National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

Metropolitan Area Network Using Photonic Switching

by

Wei Wei Cao, B.Sci.

A thesis submitted to
the School of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of

Master of Applied Science

Ottawa-Carleton Institute for Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
University of Ottawa
Ottawa, Ontario, Canada, K1N 6N5

September 30, 1993

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

ISBN 0-315-95893-6

Contents

List of Figures

List of Tables

Abstract

The high-bandwidth characteristic has made fiber optics very attractive for metropolitan area network (MAN) implementation. This paper describes the architecture and performance of an integrated fiber optics MAN called TreeNet. The upper level of the TreeNet consists of a tree and the lower level consists of linear buses. Stations are connected to the linear buses via passive taps while branches of the tree are interconnected using passive couplers. Active nodes are used between the two levels for traffic filtering. A Wavelength Division Multiplexing (WDM) multihop architecture, ShuffleNet, is used for this TreeNet. Three hierarchical design models are proposed: 1) ShuffleNet in the upper level and Synchronous Time Division Multiplexing (STDM) in the lower level, 2) ShuffleNets in both upper and lower levels with dedicated channels between stations and the active nodes, 3) same as 2) except there are no dedicated channels. The performances of networks based on these three proposals are evaluated and discussed.

Acknowledgements

I would like to thank my thesis supervisor Dr. Oliver Yang for his help, encouragement, and patience during the course of this work. I would like to particularly thank him for his many comments in the writing of the thesis and a paper.

I would like to thank my husband, Hong, for his constant encouragement and support throughout the course of my study.

This research was supported by a TRIO Research Assistantship under Dr. M. Kavehrad's Thrust on Photonic Networks and System.

Chapter 1

Introduction

1.1 Background

With the continuing success of Local Area Networks (LANs), demand is evolving in the direction of extending their capabilities towards higher data rates and wider areas coverage. This, together with the progress in fiber-optic technology, has produced the so-called Metropolitan Area Networks (MANs). The MANs can cover much wider geographical regions than the LANs, and offers data rates suitable for public telecommunication networks. The MANs are also suitable for the interconnection of LANs, Private Automatic Branch Exchanges(PABXs), or other current communication equipment, as well as new, higher data-rate equipment. They are natural candidates for providing integrated services, which is a requirement for all new telecommunication networks.

MANs should be capable of operating over areas of at least 50 kilometers in diameter, providing high speed data transmission, and supporting synchronous traffic that has stringent requirements on delay, bandwidth, robustness and reliability. The low loss and decreasing cost of optical fiber are two of the driving forces leading to widespread deployment of optical fiber system in the telecommunication networks. It is within the realm of possibility that one single-mode fiber can provide thousands of individual wavelengths, each carrying information over hundreds of kilometers at Gigabit rates. These advantages make the use of single-mode fiber very attractive for implementation of the MANs, especially for the implementation of high speed networks.

The following sections in this chapter will overview the MAN, photonic network and the evolution of B-ISDN. These are the foundation of this thesis. Then the motivation of research and approach is discussed. Finally, the organization of this thesis is presented.

1.2 Overview of the Metropolitan Area Network

1.2.1 Structure of a MAN

A MAN has a structure similar to a distributed LAN. Nodes are connected to a common high-speed medium. Each node receives the packets that are addressed to it. Since each node can transmit into the medium, there is the potