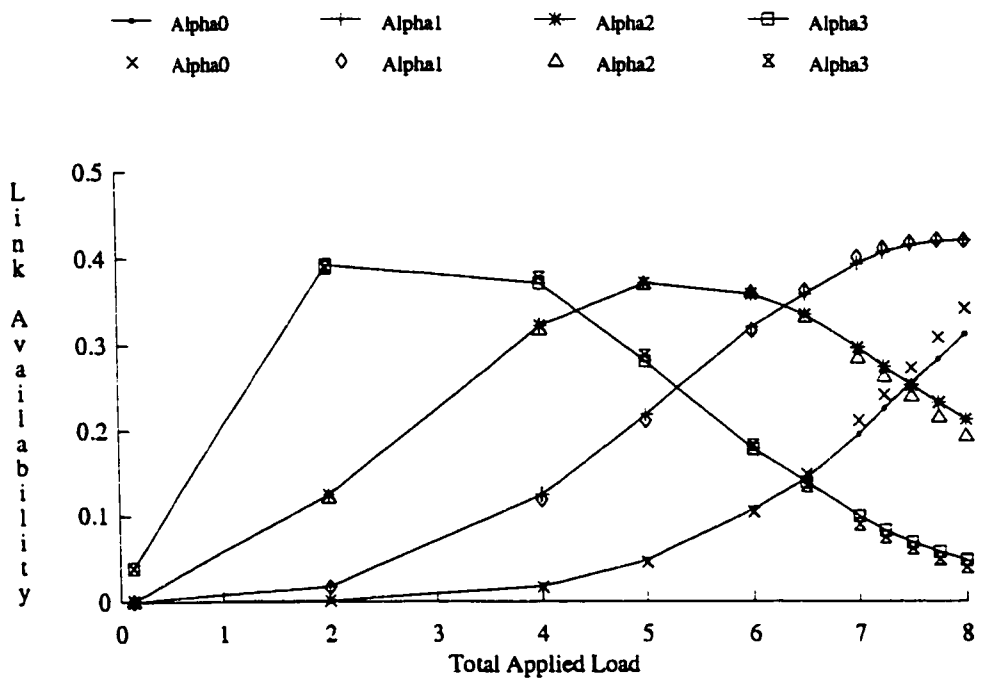


Figure 3.4.1 Flow Rate for Node (1,1)

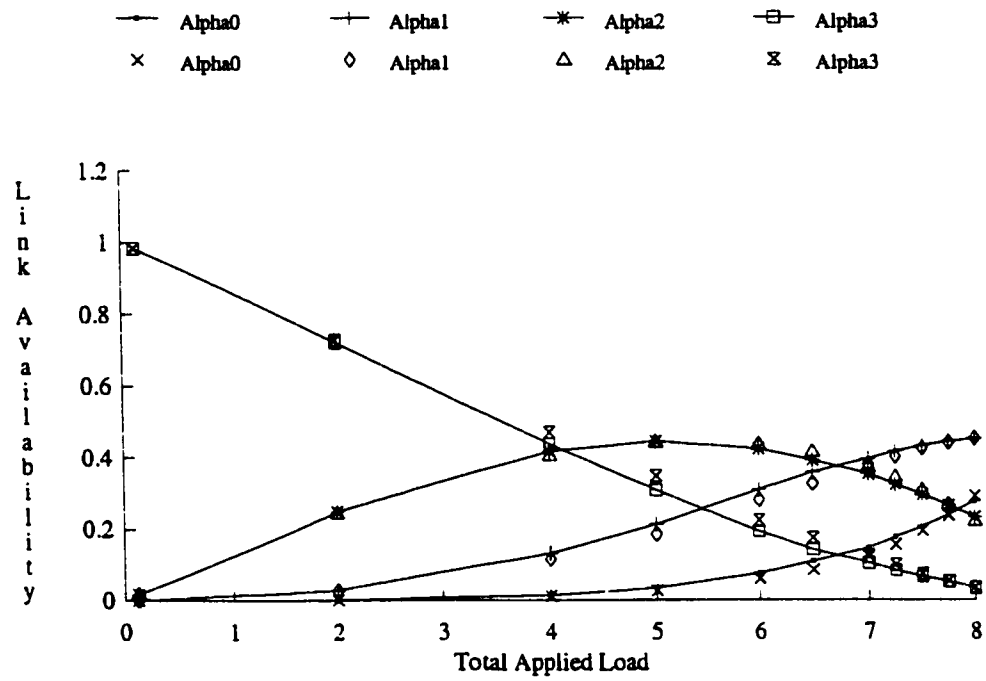


Figure 3.4.2 Flow Rate for Node (1,0)

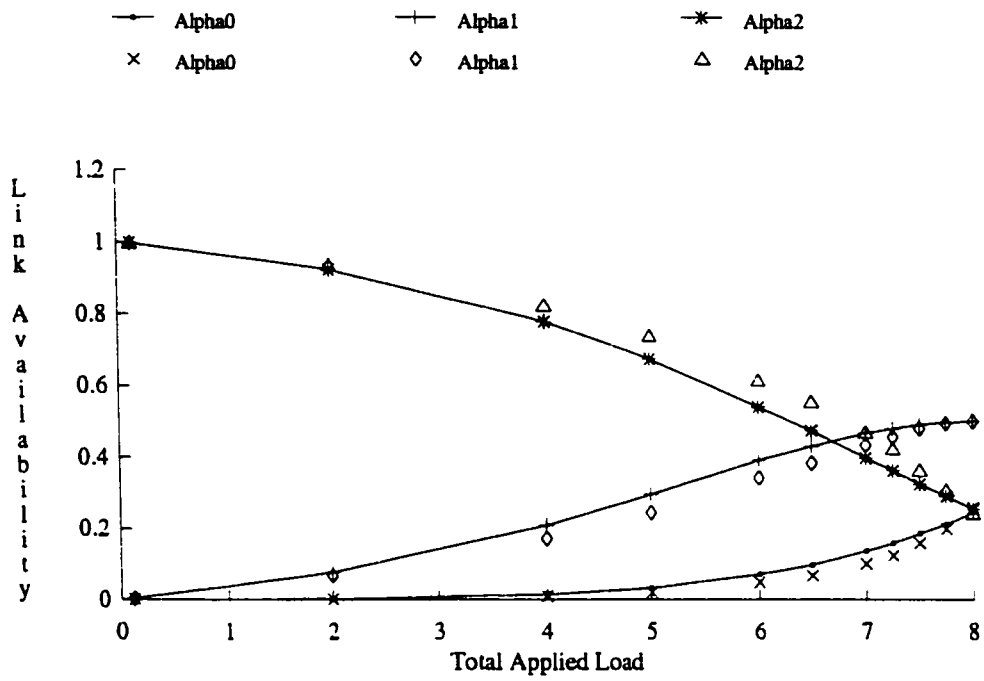


Figure 3.4.3 Flow Rate for Node (0,0)

A critical parameter in the model is the calculation of the new packet entry probability at each node since, unlike other proposed systems [Maxe87a, Borg87], this parameter varies from node to node in PMNet. Figure 3.4.4 is a comparison of calculated values for this probability, with the simulated values for the same network as in Figure 3.4.1, Figure 3.4.2 and Figure 3.4.3. Note the close correspondence of values at all nodes. Once again, the worst results occur at the corner node for the reasons described above.

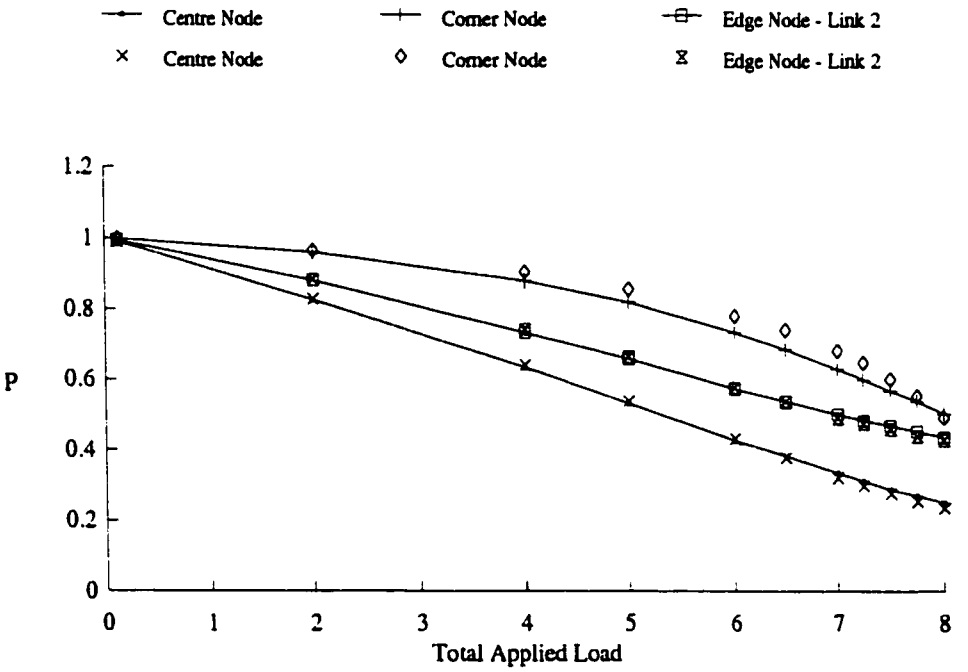


Figure 3.4.4 Probability (P) of New Traffic Entering

Figure 3.4.5 is a comparison of the overall mean transit delay for the 9-node PDS network. It is seen that the analytic model slightly underestimates the mean delay. This is attributed to the performance of the algorithm at the corner nodes, and to the fact that in a 9-node network, the corner nodes contribute significantly to the overall system delay. Also included in Figure 3.4.5, are three curves related to the bandwidth use in the system. In a 3*3 network, there are a total of 24 links interconnecting the nodes. Given that the link length is normalized to the slot length, there may be a maximum of 24 packets in the network at any one time. By use of Little's theorem, we can determine the total number of packets in the network normalized to the maximum number of packets that the network can support. We call this parameter n_t in Figure 3.4.5. Since the mean path length in the network is 2, we can also define the number of packets that would be in the system if all of them followed a shortest path(n_s). The difference between the total number in the system and the number that would be present if shortest paths were followed, can be attributed to deflections and shall be denoted as n_d. $n_d$may be thought of as the amount of bandwidth sacrificed in order to carry deflected traffic. This is, of course, a trade-off against the fact that there are no buffers in the switches. n_d will be used as a point of comparison with the other algorithms.

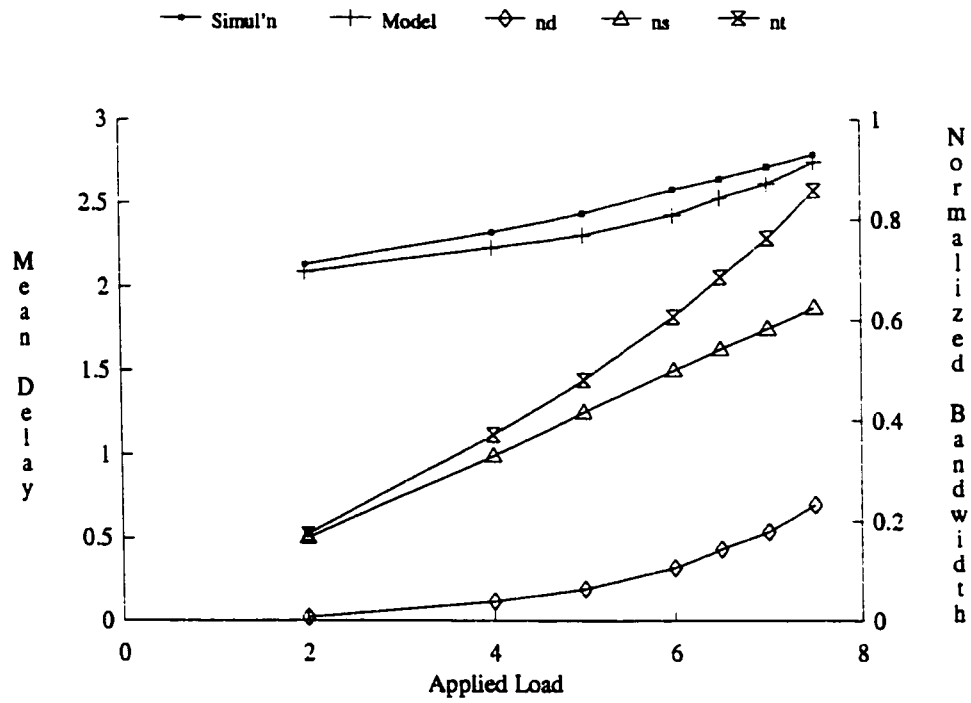


Figure 3.4.5 Mean Network Delay

Figure 3.4.6, Figure 3.4.7, Figure 3.4.8 and Figure 3.4.9 illustrate the model performance for a 9-node ROS algorithm. In this case, due to orthogonal routing, there are four topologically unique nodes. Note that the performance of the model is very good at all nodes, and is noticeably better than the PDS algorithm, especially at the corner nodes. This is due largely to the fact that, with pre-routing access, there is no possibility of deflection upon entry to the network, since new packets queue up at their desired output link and wait for an empty slot. With post-routing access, the transit traffic conditions at corner nodes are affected by the state of the input queues at adjacent nodes, and outgoing links from the adjacent nodes will always be used if the new packet queues are not empty. This introduces a correlation effect in the transit traffic distributions seen at the corner node, which is violated by the independence assumptions of the model. The degree of violation is dependent upon the traffic mixing present at the adjacent nodes, and thus the model performs better for edge and centre nodes. With pre-routing access, the new packets queue for entry into the network in accordance with the distribution of destinations relative to the node. This reduces the above correlation effect and, consequently, the model performs better for pre-routing access.

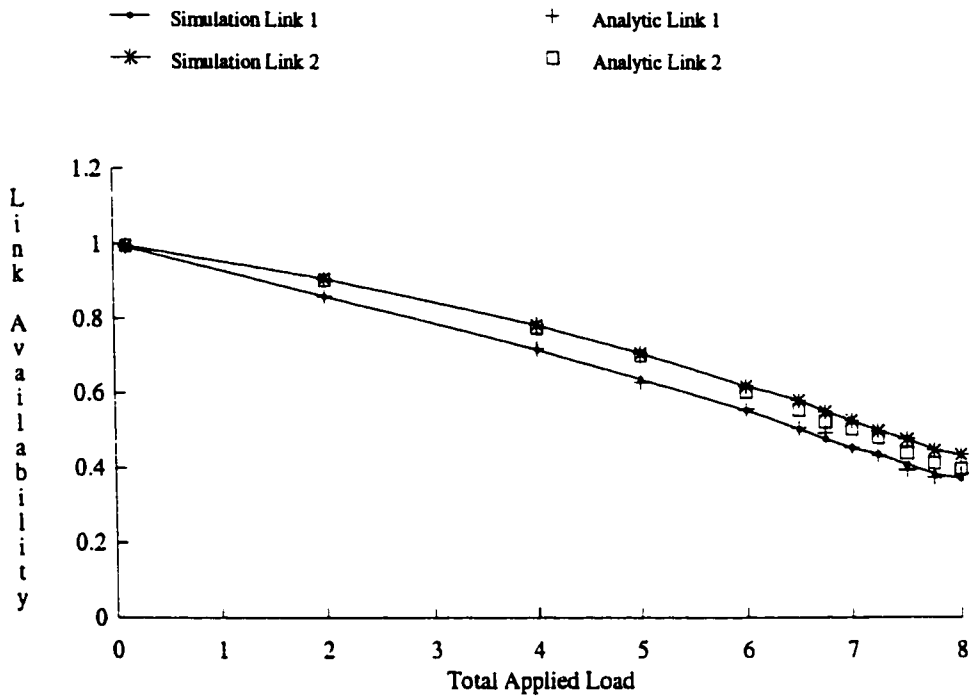


Figure 3.4.6 Flow Rate at Centre Node

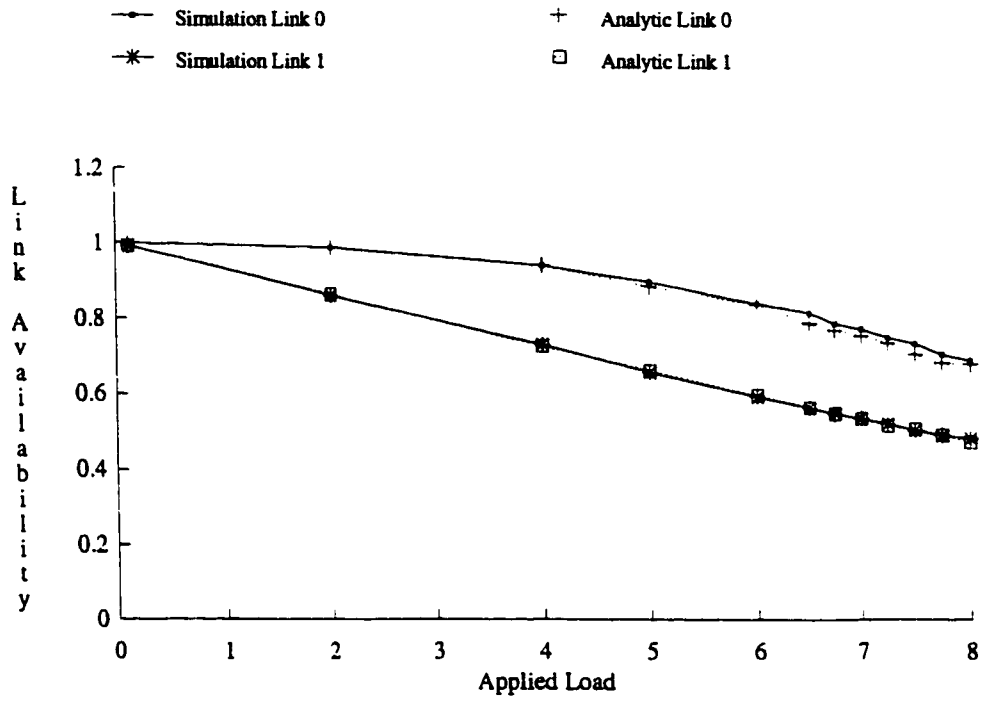


Figure 3.4.7 Flow Rate at Edge Node (0,1)

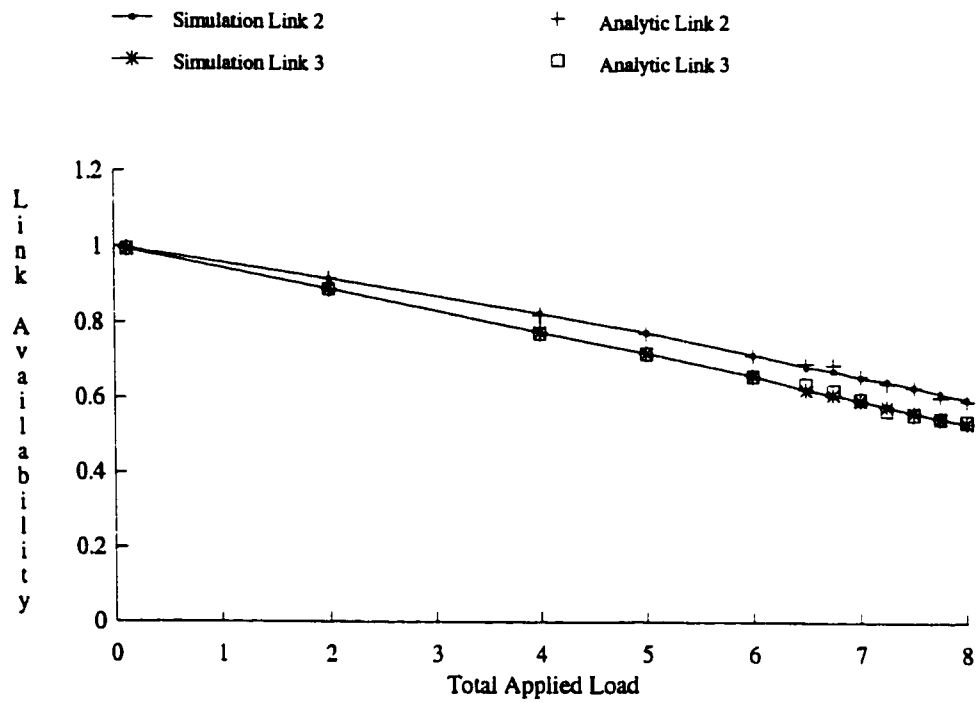


Figure 3.4.8 Flow Rate at Edge Node (2,3)

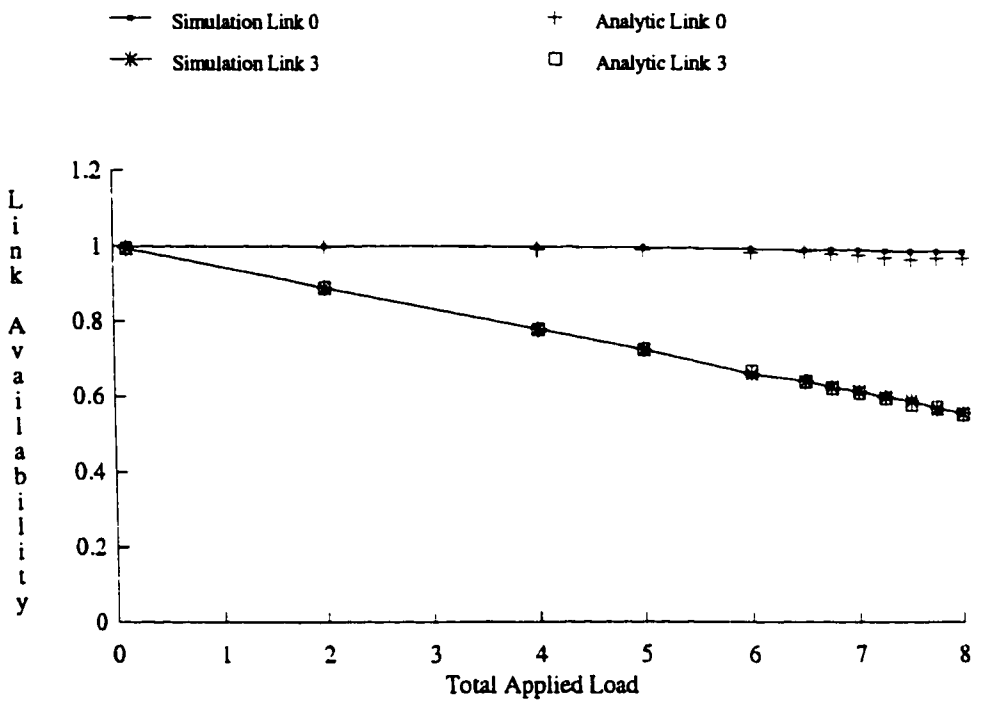


Figure 3.4.9 Flow Rate at Corner Node

The improved performance is also seen in the delay calculation of Figure 3.4.10, where the analytical delay matches very closely the simulated delay. In addition, note that the deflection delay performance of the ROS algorithm differs from that of the PDS algorithm. The sacrificed bandwidth with pre-routing access is less than with post-routing access, which is to be expected. It is difficult, however, to draw any conclusions from this, since we have seen that the trade-off of additional deflection delay in post-routing access systems of a larger size serves to significantly increase the system capacity by reducing the access delays. It is also important to notice that the actual mean transit delay in the systems considered do not vary widely over the entire operating range of the network.

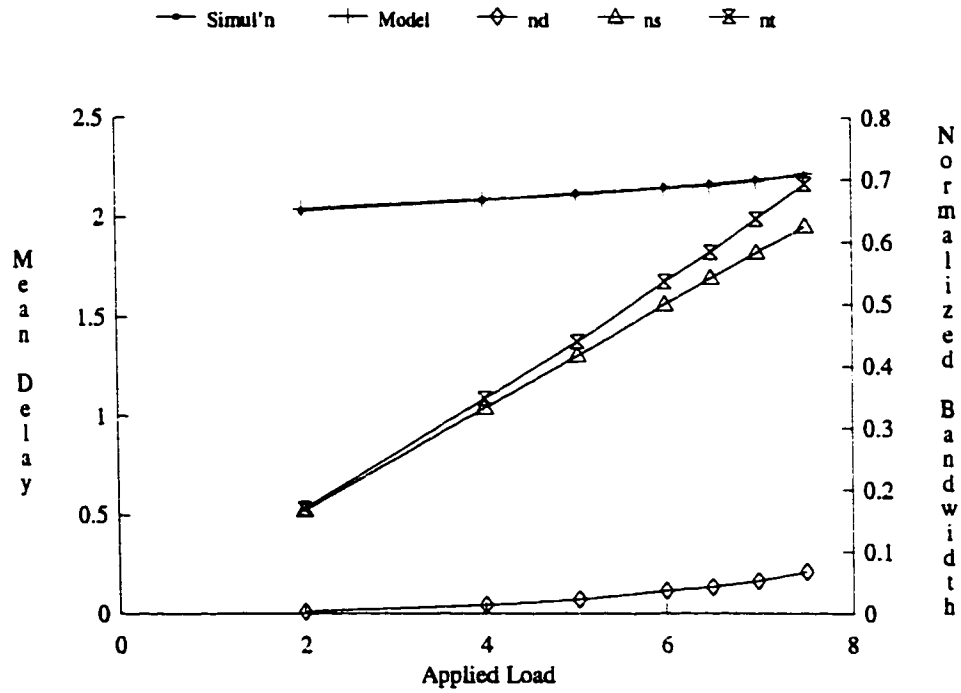


Figure 3.4.10 Mean Network Delay

The results for the RDR algorithm are included in Figure 3.4.11 and Figure 3.4.12, for the sake of completeness. As with ROS, the performance of the model is very good over the entire range. Similar correspondence between the model and simulations have been found for 16-node RDR networks.

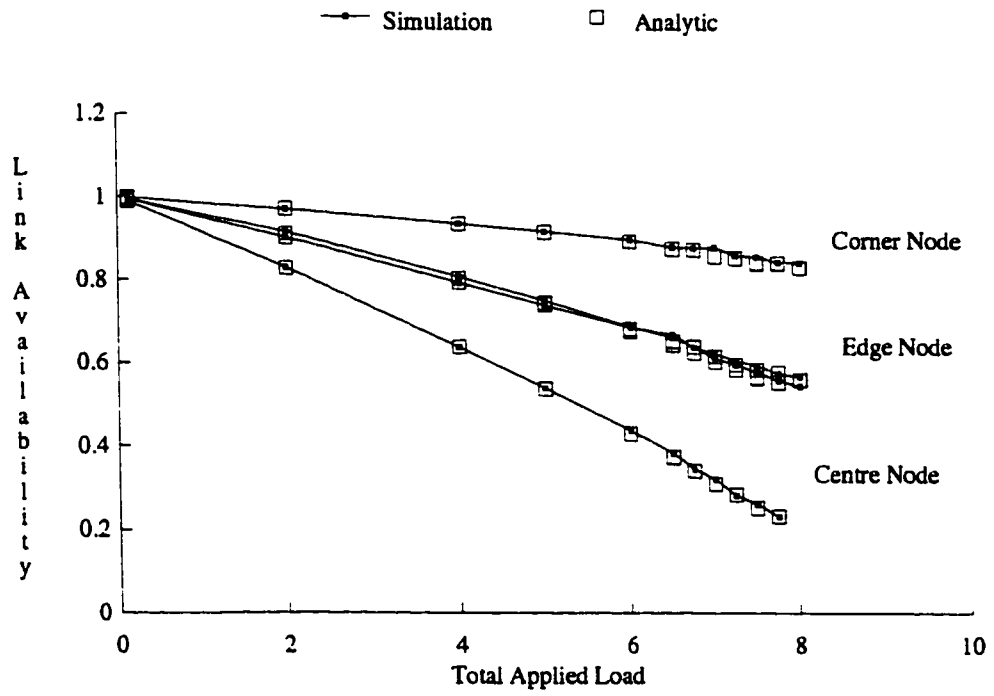


Figure 3.4.11 Flow Rate at All Nodes

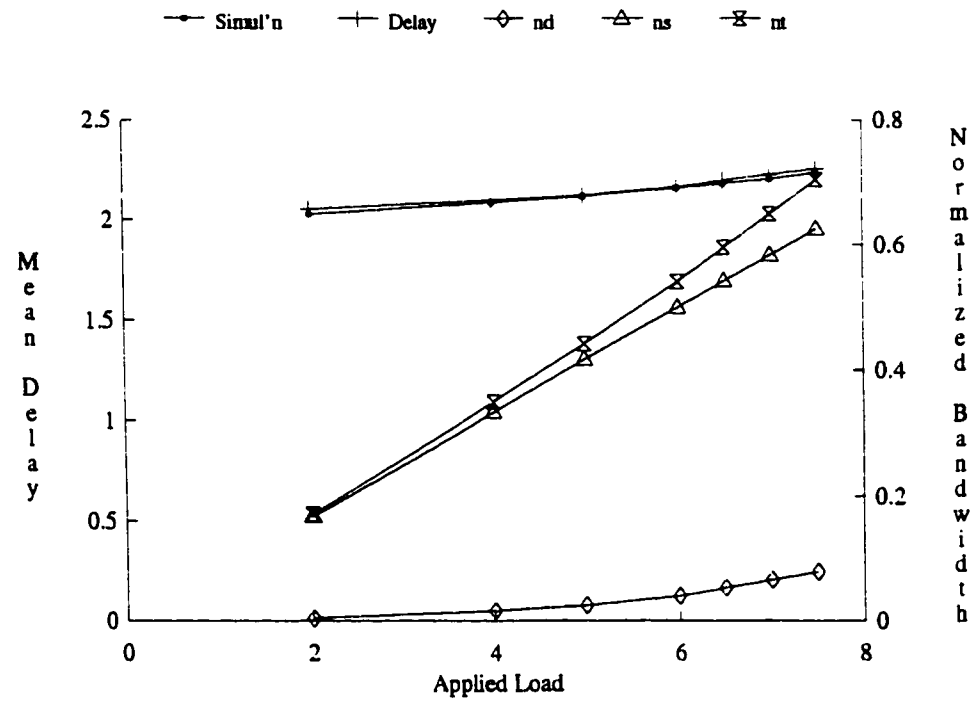


Figure 3.4.12 Mean Network Delay

3.5 An Analytic Node Delay Model

In this section, we focus our attention on the development of a single node delay model, which may be used to predict mean nodal delay performance. The model formulation also allows for investigating the use of link independence assumptions in a regular structure that utilizes deflection routing. As its inputs, the model uses the detailed link flow rates derived in the previous section. Individual node statistics are computed by assuming that the node system is independent of the transit packet arrival process for each input link. Figure 3.5.1 illustrates the situation considered.

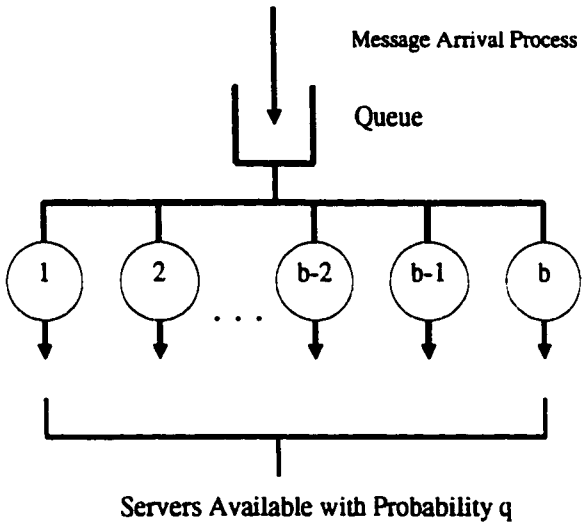


Figure 3.5.1 The Queuing Model

In this case, we are dealing with Post-Routing Access, and thus a single queue is formed for the input traffic. Later, we will show that pre-routing access is a special case of post-routing access derivation. Assuming N_L input and output links, the service rate of the queue is random, and dependent upon the flow and distribution of

transit traffic passing through the node in question. The formulation given is a generalization of the queuing model originally presented in [Bail54]. The following definitions apply in the derivation.

n_i	Number of enqueued packets at the end of slot i
d_i	Number of packets that depart the queue in slot i
a_i	Number of packets that arrive at the queue in slot i
N_L	Number of outgoing links
q_j	Probability that link j is available in a slot
α_n	Probability that there are only n links available in a slot
β_n	Probability of at least n links available in a slot
p_k	Probability of k packets enqueued

An embedded Markov chain model is developed, consisting of a single input queue with b outgoing links, which are available with probability q_j. The system is slotted, and fixed length packets arrive to the queue at a Poisson rate λ. The state equation for the queue may be written at the embedding points as

$$n_{i+1} = n_i - d_{i+1} + a_{i+1} \tag{3.5.1}$$

Assuming that the probabilities of individual link availabilities are independent, we may define the availability of the servers to the queue. Examples for the case where N_L=4 are given below.

$$\alpha_0 = \prod_{i=0}^{3} (1-q_i) \tag{3.5.2}$$

$$\alpha_1 = \sum_{i=1}^{3} q_i \sum_{\substack{j=0 \\ j \neq i}}^{3} (1-q_j) \tag{3.5.3}$$

$$\alpha_2 = \sum_{i=0}^{3} q_i \sum_{j=i+1}^{3} [q_j \prod_{\substack{k=0 \\ k \neq i \\ k \neq j}}^{3} (1-q_k)] \tag{3.5.4}$$

$$\alpha_3 = \sum_{i=0}^{3} [\ (1-q_i) \prod_{\substack{j=0 \\ j \neq i}}^{3} q_j\ ] \tag{3.5.5}$$

$$\alpha_4 = \prod_{i=0}^{3} q_i \tag{3.5.6}$$

In a similar fashion, expressions may be written for the probabilities β_i. The probability generating function, N(z), of (3.5.1) is found in the usual fashion using expectations.

$$E[z^{n_{i+1}}] = E[z^{n_i - d_{i+1}}]E[a^{i+1}] \tag{3.5.7}$$

When evaluating $E[z^{n_i-d_{i+1}}]$, it is necessary to find the conditional probabilities of departures from the queue, given the state of the queue ($Pr[d=j \mid n_i=i]$). Using the probabilities defined above, and manipulating the expressions, we get

$$\begin{aligned} E[z^{n_i-d_{i+1}}] &= \sum_{i=0}^{\infty}\sum_{j=0}^{N_L} z^{i-j} Pr[d_{i+1}=j/n_i=i]P_i \\ &= N(z)\sum_{j=0}^{N_L}\alpha_j z^{-j} - \sum_{i=1}^{N_L-1} z^{i} P_i\{\sum_{j=i}^{N_L}\alpha_j z^{-j}\} \\ &\quad - P_0\{\sum_{j=0}^{N_L}\alpha_j z^{-j} - 1\} + \sum_{i=1}^{N_L-1}\beta_i P_i \end{aligned} \tag{3.5.8}$$

By noting that $E[z^{a_{i+1}}] = e^{-\lambda(1-z)}$, substituting with (3.5.8) into (3.5.7), isolating for N(z) and rearranging, we arrive at the required probability generating function.

$$N(z) = \frac{e^{-\lambda(1-z)}[\sum_{i=1}^{N_L-1}\beta_i P_i z^{N_L} - \sum_{i=1}^{N_L-1}\sum_{j=1}^{N_L}\alpha_j z^{i-j+N_L} P_i - P_0(\sum_{j=0}^{N_L}\alpha_j z^{N_L-j} - z^{N_L})]}{z^{N_L} - e^{-\lambda(1-z)}\sum_{j=0}^{N_L}\alpha_j z^{N_L-j}} \tag{3.5.9}$$

In order to evaluate the above expression, we need the boundary condition probabilities P_j for $i \in \{0,1, ..., N_L-1\}$. We start by considering the normalizing

condition of the probability generating function in (3.5.9). In general, this condition may be stated as

$$N(z)\ |_{z=1} = \sum_{0}^{\infty} z^{i} P(n=i)\ |_{z=1} = \sum_{0}^{\infty} P(n=i) = 1 \qquad (3.5.10)$$

This condition simply states that the numerator of (3.5.9) must equal the denominator of (3.5.9) when evaluated at z=1.0. Substituting z=1.0 into (3.5.9), however, forces us to apply l'Hopitals rule. By differentiating both numerator and denominator one time and substituting z=1.0, we arrive at one equation in N_L unknowns.

$$[N_L - \sum_{j=0}^{N_L} (N_L - j)\alpha_j]P_0 + \sum_{j=1}^{N_L - 1} \sum_{k=j}^{N_L} (k-j)\alpha_k P_j = N_L - \lambda - \sum_{j=0}^{N_L} (N_L - j)\alpha_j \qquad (3.5.11)$$

The remaining equations are determined through the invocation of Rouche's theorem.

In the application of Rouche's theorem, consider the denominator of (3.5.9) and define

$$f(z) = z^{N_L} \qquad (3.5.12)$$

$$g(z) = e^{-\lambda(1-z)} \sum_{j=0}^{N_L} \alpha_j z^{b-j} \qquad (3.5.13)$$

Rouche's theorem states that, given f(z) and g(z) are analytic in a region **R**, and given that on a contour **C** in region **R** $f(z) \neq 0$ and $|\ f(z)\ | > |\ g(z)\ |$, then f(z) and $f(z) + g(z)$ have the same number of zeroes within **C**. In the case defined above, both f(z) and g(z) are analytic for $|\ z\ | \leq 1.0$. We now consider a region that is

slightly larger than the unit circle **R**: $|z| \leq 1+\delta$. If δ is small, then f(z) and g(z) are also analytic in **R**. We now define a contour **C**: $|z| < 1+\delta' < 1+\delta$ and write expressions for f(z) and g(z).

$$| f(z) | = 1 + N_L\delta' \tag{3.5.14}$$

$$| g(z) | = 1 + \delta'[\lambda + \sum_{j=0}^{N_L-1} \alpha_j(N_L - j)] \tag{3.5.15}$$

In order to apply Rouche's theorem, we require that $| f(z) | < | g(z) |$ evaluated on **C**. This translates, after manipulation, directly into the condition that

$$\lambda < N_L - \sum_{j=0}^{N_L-1} \alpha_j(N_L - j) \tag{3.5.16}$$

Inspection of (3.5.16) indicates that it is the necessary condition for a stable queue, since the arrival rate must, at all times, be less than the average number of available lines. Since, in the region of interest, the condition of (3.5.16) must hold, then $| f(z) | < | g(z) |$ on **C** and Rouche's theorem may be applied. Since f(z) has N_L zeroes within **C**, then $F(z) = g(z)$ also has N_L zeroes within **C**. Therefore, when evaluated at zeroes z_i, we may write

$$z_i^{\,b} = e^{-\lambda(1-z_i)}[\sum_{j=0}^{N_L} \alpha_j z_i^{\,b-j}] \tag{3.5.17}$$

Note that if any of the roots z_i are multiple roots, the derivatives of both sides of (3.5.17) will be zero since a single derivative will not remove a pole $(z-z_i)^j$ for j>1.

By differentiating both sides of (3.5.17), it can easily be shown that the case for multiple poles requires that the stability condition of the queue be violated. Therefore, all poles of N(z) are simple. Since we know that N(z) is bounded on the unit disk, then the zeroes of the denominator of N(z) must be cancelled by zeroes of the numerator of N(z). An equation for the case of the pole at z=1.0 was given in (3.5.11). By numerically determining the remaining N_L-1 roots of the denominator function, we can substitute them into the numerator function and set it equal to zero, in order to obtain the remaining N_L-1 equations.

$$[z_i^{N_L} - \sum_{j=0}^{N_L} \alpha_j z_i^{N_L - j}]P_0 + \sum_{j=1}^{N_L - 1} [\beta_j z_i^{N_L} - \sum_{k=j}^{N_L} \alpha_k z_i^{j-k+N_L}]P_j = 0 \qquad (3.5.1.8)$$

The set of N_L linear equations may now be solved to determine the boundary condition probabilities P_j for $j \in (0,1, ..., N_L-1)$. The P_{js} are substituted into (3.5.9), in order to completely specify N(z).

The next step in the calculation is to find an expression for access delay, given N(z). A straight-forward application of Little's theorem is not appropriate, since the embedding points of the Markov chain do not sample the steady state distributions of the system state. This portion of the derivation is formulated as shown in Figure 3.5.2. Define n_0 as the number of enqueued packets at the start of an arbitrary slot. Some time, τ, after the start of that slot, a packet arrives. We are interested in calculating the access delay experienced by that packet. At the start of the next slot, the number of packets enqueued in front of our selected packet will be determined by

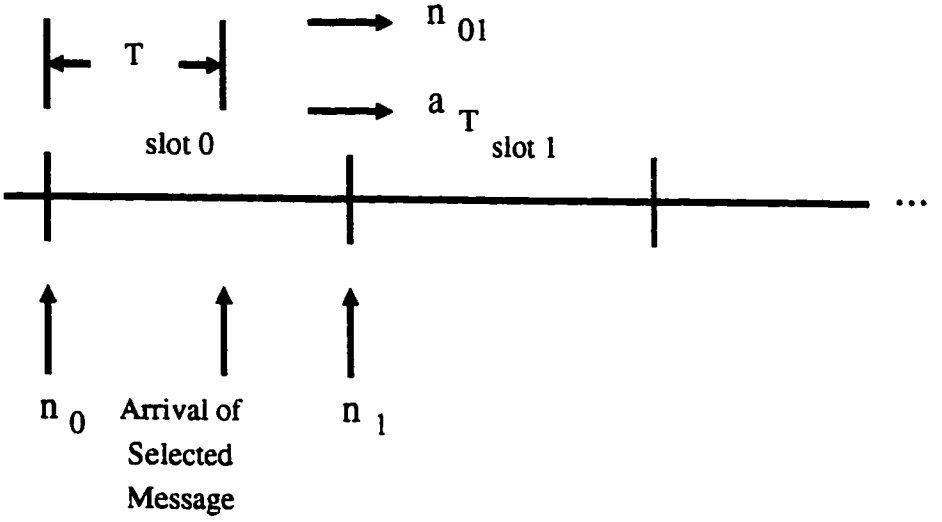


Figure 3.5.2 Delay Model Problem Formulation

the number of packets carrying over from the previous slot (n_{01}) and the number of new packets a_τ arriving in the interval τ.

$$n_1 = n_{01} + a_\tau + 1 \tag{3.5.19}$$

If we define m as the number of slots needed to transmit n_1 packets, then the access delay experienced by our packet is given by the sum of two random variables.

$$d_A = (1-\tau) + (m-1) = m - \tau \tag{3.5.20}$$

Taking expectations, we find the mean access delay via

$$E[d_A] = \overline{d_A} = E[m] - E[\tau] \tag{3.5.21}$$

Since the conditioned Poisson arrival is uniformly distributed across the slot, $E[\tau]$ is given by $\frac{1}{2}$ slot. In order to find E[m], we need the probability density function (pdf) of m.

Returning to (3.5.19), we see that the three random variables on the righthand side are independent. The pdf of n_1 is therefore given by the convolution of the pdf's of the righthand side variables. First, consider $n_{01} = \max\{n_0-d,0\}$, which has a pdf given by

$$p_{n_{01}}(0) = \sum_{i=0}^{N_L} \sum_{j=0}^{i} p_n(j) p_d(N_L-i+j) \tag{3.5.22}$$

$$p_{n_{01}}(k) = \sum_{i=0}^{N_L} p_d(i) P_n(i+k) \quad k>0 \tag{3.5.23}$$

Note that the max function in n_{01} serves to cause an accumulation of probability at the origin in (3.5.22). The pdf of a_τ is found by first conditioning on τ, where

$$p_{a_\tau}(k \mid \tau) = \frac{(\lambda\tau)^k e^{-\lambda\tau}}{k!} \tag{3.5.24}$$

After removing the condition and considerable manipulation, we arrive at

$$p_{a_\tau}(k) = \frac{1}{\lambda} - \frac{e^{-\lambda}\lambda^{k-1}}{k!} \left[1 + \sum_{i=1}^{k} \frac{k!}{(k-i)!\ \lambda^i}\right] \tag{3.5.25}$$

We may now write the pdf of n_1 as the following convolution

$$p_{n_1}(n) = \sum_{i=0}^{n-1} p_{n_{01}}(i)\ p_{a_\tau}(n-i-1) \tag{3.5.26}$$

In order to solve this, we need the pdf of d, which is given directly from the previous analysis.

$$p_d(i) = \alpha_i \qquad i \le N_L \tag{3.5.27}$$

In addition, $p_n(i)$ is determined from our expression for N(z), by evaluating N(z) at sufficiently many, say K, equally spaced points around the unit circle in the z plane and performing the inverse discrete Fourier transform, namely

$$p_n(i) = \frac{1}{K} \sum_{k=0}^{K-1} N(e^{j2\pi k/K}) e^{j2\pi ik/K} \tag{3.5.28}$$

The final step in the derivation involves determining the probability of taking i slots to transmit our selected packet. We start by conditioning this probability on the number of packets, n_1, and denoting it as $p_m(i \mid n_1=n) = Pr(m=i \mid n_1=n)$. At the start of the slot immediately in front of the slot in which our selected packet arrived, there are n_1 packets up to and including the selected packet. The number of packet that may leave the queue in a single slot is determined by (3.5.27). The number of packets that may leave in i slots, by applying our original independence assumption, is therefore given by the i-fold convolution of (3.5.27). $p_m(i \mid n_1=n)$ will therefore be found by subtracting the cumulative distribution functions for i and i-1 slots or, in other words, the difference between the i-fold and (i-1)-fold convolutions of (3.5.27).

$$p_m(i \mid n_1=n) = \sum_{j=n}^{\infty} [p_d^{(i)}(j) - p_d^{(i-1)}(j)] \; . \tag{3.5.29}$$

In (3.5.29), $p_d^{(i)}$ denotes the i-fold convolution of p_d. (3.5.26) may be used to remove the condition in (3.5.29) to arrive at $p_m(i)$ and thus determine E[m] as required. The access delay is then directly determined by (3.5.21).

The above model is generalized, in order to include the availability of multiple servers to the queue. In the case of pre-routing access, there is a single server, whose availability is stochastic, and a single queue. This simplifies the analysis. For example, using the notation of the previous analysis, we may write N(z) as

$$N(z) = \frac{\alpha_1 P_0 (z-1) e^{-\lambda(1-z)}}{z(1 - \alpha_0 e^{-\lambda(1-z)}) - \alpha_1 e^{-\lambda(1-z)}} \tag{3.5.30}$$

By definition, N(z)=1.0 when evaluated at z=1.0. This allows us to solve directly for P_0.

$$P_0 = 1 - \frac{\lambda}{\alpha_1} \tag{3.5.31}$$

Substituting (3.5.31) into (3.5.30) completely specifies N(z), and a similar delay cycle analysis may be applied.

3.5.1 Results

As with the flow-rate model, in order to investigate the performance of the node delay model, the PDS, ROS and RDR algorithms were selected. Once again, the analytic model results are compared with the exact simulation results. The curves were generated by varying the arrival rate at a certain node in the network while maintaining a constant background traffic load (BL) at all other nodes in the network.

Figures 3.5.3 to 3.5.5 show the model performance for the PDS algorithm at several nodes in a 9-node network. There are only three topologically unique nodes in the network: the centre node (1,1), the edge node (1,0) and the corner node (0,0). The results are comparable across all nodes and are seen to be good over a wide range of operating conditions.

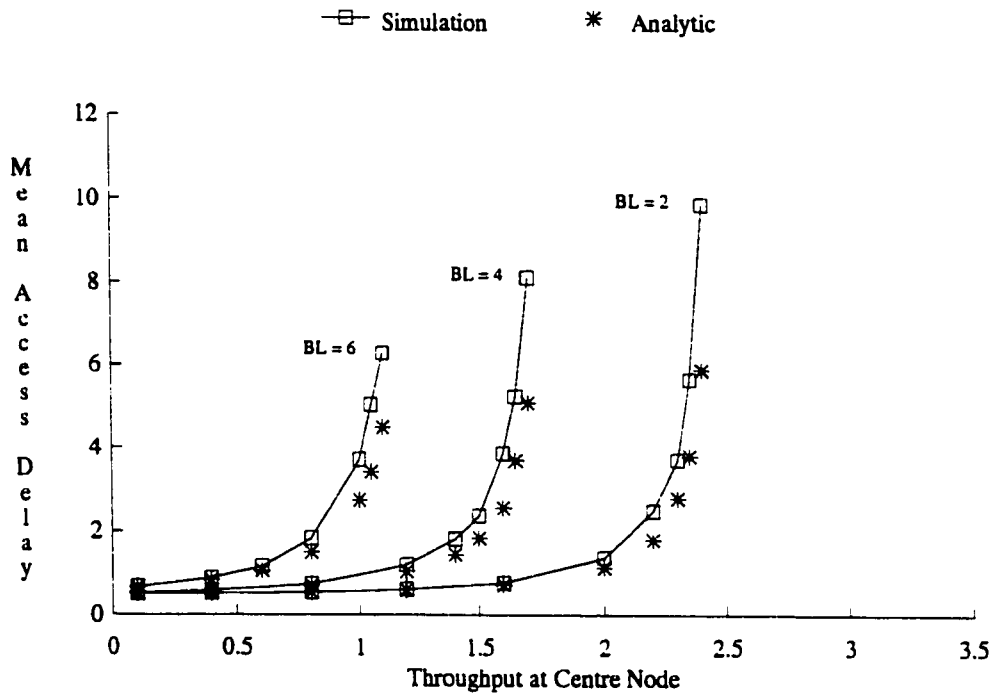


Figure 3.5.3 Access Delay for (1,1) 3*3 PDS Network

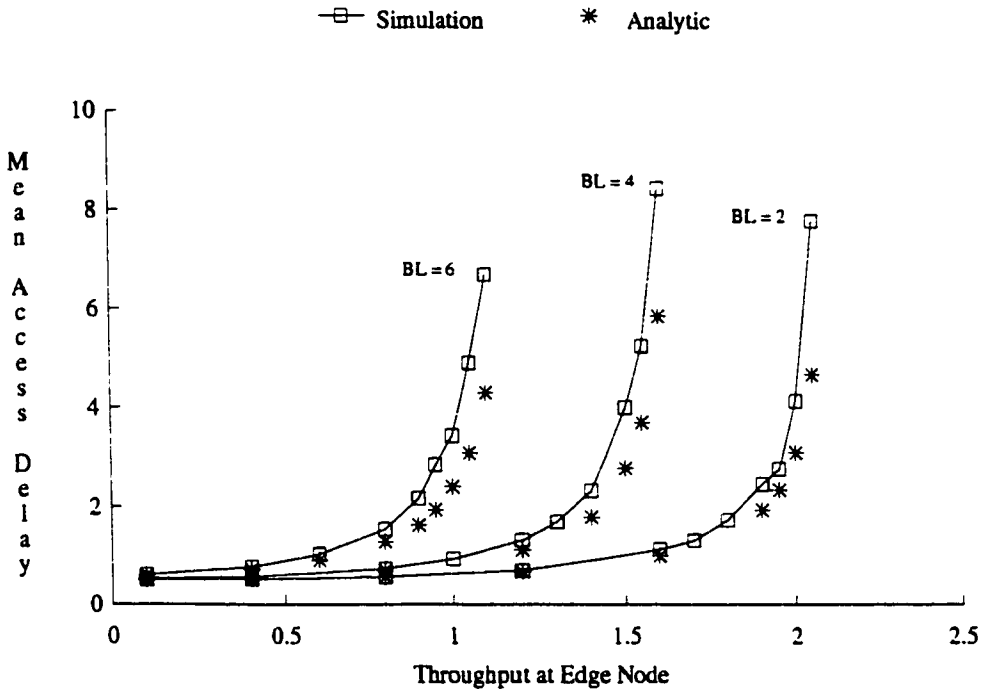


Figure 3.5.4 Access Delay for (1,0) 3*3 PDS Network

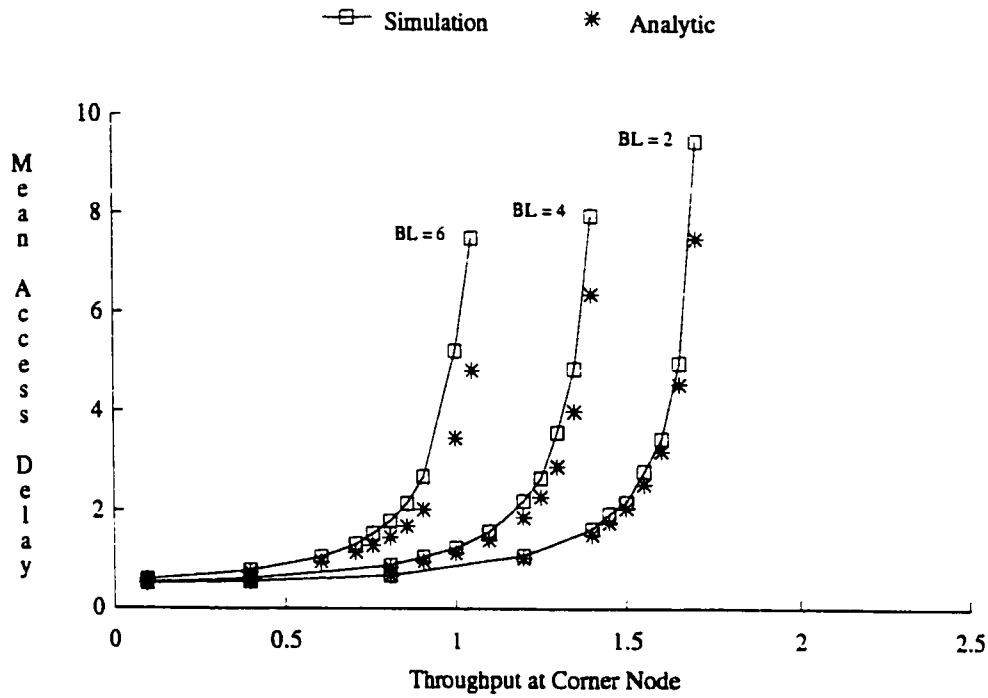


Figure 3.5.5 Access Delay for (0,0) 3*3 PDS Network

Figures 3.5.6 to 3.5.8 show the effective server capacity as a function of throughput at each of the nodes in Figures 3.5.3 to 3.5.5. These curves are included to illustrate the correlation between the local queue state and the transit packet arrival process. With the exception of the difference between the values of server capacities, the relationship with the throughput of the node is similar for all nodes.

Results for larger networks indicated a larger discrepancy between the model performance at various nodes and values of BL. This occurs due to a larger variation in the transit traffic statistics between nodes in a larger network.

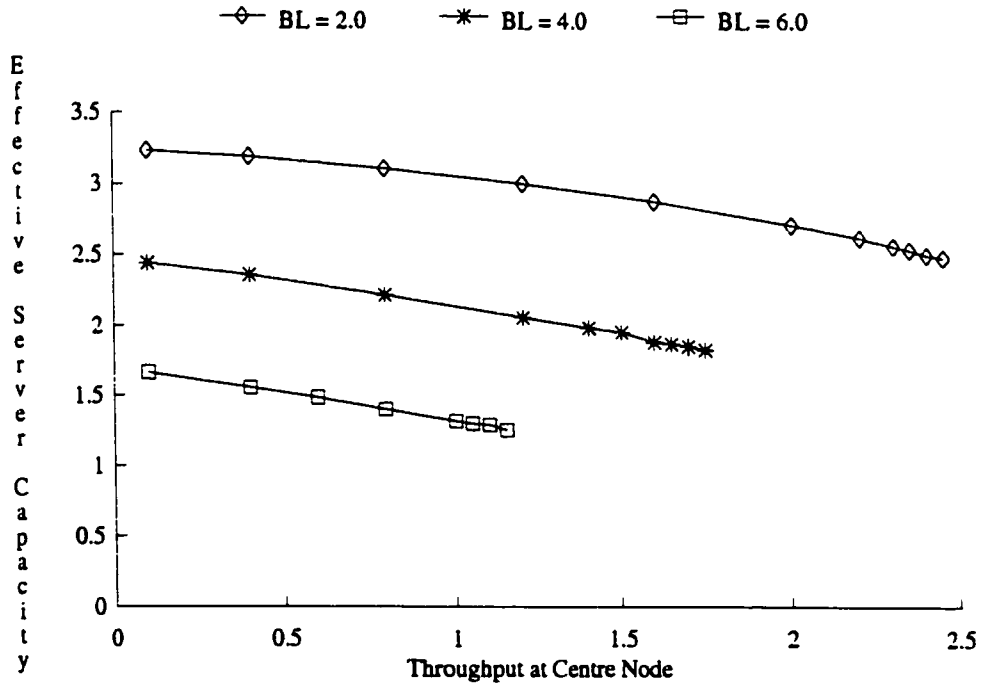


Figure 3.5.6 Server Capacity at (1,1) 3*3 PDS Network

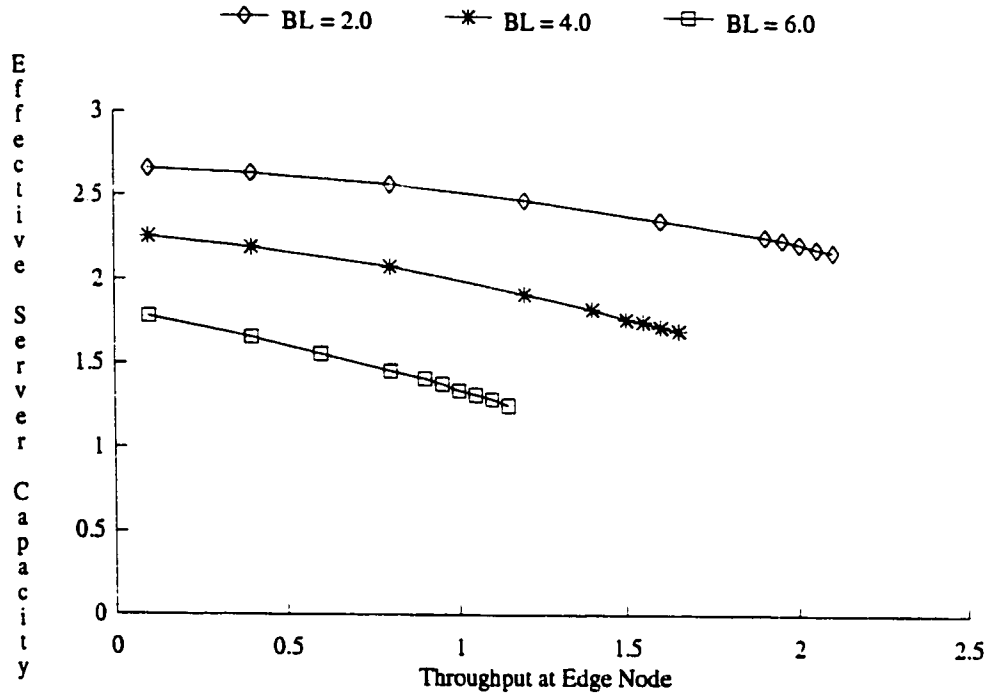


Figure 3.5.7 Server Capacity at (1,0) 3*3 PDS Network

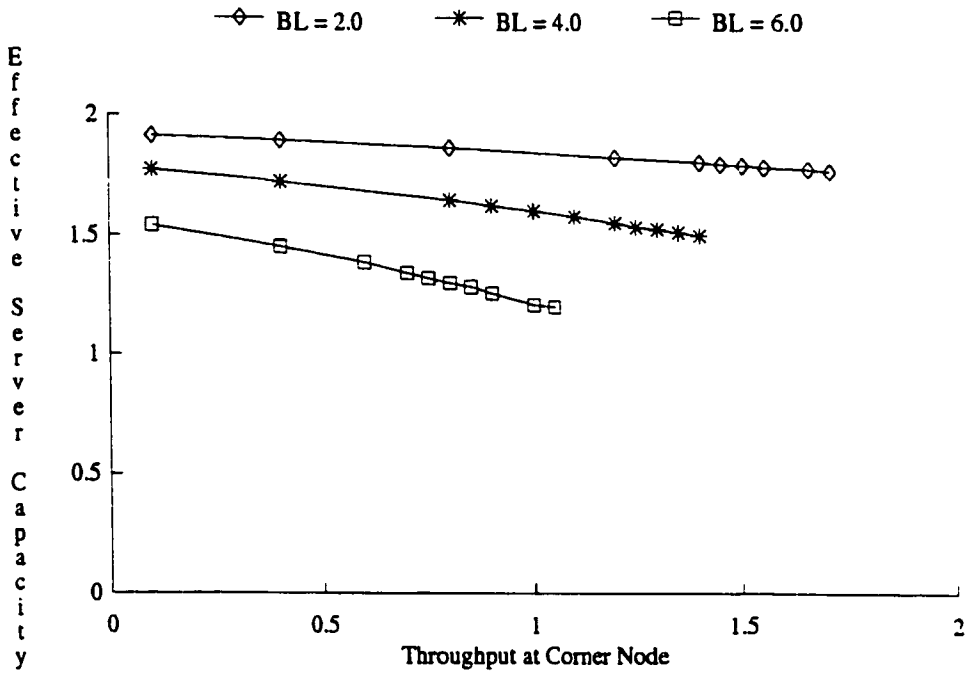


Figure 3.5.8 Server Capacity at (0,0) 3*3 PDS Network

Figures 3.5.9 to 3.5.12 are typical of the model performance for a 25-node network (in this case a 5*5-node PDS network). The nodes were numbered from the bottom lefthand corner (0,0) to the upper righthand corner (4,4). Figure 3.5.9 corresponds to the centre node, while Figures 3.5.10 to 3.5.12 correspond to the (1,1), (2,0) and (0,0) nodes of the same network. As before, the curves were generated by varying the traffic at the node in question while maintaining constant traffic load (BL) at all other nodes in the network. The difference between the simulated and analytic curves represents the effects of the transit link independence assumptions made in the model formulation. It can be seen that for the (2,2) node, the correspondence between simulated and analytic results is quite close, even when the total load on the system is quite large. It should be noted that the analytic model underestimates the true delay in all cases. This is attributed to the spatial and temporal independence assumptions for the transit traffic. In the centre node case, however, this effect is reduced due to traffic mixing caused in part by the deflection process. It can be seen that for the (1,1), (2,0) and (0,0) nodes, the analytic model provides reasonable but increasingly poorer overall performance. These nodes are best divided into two general categories when discussing the performance of the model. Nodes (2,2) and (1,1) do not lie on the topological edges of the network and, due to the PDS algorithm, experience a greater mixing of transit traffic. This mixing, caused in part by the process of deflection, serves to reduce the temporal correlation present in the transit traffic. Since the analytical model is based on the absence of such correlation, it produces reasonably accurate results over a wide range of operating conditions. The mixing is better at node (2,2), and thus the performance is better.

Nodes (2,0) and (0,0) lie along the topological edges of the network and, in general, see less transit traffic and therefore less traffic mixing. The temporal independence assumption is valid for moderate loads due to this reduced traffic level. In fact, comparison of these results with those of nodes (2,2) and (1,1) at light loads show that the model performs better at the edge nodes. The situation changes significantly however when the background load increases. In this case, when the edge node queues are non-empty, transit packets are much more likely to be received due to the post-routing access algorithm at the node in question, as well as at the adjacent nodes. The result is correlation between the local queue state and the transit packet arrival process. The effect is most pronounced at the (0,0) node.

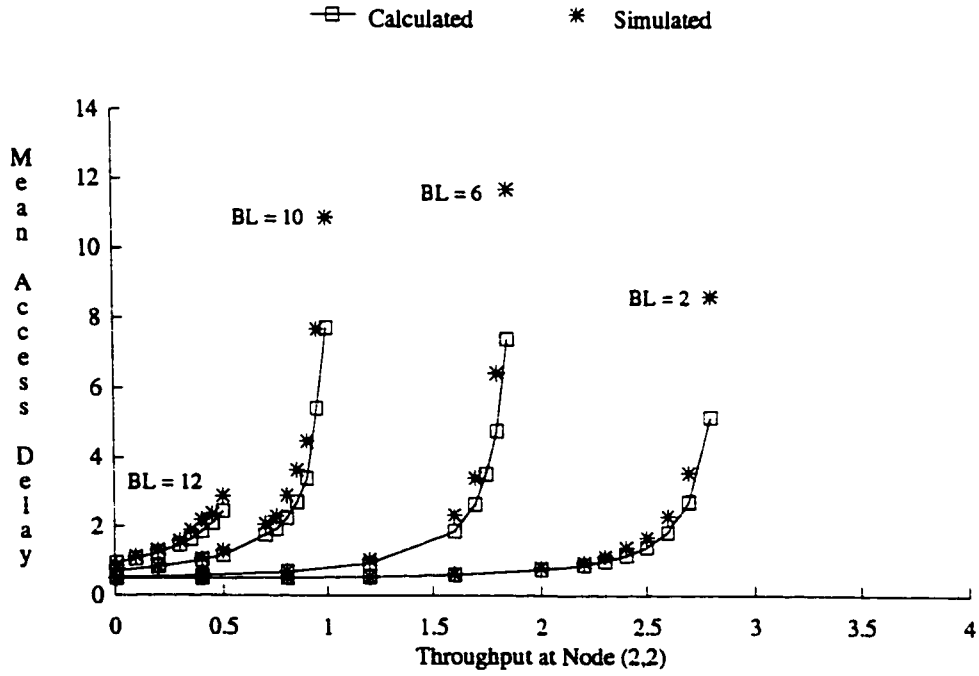


Figure 3.5.9 Access Delay (2,2) 5*5 PDS Network

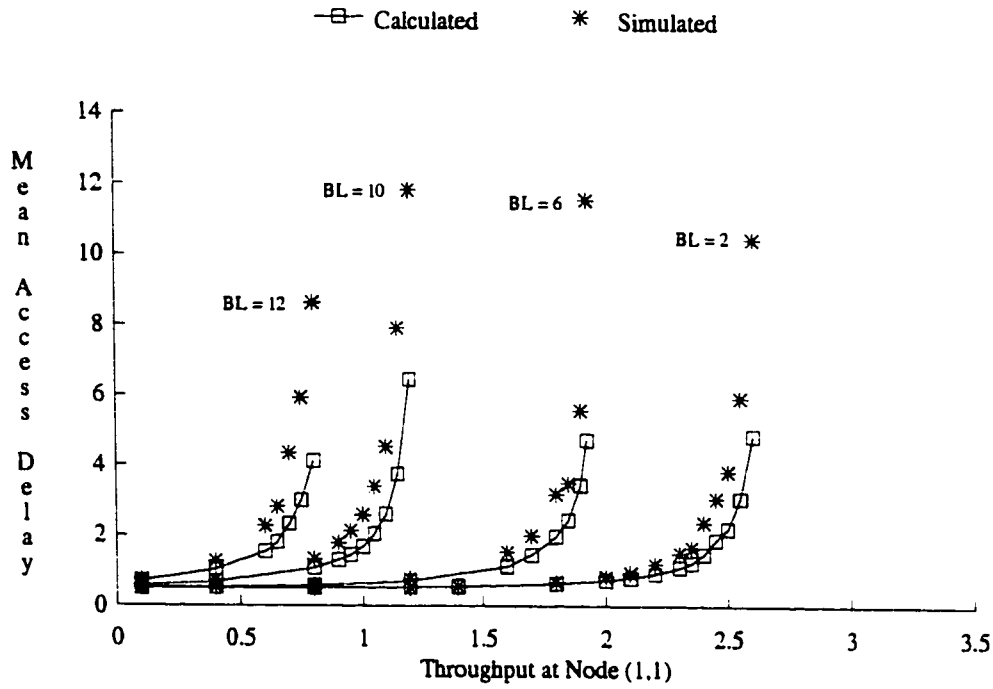


Figure 3.5.10 Access Delay (1,1) 5*5 PDS Network

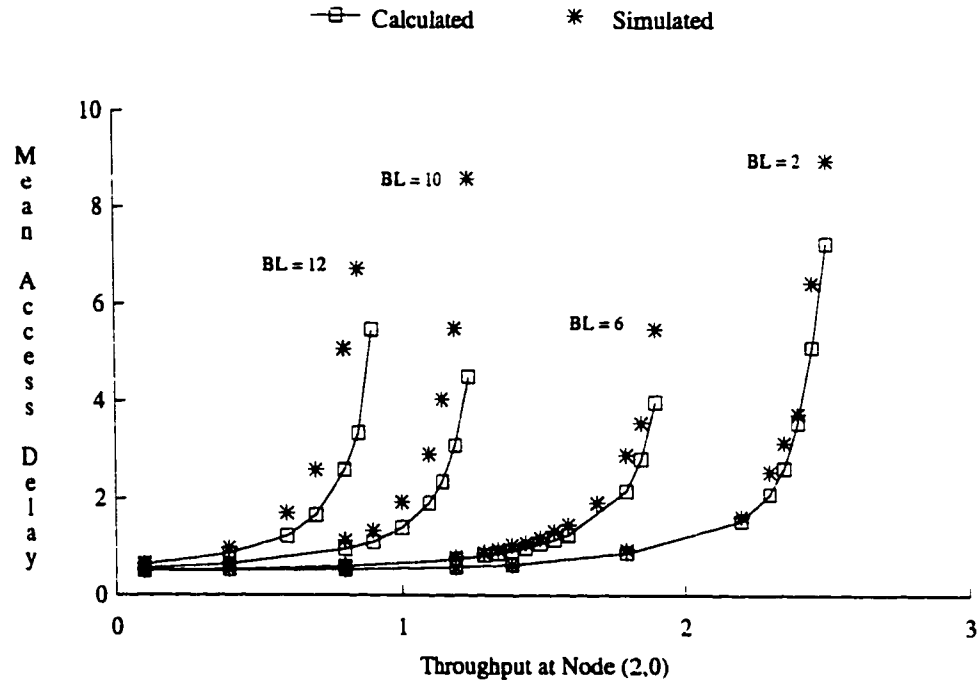


Figure 3.5.11 Access Delay (2,0) 5*5 PDS Network

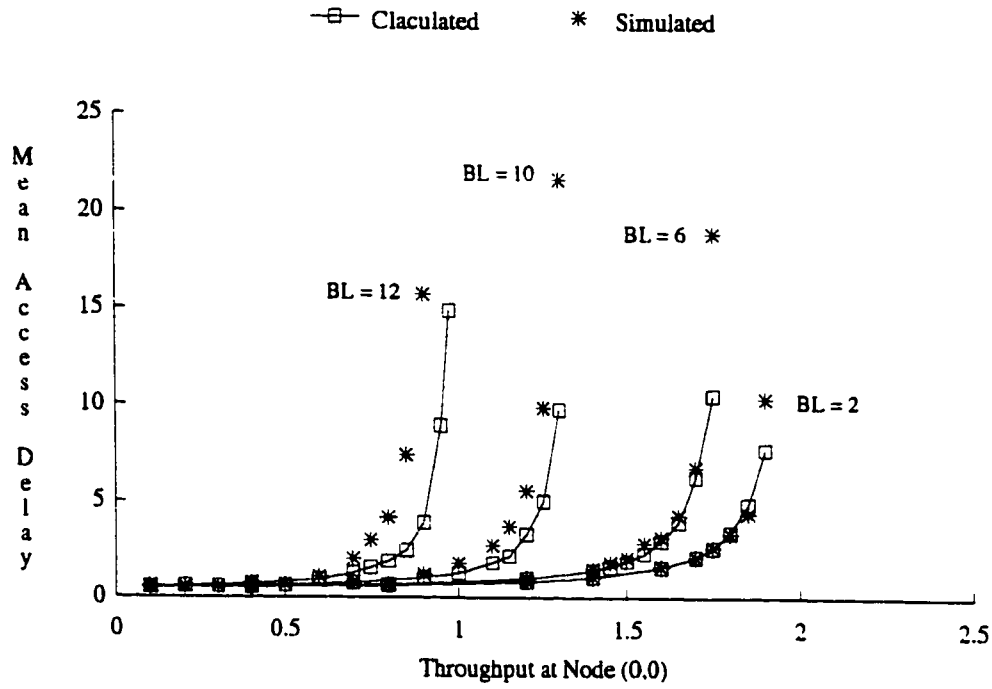


Figure 3.5.12 Access Delay (0,0) 5*5 PDS Network

Figures 3.5.13 and 3.5.14 are included to illustrate this effect. Figure 3.5.13 corresponds to the (2,2) node and shows the effective server capacity available as a function of the throughput at that node. While it is apparent that the two are not independent, the relationship between the available server capacity and the throughput of the nodes does not change considerably as the background load is varied. Figure 3.5.14 illustrates that the same cannot be said for the (0,0) node. Here, the relationship is a strong function of the background load. At small values of BL, the independence assumption is more appropriate than at larger values. The net result is that the analytical model accurately estimates the capacity available to the node at light loads, but tends to slightly overestimate the value at large values of BL. In general, the amount of correlation present in the transit slots is dependent upon the position of the node within the network, the access, routing and allocation strategies used, and the size of the network.

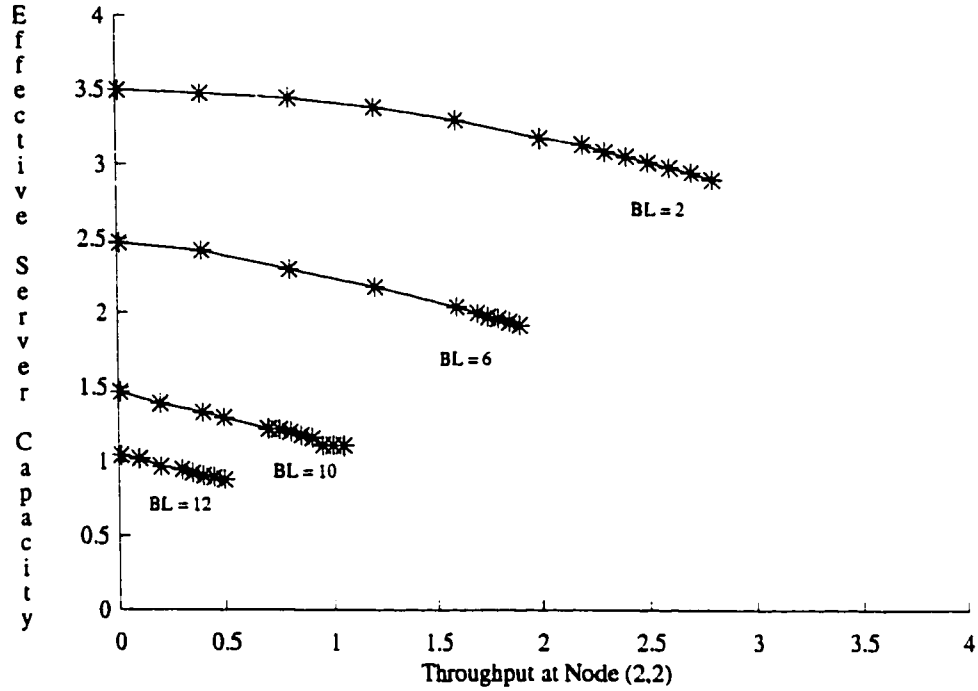


Figure 3.5.13 Effective Capacity (2,2) 5*5 PDS Network

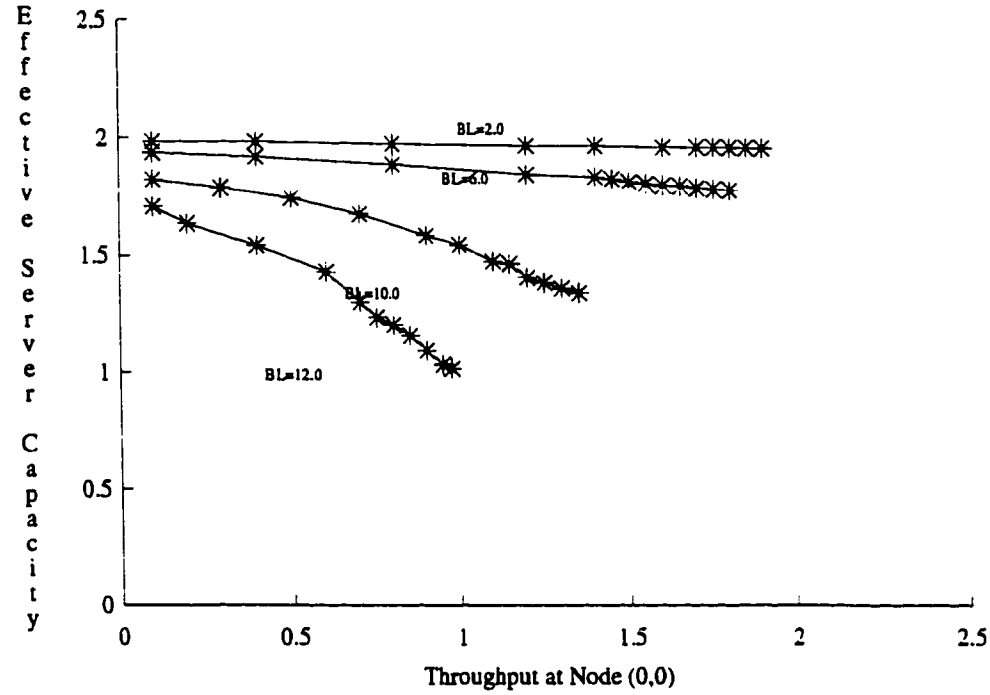


Figure 3.5.14 Effective Capacity (0,0) 5*5 PDS Network

Figures 3.5.15 to 3.5.17 illustrate the model performance for a 9-node ROS algorithm. In this case, note that the model performs better at the edge and corner nodes than it does at the centre node. This is attributed to the pre-routing access at adjacent nodes, which increases the correlation between the transit slots and the queue states at the centre node.

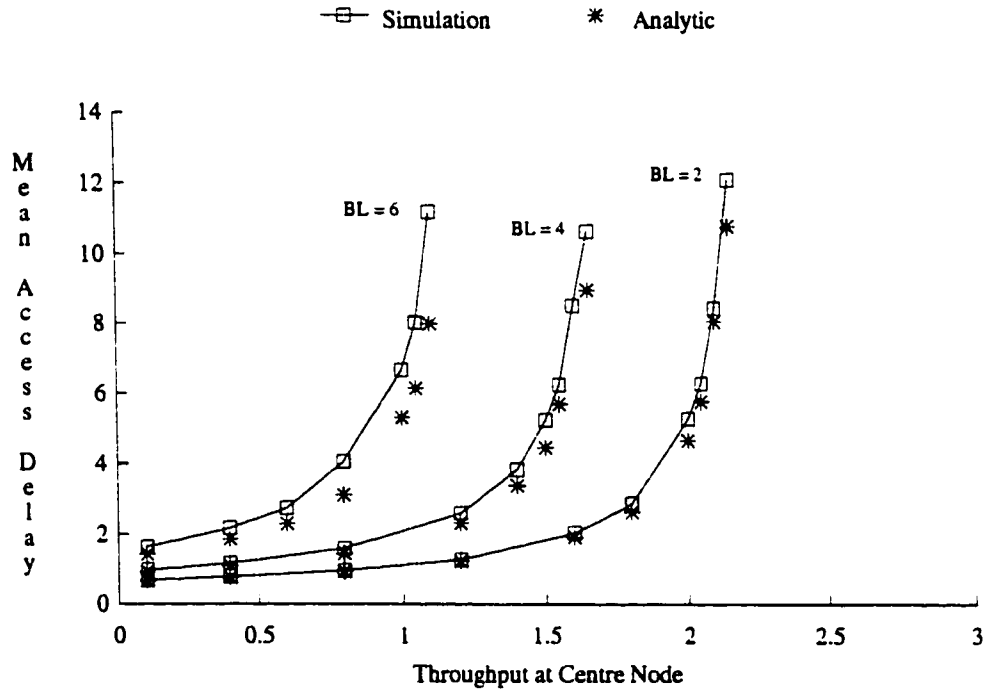


Figure 3.5.15 Access Delay for (1,1) 3*3 ROS Network

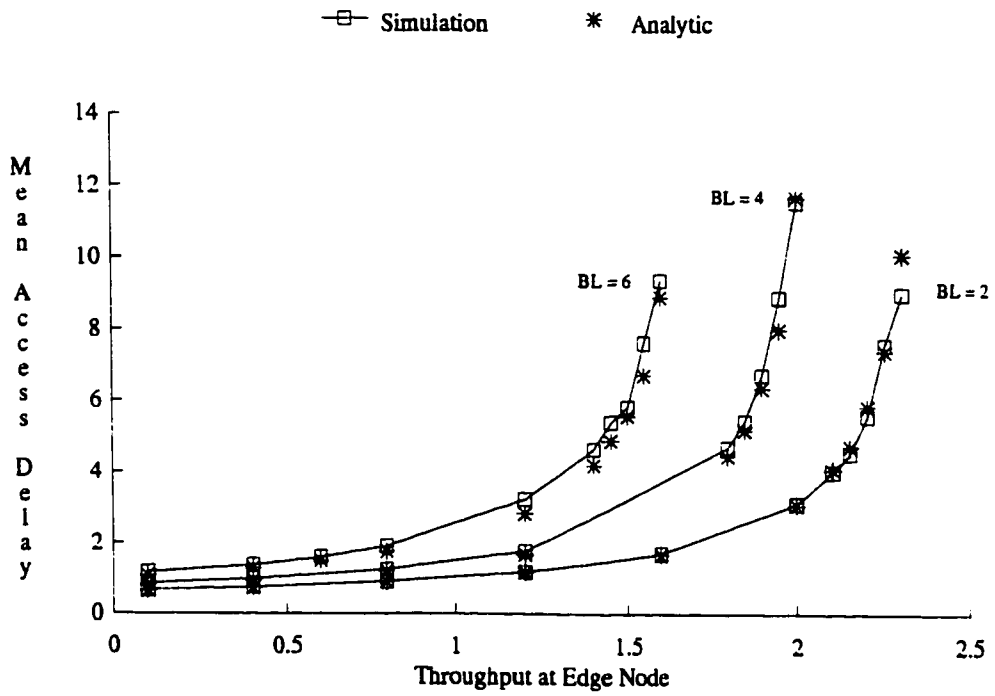


Figure 3.5.16 Access Delay for (1,0) 3*3 ROS Network

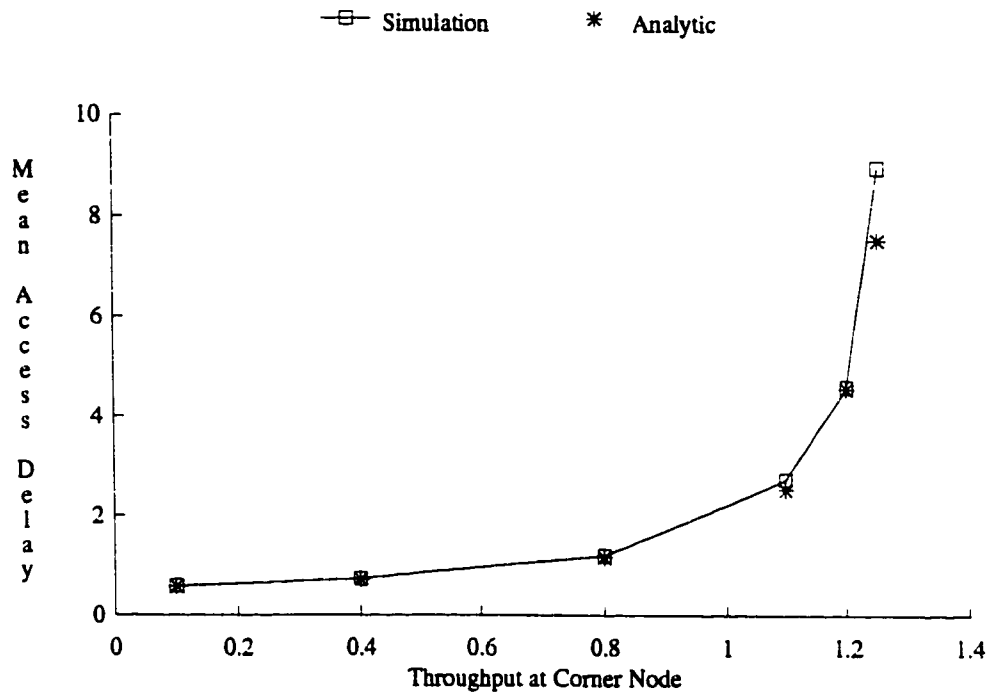


Figure 3.5.17 Access Delay for (0,0) 3*3 ROS Network

Figures 3.5.18 to 3.5.20 illustrate this correlation effect at the centre node.

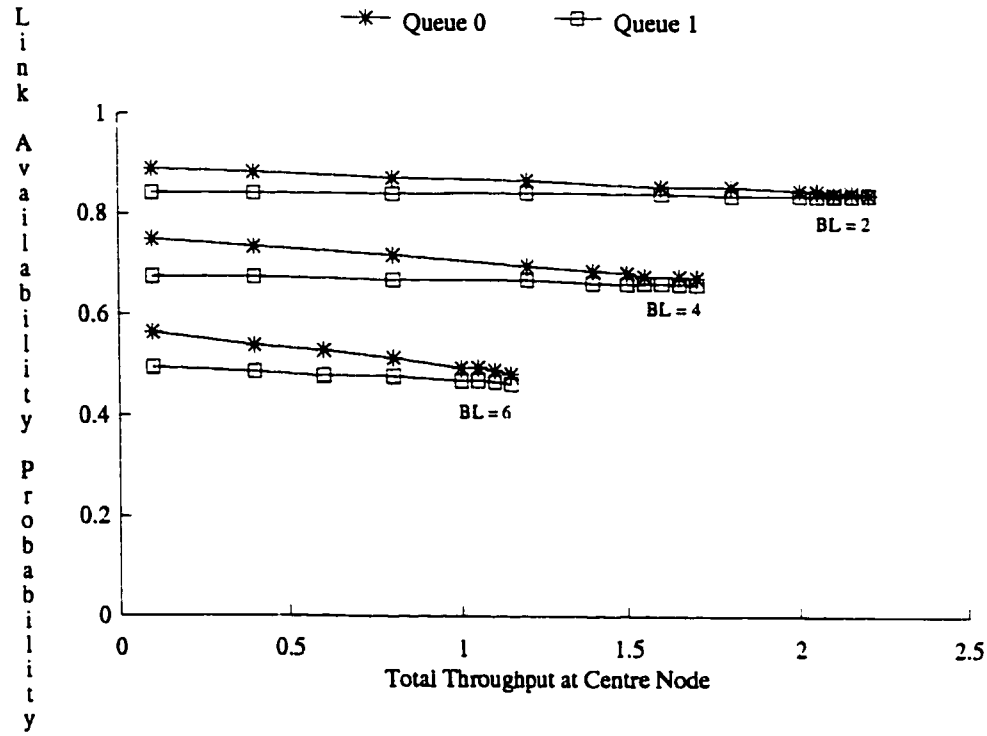


Figure 3.5.18 Server Capacity at (1,1) 3*3 ROS Network

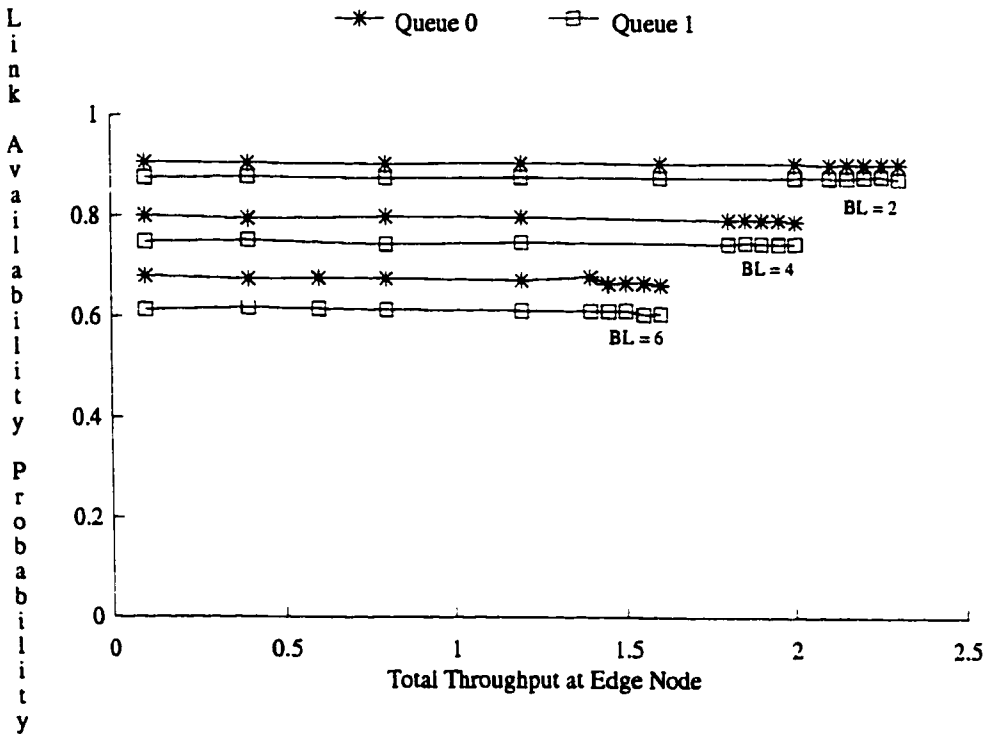


Figure 3.5.19 Server Capacity at (1,0) 3*3 ROS Network

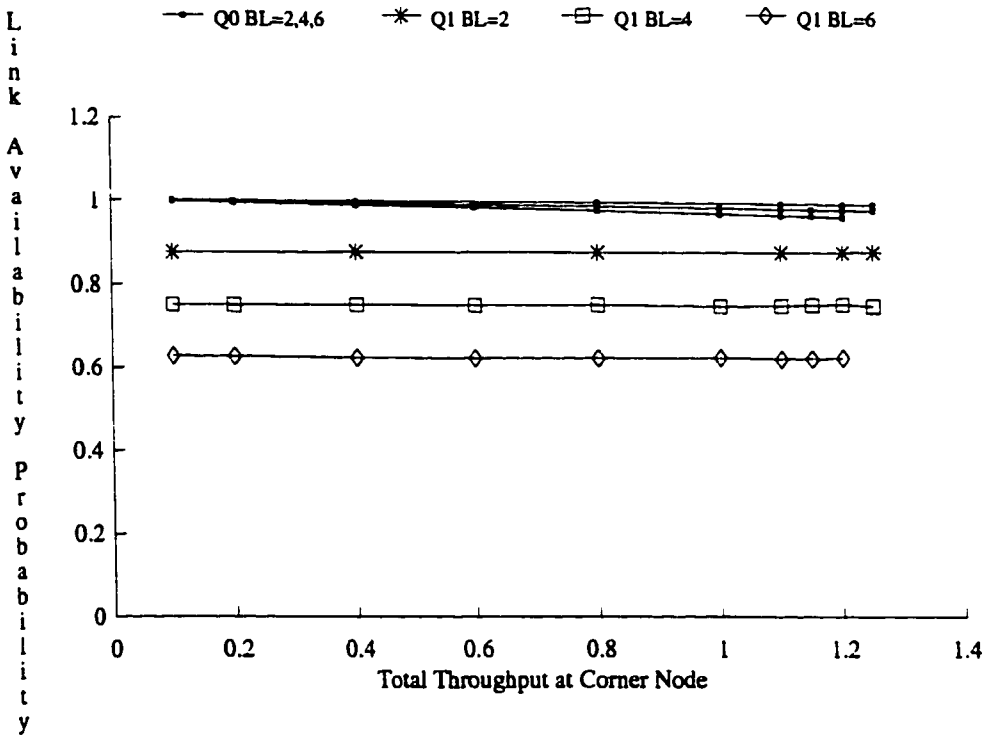


Figure 3.5.20 Server Capacity at (0,0) 3*3 ROS Network

Figures 3.5.21 to 3.5.23, and 3.5.24 to 3.5.26 are included for completeness and show good model performance for the RDR algorithm.

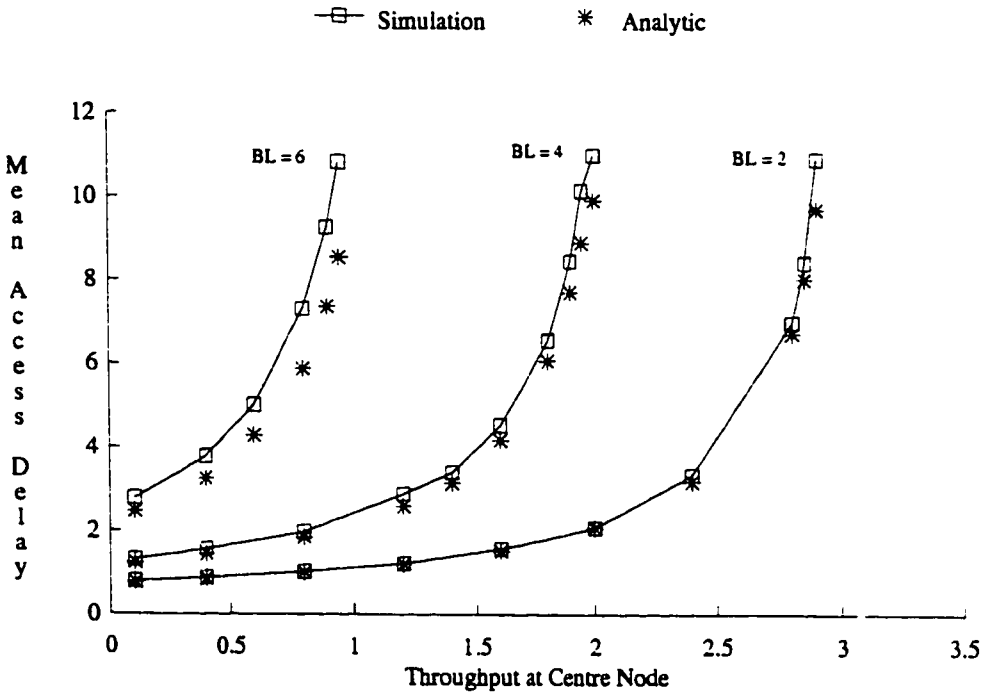


Figure 3.5.21 Access Delay for (1,1) 3*3 RDR Network

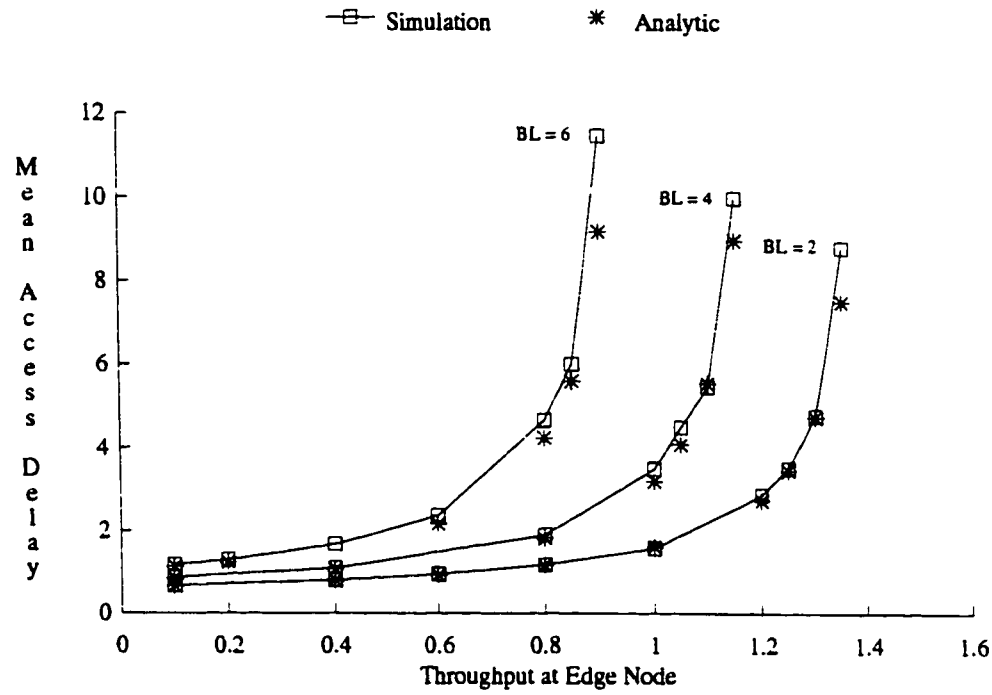


Figure 3.5.22 Access Delay for (1,0) 3*3 RDR Network

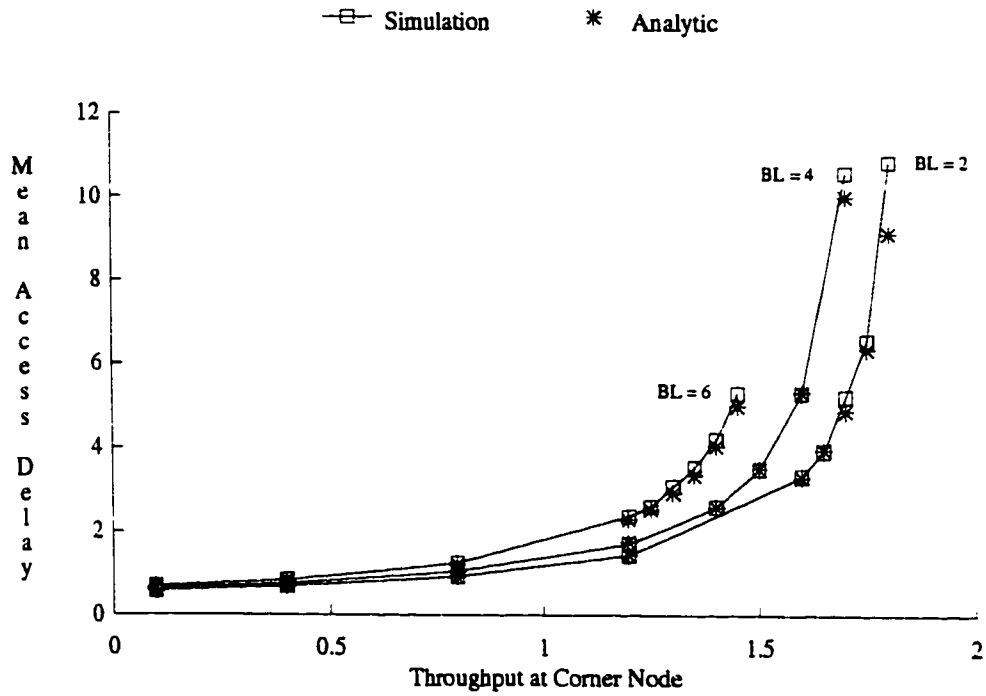


Figure 3.5.23 Access Delay for (0,0) 3*3 RDR Network

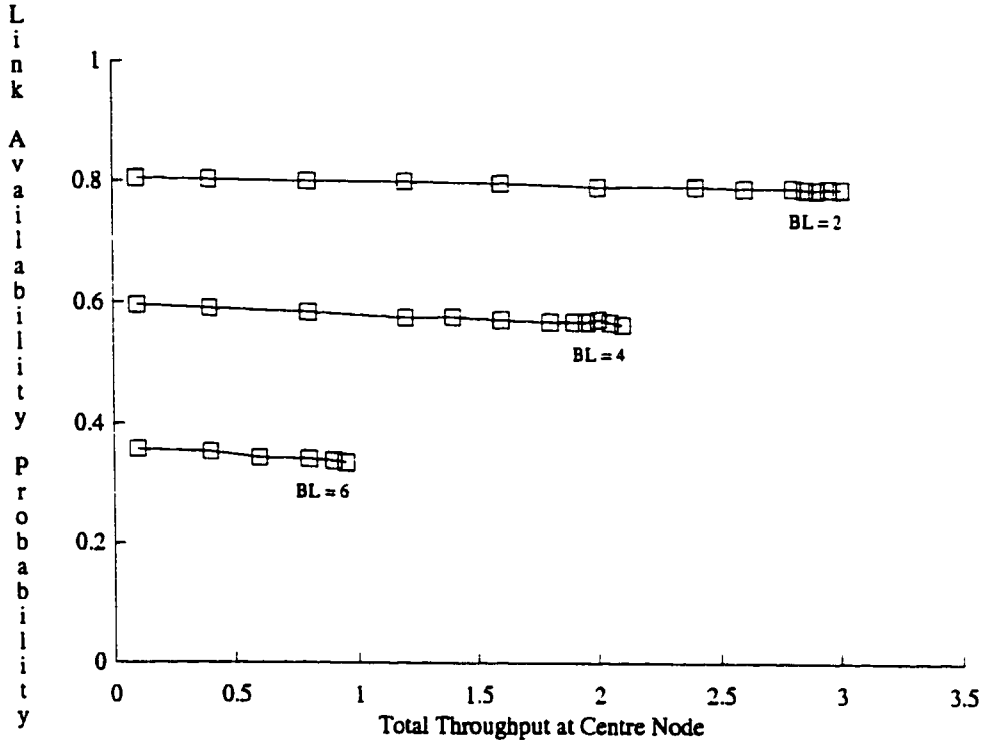


Figure 3.5.24 Server Capacity at (1,1) 3*3 RDR Network

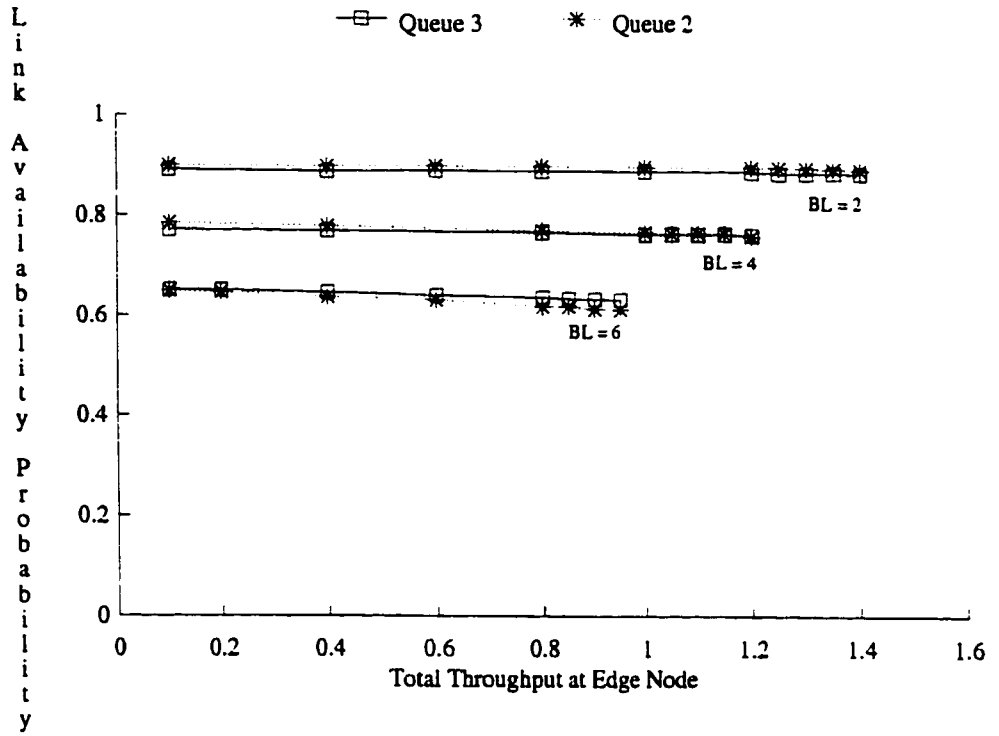


Figure 3.5.25 Server Capacity at (1,0) 3*3 RDR Network

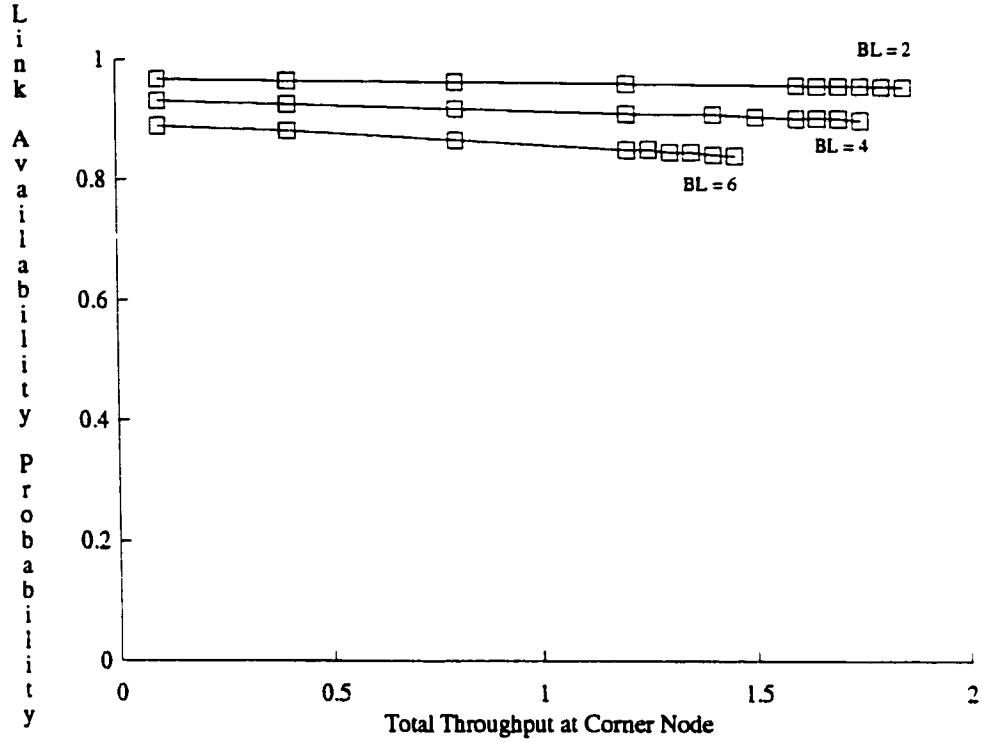


Figure 3.5.26 Server Capacity at (0,0) 3*3 RDR Network

3.6 Summary

One obvious conclusion of this work is that it is feasible to design distributed routing traffic processing algorithms for PMNets. Further work is still required to determine an appropriate implementation, but the idea of simplifying the node design, through the use of deflection routing, appears viable. The nature of PMNets allows for higher network connectivity (and bandwidth allocation) than that which is available using more traditional extended-LAN approaches. The network design is also simplified by the inherent properties of traffic processing, which eliminate the requirement for external flow control techniques and directly offers dynamic routing capabilities to appropriately use the system bandwidth.

The traffic processing within a node may be broken down into three basic strategies, namely

(a) access strategy

(b) routing strategy

(c) allocation strategy

These components, in conjunction with the use of preference vectors and the concept of packet orientation, are important because they provide a valuable framework within which to consider and study the nature of traffic processing for PMNets. Although the definition of these strategies allows for the individual consideration of each, the performance of the network is determined by the combination and interplay of all three. In addition, they are far from independent. This fact leads to some important conclusions.

The performance of the traffic processing is highly dependent upon the complimentary nature of the three strategies used. As a result, trade-offs must occur between the optimization of individual strategies and optimization of overall performance. In every case, it is the access delay that dominates, as the system approaches capacity. Any algorithm design must therefore take this fact into account. It is important to note that the access delay is dependent, to varying degrees, upon all three traffic processing strategies. The optimization of any two of the three, to the exclusion of the third, will consequently not optimize overall performance. Some observations about the relationships of the three strategies are in order. It would appear from our results that the performance of the network is more dependent upon access and allocation than on routing, once attempts are made to decouple access and routing (post-routing access). This statement assumes that the routing algorithm meets the basic requirement of being a shortest path method. The observation is somewhat counter-intuitive when one considers the underlying role that the routing plays. It is also interesting to note that the literature on the design of such algorithms [Maxe87a, Borg87] largely ignores the importance of the allocation algorithm by randomizing it. Although this may prove to be necessary in order to avoid deadlock conditions, our results show that post-routing access with random allocation gives among the worst performance of all twelve algorithms. In fact, if random allocation is used, it is better to apply pre-routing access. Once again, this seems counter-intuitive if one is trying to decrease access delay. The answer, however, lies in the relationship between the three traffic processing strategies, and the trade-off between allowing deflections on entry to the network in order to obtain correspondingly larger reductions in access delay. In some algorithms, the trade-off works (e.g. PDS); and in others, it does not

(e.g. PDR). This same type of reasoning applies elsewhere. Since access delay is important to system performance, one might think that uniformly building the input queues would improve the overall performance by ensuring that the stability of the system is not determined by the worst input queue. In some cases, this is true (PDS/RDS); and in other cases, it is not (ROR/POR). The consideration of all three routing components is essential to ensure good performance. In particular, not recognizing the importance of link allocation, may be a major oversight.

The performance of the algorithms, in decreasing order, is given below.

1) PDS
2) POS
3) PDD
4) POD
5) ROS
6) ROD
7) ROR
8) RDS
9) POR
10) PDR
11) RDD
12) RDR

The first four algorithms illustrate the advantages of coupling post-routing access, which is a less restricted style of access, with a reasonable allocation strategy.

The performance is more dependent upon the type of allocation used than the routing strategy. This may prove important when selecting a routing strategy for implementation, since a significantly less complicated routing strategy may produce only marginally poorer performance. The next three algorithms illustrate that, if pre-routing access is used, the routing strategy is the most important factor and, regardless of the allocation strategy selected, the orthogonal routing method produces better results. They also illustrate that, in general, the order of preference of allocation method is the secondary counter-sort followed by the distance-sort and the random allocation. The final five curves may be divided into two groups. The first group consists of the pre-routing access strategy algorithms (RDS, RDD, RDR) where the observations just stated apply. The other two curves illustrate the poor performance of random allocation, even if post-routing access is used.

Finally, the best algorithms seem to be those that allow access to the network for any available slot (post-routing), combined with a reasonable allocation strategy (secondary counter or distance-sort). In this case, the routing strategy plays a lesser role in determining system performance.

PMNet is a network architecture designed to take advantage of future multi-channel, fibre-optic technology. In this work, a variety of traffic processing algorithms were discussed, and their general performance was compared. It was found that the performance of an algorithm depends upon the relationship between the three individual traffic processing strategies.

A model for calculating detailed link flows rates was developed, yielding accurate results over a wide range of operating parameters. An analytic node delay model was also introduced, which indicated that link independence assumptions may be employed over a wide range of operating conditions for the algorithms tested. The combination of these two models provides a complete delay model for deflection routing in the PMNet, which is valid over a wide operating range.

4.0 SUMMARY

4.1 Discussion

The underlying premise of this work is that fibre-based networks offer design opportunities that are different and largely unexplored. The potential abundance of bandwidth through techniques such as WDM, combined with a broadcast media using novel devices such as transmissive star couplers, opens up design flexibilities that have not been possible in the past. PBNets and PMNets serve as two related examples of new design techniques. In each case, the properties of the underlying fibre optic infrastructure are exploited to provide a multi-hop network that is topologically regular. Topological regularity is used to simplify the processing required at each node in the multi-hop implementation. In our case, we have selected a bus and a two-dimensional grid topology. Others are obviously possible.

With both PBNets and PMNets, the fibre infrastructure is a slotted broadcast media. In order to create a multi-hop or point-to-point network, it was necessary to design media access techniques that define ways to access the media in the presence of transit traffic. The media access must be designed to perform well, while providing fair access to all users. In the case of PBNets, the media access required a modification of an emerging IEEE standard. In the case of PMNets, the media access involved the development of access strategies within the novel concept of deflection routing networks.

A multi-hop implementation on a broadcast media allows for a whole host of novel network design techniques. Associated with this work, three general techniques have been suggested, all of which involve the flexible allocation of bandwidth. Receiver Allocation and Topological Design have been applied to PBNets.

The work described in this thesis has been published in the IEE Electronics Letters [Bign90], the IEEE Transactions on Communications [Todd92], the IEEE Journal of Selected Areas in Communications [Todd94], numerous IEEE conferences [Bign89a], [Todd90], [Todd91], [Todd91a] and an internal report [Bign89].

Specific contributions of the thesis are the development of media access protocols for PBNets, which perform well but are fair under overload conditions. The insight necessary to design this media access was gained from detailed simulations and led directly to the development of concepts associated with the design techniques of Receiver Allocation, Topological Design and Bandwidth Allocation. Associated with this design work, a complete analytical delay model (including input queuing delays) was derived and was used to verify the performance of PBNets. The concept of PMNets as an extension to PBNets allowed for the use of similar flexible network design techniques for allocating bandwidth, and included the framework within which to consider traffic processing at individual nodes. Simulation work led to an understanding of the effects that the interaction of the traffic processing strategies had on performance. Traffic processing algorithms were designed and simulated in detail to improve the understanding of deflection routing in PMNets. The development of a

complete analytic model for both input queuing and transit delays, and comparison with detailed simulations, verified the accuracy of certain independence assumptions.

A number of possible areas for future research have been uncovered as a result of this work. These are described in the next section.

4.2 Future Work

Although future work related to this thesis can take many directions, two specific themes are suggested here. The first theme relates to further investigations of the flexible allocation of bandwidth in future networks. Possible areas consist of:

(a) **Further Application of Existing Network Design Techniques**

In PBNets, Receiver Allocation and Topological Design techniques have been investigated. The possibility of allocating pairs of receivers and transmitters, thus creating new parallel links, has been discussed but not investigated. This has been called the Bandwidth Allocation design technique. PMNets were conceived and designed to allow for flexible deployment of bandwidth. Both Topological and Bandwidth Allocation design techniques could be investigated on PMNets. Within each of these techniques, there would be specific algorithm design and verification requirements.

(b) **New Network Design Techniques**

The techniques of Receiver Allocation, Bandwidth Allocation and Topological Design are not exhaustive. There is research potential in defining

and investigating other design techniques that take advantage of the properties of multi-hop networks like PBNet and PMNet. This could include the dynamic allocation of bandwidth. In addition, there is the possibility of considering different topologies (other than bus and grid topologies).

(c) **Networks of networks**

It is generally accepted that communities of interest will develop on future networks. PMNets were designed to allow for the interconnection of networks, but little work has been done to develop the concept in conjunction with the flexible allocation of bandwidth.

A second theme for future research relates to obtaining a more complete understanding of PBNets and PMNets. This leads to the following possible research areas:

(a) **Consideration of Different Traffic Types**

There has been little investigation into the manner in which PBNets and PMNets would support multiple priority or isochronous traffic types. With PBNets, this could likely build on existing standards, but with PMNets, it would involve more original design work. The development of techniques to support these traffic types, while still supporting concepts of flexible bandwidth allocation, would prove particularly interesting. In addition, the issues of unbalanced traffic and fairness considerations need to be addressed further for PMNets.

(b) **Comparisons with Other Networks**

PBNets and PMNets were developed for the investigation of flexible design techniques. In the literature, many networks have been proposed for a variety of purposes. Comparisons to-date have been incomplete and qualitative. It may be valuable to undertake a more disciplined comparison of such networks.

(c) **Analytical Modelling**

Analytical modelling often leads to, or confirms, a deeper level of insight into the behaviour of networks. This is particularly true with network concepts that are new and poorly understood. At a minimum, the use of analytical work to confirm the results of simulations, and vice-versa, is valuable. Modelling delay in PBNets and PMNets is still significantly unexplored, especially under non-uniform conditions. A study to model the distribution of delay for transit packets in PMNet could, for example, give valuable insight into how to support isochronous traffic on the network.

REFERENCES

[Abou88] O.S. Aboul-Magd and A. Leon-Garcia, "Performance Analysis of a Finite Buffer Burst-Switched Node," IEEE INFOCOM'88, pp.669-677, New Orleans, LA, March 1988.

[Acam87] A.S. Acampora, "A Multichannel Multihop Local Lightwave Network," Proceedings of Globecom'87, Tokyo, Japan, November 1987.

[Acam88] A.S. Acampora, M.J. Karol and M.G. Hluchyj, "Multihop Lightwave Networks: A New Approach to Achieve Terabit Capabilities," Proc. ICC'88, Vol.1, pp.1478-1484, 1988.

[Ades87] S. Ades, "A High Speed Network Interface for Integrated Services," IEEE INFOCOM'87, pp.1092-1101, San Francisco, CA.

[Ahma88] H. Ahmadi, W. Denzel, E. Port and C.A. Murphy, "A High-Performance Switch Fabric for Integrated Circuit and Packet Switching," IEEE INFOCOM'88, pp.9-18, New Orleans, LA, March 1988.

[Anid88] G.J. Anido and A.W. Seeto, "Multipath Interconnection: A Technique for Reducing Congestion Within Fast Packet Switching Fabrics," IEEE Journal on Selected Areas in Communications, Vol.6, No.9, pp.1480-1488, December 1988.

[Arth88] E. Arthurs, R. Boorstyn and T.T. Lee, "The Architecture of a Multicast Broadband Packet Switch," IEEE INFOCOM'88, pp.1-8, New Orleans, LA, March 1988.

[Bail54] N.T.J. Bailey, "On Queuing Processes with Bulk Service," Journal of the Royal Statistical Society (1954).

[Bane96] D. Banerjee and B. Mukherjee, "A Practical Approach for Routing and Wavelength Assignment in Large Wavelength-Routed Optical Networks," IEEE Journal on Selected Areas in Communications, Vol.14, No.5, p.903-911, June 1996.

[Bann95] J. Bannister, F. Borgonovo, L. Fratta and M. Gerla, "A Performance Model of Deflection Routing in Multibuffer Networks with Nonuniform Traffic," IEEE Transactions on Networking, Vol.3, No.5, pp.509-519, October 1995.

[Bern88] F. Bernanbei, A. Forcina, M. Listanti and F.U. Bordoni, "On Non-Blocking Properties of Parallel Delta Networks," IEEE INFOCOM'88, pp.326-333, New Orleans, LA, March 1988.

[Bert87] D. Bertsekas and R. Gallagher, "Data Networks," Prentice-Hall Inc., Eaglewood Cliffs, NJ, 1987.

[Bign89] A.M. Bignell and T.D. Todd, "SIGnet: An Ultra-High-Speed Interconnected Lightwave Network," Internal Report, 1989.

[Bign89a] A.M. Bignell and T.D. Todd, "SIGnet: A New Ultra-High-Speed Lightwave Network Architecture," IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, pp.40-43, Victoria, BC, June 1989.

[Bign90] A.M. Bignell and T.D. Todd, "Analytic Node Model for Deflection Routing Networks," IEE Electronics Letters, Vol.26, No.1, January 4, 1990.

[Bisd90] C. Bisdikian, "Waiting Time Analysis in a Single Buffer DQDB (802.6) Network," IEEE INFOCOM'90, pp.610-616, San Francisco, CA, June 5-7, 1990.

[Bono94] A. Bononi and P.R. Prucal, "New Structures of the Optical Node in Multihop Transparent Optical Networks with Deflection Routing," IEEE INFOCOM'94, pp.415-422, Toronto, ON, June 1994.

[Bore95] M.S. Borella and B. Mukherjee, "Efficient Scheduling of Nonuniform Packet Traffic in a WDM/TDM Local Lightwave Network with Arbitrary Transceiver Tuning Latencies," IEEE INFOCOM'95, pp.129-137, Boston, MA, April 1995.

[Borg87] F. Borgonovo and E. Cadorin, "HR4-Net: A Hierarchical Random-Routing Reliable and Reconfigurable Network for Metropolitan Area," Proc. IEEE INFOCOM'87, pp.320-326, San Francisco, CA, April 1987.

[Borg93] F. Borgonovo, A. Lombardo, S. Palazzo and D. Panno, "FQDB: A fair Multisegment MAC Protocol for Dual Bus Networks," IEEE Journal on Selected Areas in Communications, Vol.11, No.8, pp.1240-1248, October 1993.

[Borg94] F. Borgonovo, L. Fratta and J. Bannister, "On the Design of Optical Deflection-Routing Networks," IEEE INFOCOM'94, pp.120-129, Toronto, ON, June 1994.

[Brea90] R. Breault and V. Phang, "DQDB Performance Improvement with Erasure Nodes," Contribution 802.6-90/21 to the IEEE 802.6 Working Group, March 1990.

[Brew95] G.B. Brewster and M.K. Vernon, "The Fairness of DQDB Networks with Slot Reuse," IEEE INFOCOM'95, PP.1154-1162, Boston, MA, April 1995.

[Chen93] J. Chen, I. Cidon and Y. Ofek, "A Local Fairness Algorithm for Gigabit LANs/MANs with Spatial Reuse," IEEE Journal on Selected Areas in Communications, Vol.11, No.8, pp.1183-1192, October 1993.

[Chou96] A.K. Choudhury and E.L. Hahne, "Dynamic Queue Length Thresholds in a Shared Memory ATM Switch," IEEE INFOCOM'96, PP.679-779, San Francisco, CA, March 1996.

[Chla93] I. Chlamtac and A. Fumagalli, "An Optical Switch Architecture for Manhattan Networks," IEEE Journal on Selected Areas in Communications, Vol.11, No.4, pp.550-559, May 1993.

[Chla96] Chlamtac, A. Fumagalli, Kazovsky, Melman, Nelson, Poggiolini, Cerisola, Choudhury, Fong, Hofmeister, Lu, Mekkittikul, Sabido, Suh and Wong, "CORD: Contention Resolution by Delay Lines," IEEE Journal on Selected Areas in Communications, Vol.14, No.5, pp.1014-1028, June 1996.

[Chla96a] I. Chlamtac, A. Fumagalli and C.J. Suh, "A Delay Line Receiver Architecture for All-Optical Networks," IEEE INFOCOM'96, pp.419-426, San Francisco, CA, March 1996.

[Cido97] I. Cidon, L. Georgiadis, R. Guerin and Y. Shavitt, "Improved Fairness Algorithms for Rings with Spatial Reuse," IEEE Transactions on Networking, Vol.5, No.2, p.190-203, April 1997.

[Cont89] M. Conti, E. Gregori and L. Lenzini, "DQDB Media Access Control Protocol: Performance Evaluation and Unfairness Analysis," Third IEEE Workshop MAN's, pp.375-408, San Diego, CA, March 1989.

[Cont90] M. Conti, E. Gregori and L. Lenzini, "DQDB Under Heavy Load: Performance Evaluation and Fairness Analysis," Proc. of IEEE INFOCOM'90, pp.313-320, San Francisco, CA, June 1990.

[Cruz88] R.L. Cruz, "Maximum Delay in Buffered Multistage Interconnection Networks," IEEE INFOCOM'88, pp.135-144, New Orleans, LA, March 1988.

[Cruz96] R.L. Cruz and J.-T. Tsai, "COD: Alternative Architectures for High Speed Packet Switching," IEEE Transactions on Networking, Vol.4, No.1, pp.11-22, February 1996.

[Dobo96] W. Dobosiewicz and P. Gburzynski, "A Bounded-hop-count Deflection Scheme for Manhatten-street Networks," IEEE INFOCOM'96, pp.172-179, San Francisco, CA, March 1996.

[Eckb88] A.E. Eckberg and T.-C. Hou, "Effects of Output Buffer Sharing on Buffer Requirements in an ATDM Packet Switch," IEEE INFOCOM'88, pp.459-466, New Orleans, LA, March 1988.

[Eise88] M. Eisenberg and N. Mehravari, "Performance of the Multichannel Multihop Lightwave Network under Non-uniform Traffic," IEEE JSAC, Vol.6, August 1988.

[Eng88] E.Y. Eng, M.G. Hluchyj and Y.S. Yeh, "Multicast and Broadcast Services in a Knockout Packet Switch," IEEE INFOCOM'88, pp.29-34, New Orleans, LA, March 1988.

[Fili89] J. Filipiak, "Access Protection for Fairness in a Distributed Queue Dual Bus Metropolitan Area Network," IEEE ICC'89, pp.635-639, Boston, MA, June 1989.

[Fine84] M. Fine and F.A. Tobagi, "Demand Assignment Multiple Access Schemes in Broadcast Bus Local Area Networks," IEEE Transactions on Computers, Vol.C-33, No.12, pp.1130-1159, December 1984.

[Gree86] A.G. Greenberg and J. Goodman, "Sharp Approximate Models of Adaptive Routing in Mesh Networks," Teletraffic Analysis and Computer Performance Evaluation, Elsevier Science Publishers B.V. (North-Holland), pp.255-270, 1986.

[Gree96] P.E. Green, "Optical Networking Update," IEEE Journal on Selected Areas in Communications, Vol.14, No.5, pp.764-779, June 1996.

[Haas93] Z. Haas, "Growability of the 'Staggering Switch' Architecture," IEEE ICC'93, pp.578-586, Geneva, Switzerland, May 1993.

[Hahn90] E.L. Hahn, A.K. Choudhury and N.F. Maxemchuk, "Improving the Fairness of Distributed-Queue-Dual-Bus Networks," Proc. of IEEE INFOCOM'90, pp.175-184, San Francisco, CA, June 1990.

[Hahn91] E.L. Hahn and N.F. Maxemchuk, "Fair Access of Multi-Priority Traffic to Distributed-Queue Dual Bus Networks," Proceedings of IEEE INFOCOM'91, pp.889-900, Bal Harbour, FL, April 1991.

[Hahn92] E.L. Hahn, A.K. Choudhury and N.F. Maxemchuk, "DQDB Networks With and Without Bandwidth Balancing," IEEE Transactions on Communications, Vol.40, No.7, pp.1192-1204, July 1992.

[Hall96] E. Hall, J. Kravitz, R. Ramaswami M. Halvorson, S. Tenbrink and R. Thomsen, "The Rainbow-II Gigabit Optical Network," IEEE Journal on Selected Areas in Communications, Vol.14, No.5, pp.814-822, June 1996.

[Haye84] J.F. Hayes, "Modelling and Analysis of Computer Communication Networks," New York: Plenum Press, 1984.

[Hluc88] M.G. Hluchyj and M.J. Karol, "Shuffle Net: An Application of Generalized Perfect Shuffles to Multihop Lightwave Networks," IEEE INFOCOM'88, pp.379-390, New Orleans, LA, March 1988.

[Huan95] N. Huang and S. Sheu, "DTCAP - A Distributed Tunable-channel Access Protocol for Multi-channel Photonic Dual Bus Networks," IEEE INFOCOM'95, pp.908-917, Boston, MA, April 1995.

[Huan95a] N. Huang and S. Sheu, "A Waste-free Congestion Control Scheme for Dual Bus High-speed Networks," IEEE ICC'95, pp.940-948, Seattle, WA, June 1995.

[IEEE87] Minutes of the IEEE 802.6 Metropolitan Area Networks Committee, November 9-13, Ford Lauderdale, Fla., 1987.

[IEEE90] "Distributed Dual Bus (BQDB) Subnetwork of a Metropolitan Area Network (MAN)," IEEE 802.6 Proposal and submissions, February 7, 1990.

[Ines95] J. Iness, S. Banerjee and B. Mukherjee, "GEMNET: A Generalized Suffle-exchange-based, Regular, Scalable, Modular, Multihop, WDM Lightwave Network," IEEE Transactions on Networking, Vol.3, No.4, pp.470-476, August 1995.

[Jano96] M.W. Janoska and T.D. Todd, "A Single-hop Wavelength Routed LAN/MAN Architecture," IEEE INFOCOM'96, pp.402-409, San Francisco, CA March 1996.

[Jue96] J.P. Jue, M.S. Boreela and B. Jukherjee, "Performance Analysis of the Rainbow WDM Optical Network Prototype," IEEE Journal on Selected Areas in Communications, Vol.14, No.5, pp.945-952, June 1996.

[Kaba96] M. Kabatepe and K. Vastola, "The Fair Distributed Queue (FDQ) Protocol for High-speed Metropolitan-area Networks," IEEE Transactions on Networking, Vol.4, No.3, pp.331-339, June 1996.

[Karo88] M.J. Karol and S. Shaikh, "A Simple Adaptive Routing Scheme for ShuffleNet Multihop Lightwave Network," Proceedings of Globecom'88, Hollywood, FL, November 1988.

[Karv93] D. Karvelas and M. Papamichail, "The No Slot Wasting Bandwidth Balancing Mechanism for Dual Bus Architectures," IEEE Journal on Selected Areas in Communications, Vol.11, No.8, p.1214-1228, October 1993.

[Kova96] M. Kavacevic, "On Torus Topologies with Random Extra Links," IEEE INFOCOM'96, pp.410-418, San Francisco, CA, March 1996.

[Lee88a] T.T. Lee, R. Boorstyn and E. Arthurs, "The Architecture of a Multicast Broadband Packet Switch," IEEE INFOCOM'88, pp.1-8, New Orleans, LA, March 1988.

[Lee88b] T.T. Lee, "Nonblocking Copy Networks for Multicast Packet Switching," IEEE Journal on Selected Areas in Communications, Vol.6, No.9, pp.1455-1467, December 1988.

[Lee95] K.-O. Lee and V.O. Li, "Optimization of a WDM Optical Packet Switch with Wavelength Converters," IEEE INFOCOM'95, pp.423-429, Boston, MA, April 1995.

[Lee95a] W.-T. Lee and L.Y. Kung, "Binary Addressing and Routing Schemes in the Manhatten Street Network," IEEE Transactions on Networking, Vol.3, No.1, pp.26-30, February 1995.

[Liew96] S. Liew, "A General Packet Replication Scheme for Multicasting with Application to Shuffle-exchange Networks," IEEE Transactions on Communications, Vol.44, No.8, pp.1021-1030, August 1996.

[Liew97] S. Liew, "On the Stability of Shuffle-exchange and Bidirectional Shuffle-exchange Deflection Networks," IEEE Transactions on Networking, Vol.5, No.1, pp.87-94, February 1997.

[Limb82] J.O. Limb and C. Flores, "Description of Fastnet, A Unidirectional Local Area Communications Network," Bell System Technical Journal, September 1982.

[MacD88] R.I. MacDonald, "Terminology for Photonic Matrix Switches," IEEE Journal on Selected Areas in Communications, Vol.6, No.7, pp.1141-1151, August 1988.

[Magl87] B. Maglaris, R. Boorstyn, S. Panwar and T. Spirtos, "Routing in Burst-Switched Voice/Data Integrated Networks," IEEE INFOCOM'87, pp.162-169, San Francisco, CA.

[Math87] P.W. Mathewson and S.R. Wilbur, "An Integrated Services Switching System Based Upon a Single-Buffered Banyan Network," IEEE INFOCOM'87, pp.766-772, San Francisco, CA.

[Maxe85] N.F. Maxemchuk, "Regular and Mesh Topologies in Local and Metropolitan Area Networks," AT&T Tech. Journal, Vol.64, pp.1659-1686, Sept., 1985.

[Maxe87a] N.F. Maxemchuk, "Routing in the Manhattan Street Network," IEEE Trans. on Commun., Vol.COM-35, No.5, pp.503-512, May 1987.

[Maxe88] N.F. Maxemchuk, "Distributed Clocks in Slotted Networks," IEEE INFOCOM'88, pp.119-125, New Orleans, LA, March 1988.

[Maxe89] N.F. Maxemchuk, "Comparison of Deflection and Store-and-Forward Techniques in the Manhattan Street Network," to appear in IEEE INFOCOM'89, Ottawa, Ontario, Canada, May 1989.

[Maxe93] N. Maxemchuk and R. Krishnan, "A Comparison of Linear and Mesh Topologies - DQDB and the Manhatten Street Network," IEEE Journal on Selected Areas in Communications, Vol.11, No.8, p.1278-1289, October 1993.

[Midw93] J.E. Midwinter, "Photonics in Switching: Volume 1 - Background and Components," Academic Press, San Diego, CA, 1993.

[Midw93a] J.E. Midwinter, "Photonics in Switching: Volume 2 - Systems," Academic Press, San Diego, CA, 1993.

[Modi96] E. Modiano and A. Ephremides, "Efficient Algorithms for Performing Packet Broadcasts in a Mesh Network," IEEE Transactions on Networking, Vol.4, No.4, pp.639-648, August 1996.

[Mokh95] A. Mokhtar and M. Azizoglu, "Hybrid Multiaccess for All-optical LANs with Nonzero Tuning Delays," IEEE ICC'95, .1272-1280, Seattle, WA, June 1995.

[Muir96] A. Muir and J.J. Grcia-Luna-Aceves, "Distributed Queue Packet Scheduling for WDM-Base Networks," IEEE INFOCOM'96, pp.938-944, San Francisco, CA, March 1996.

[Mukh90] B. Mukherjee and S. Banerjee, "Alternative Strategies for Improving the Fairness in and an Analytic Model of DQDB Networks," Computer Science Department, CSE-90-32, University of California, Davis, July 1990.

[Nara95] B. Narahari, S. Shende and R. Simha, "Efficient Algorithms for Erasure Node Placement on DQDB Networks," IEEE ICC'95, p.935-943, Seattle, WA, June 1995.

[Neum88] P. Neuman, "A Broad-Band Packet Switch for Multi-Service Communications," IEEE INFOCOM'88, pp.19-28, New Orleans, LA, March 1988.

[Neum88a] P. Neuman, "A Fast Packet Switch for the Integrated Services Backbone Network," IEEE Journal on Selected Areas in Communications, Vol.6, No.9, pp.1468-1479, December 1988.

[Okos87] T. Okoshi, "Recent Advances in Coherent Optical Fiber Communication Systems," IEEE Journal of Lightwave Technology, Vol.LT-5, No.1, January 1987.

[Pach93] A. Pach, S. Palazzo and D. Panno, "Slot Pre-using in IEEE 802.6 Metropolitan Area Networks," IEEE Journal on Selected Areas in Communications, Vol.11, No.8, pp.1249-1258, October 1993.

[Park95] S.-W. Park and Y.-C. Kim, "A Virtual Topology for WDM Multihop Lightwave Networks," IEEE INFOCOM'95, pp.701-709, Boston, MA, April 1995.

[Qiao96] C. Qiao, "Analysis of Space-time Tradeoffs in Photonic Switching Networks," IEEE INFOCOM'96, pp.822-829, San Francisco, CA, March 1996.

[Rama95] R. Ramaswami and K. Sivarajan, "Routing and Wavelength Assignment in All-optical Networks," IEEE Transactions on Networking, Vol.3, No.5, pp.489-500, October 1995.

[Rama95] R. Ramaswami and K. Sivarajan, "Design of Logical Topologies for Wavelength-routed All-optical Networks," IEEE INFOCOM'95, p.1316-1324, Seattle, WA, June 1995.

[Reed87] D.A. Reed and R.M. Fujimoto, "Multicomputer Networks: Message-Based Parallel Processing," MIT Press, 1987.

[Rous95] G.N. Rouskas and M.H. Ammar, "Minimizing Delay and Packet Loss in Single-hop Lightwave WDM Networks Using TDM Schedules," IEEE ICC'95, pp.1267-1271, Seattle, WA, June 1995.

[Rubi96] I. Rubin and H. Wu, "Performance Analysis and Design of CQBT Algorithm for a Ring Network with Spatial Reuse," IEEE Transactions on Networking, Vol.4, No.4, pp.649-659, August 1996.

[Sabe96] R. Sabella, E. Iannone and E. Pagano, "Optical Transport Networks Employing All-optical Wavelength Conversion: Limits and Features," IEEE Journal on Selected Areas in Communications, Vol.14, No.5, p.968-977, June 1996.

[Schu97] K.J. Schultz and P.G. Gulak, "Physical Performance Limits for Shared Buffer ATM Switches," IEEE Transactions on Communications, Vol.45, No.8, pp.997-1005, August 1997.

[Seo96] S.-W. Seo, P.R. Pruncal and H. Kobayashi, "Generalized Multihop Shuffle Networks," IEEE Transactions on Communications Vol.44, No.9, p.1205-1216, September 1996.

[Shar94] O. Sharon and A. Segall, "On the Efficiency of Slot Reuse in the Dual Bus Configuration," IEEE INFOCOM'94, p.758-766, Toronto, ON, June 1994.

[Shar95] O. Sharon, "A Proof for Lack of Starvation in DQDB with and without Slot Reuse," IEEE INFOCOM'95, pp.1180-1187, Boston, MA, April 1995.

[Tant94] A. Tantawy and C. Bisdikian, "Source Assisted Partial Destination Slot Release in Slotted Networks," IEEE INFOCOM'94, pp.736-744, Toronto, ON, June 1994.

[Todd90] T.D. Todd and A.M. Bignell, "Performance Modelling on the SIGnet MAN Backbone," Proc. IEEE INFOCOM'90, pp.192-199, San Francisco, CA, June 1990.

[Todd91] T.D. Todd, Z. Khurshid, A.M. Bignell and S. Sivakumaran, "Photonic Multihop Bus Networks," IEEE INFOCOM'91, pp.981-990, Bal Harbour, FL, April 1991.

[Todd91a] T.D. Todd, A.M. Bignell and S. Sivakumaran, "Photonic Bus Local and Metropolitan Area Networks," ICC'91.

[Todd92] T.D. Todd and A.M. Bignell, "Traffic Processing Algorithms for the SIGNET Metropolitan Area Network," IEEE Transactions on Communications, pp.576-586, March 1992.

[Todd94] T.D. Todd and A.M. Bignell, "A Slot Reuse Protocol for Rearrangeable Dual-Bus Networks," IEEE Transactions on Communications, Vol.42, No.2/3/4, pp.1131-1140, March 1994.

[Trid97] S. Tridandapani, B. Mkhurjee and G. Hallingstad, "Channel Sharing in Multi-hop Lightwave Networks: Do We Need More Channels," IEEE Transactions on Networking, Vol.5, No.5, pp.719-730, October 1997.

[Turn88] J.S. Turner, "Design of a Broadcast Packet Switching Network," IEEE Trans. on Commun., Vol.36, No.6, pp.734-743, June 1988.

[Wong89] J.W. Wong, "Throughput of DQDB Networks Under Heavy Load," EFOC/CAN-89, pp.146-151, Amsterdam, The Netherlands, June 1989.

[Yate96] J. Yates, J. Lacey, D. Everitt and M. Summerfield, "Limited-range Wavelength Translation in All-optical Networks," IEEE INFOCOM'96, pp.954-961, San Francisco, CA, March 1996.

[Zade63] L.A. Zadeh and C.A. Desoer, "Linear System Theory: The State Space Approach," McGraw Hill, New York, NY, 1963.

[Zuke90] M. Zukerman and P.G Potter, "A Protocol for Erasure Node Implementation within the DQDB Framework," Proceedings of IEEE Globecom'90, pp.1400-1404, San Diego, CA, December 1990.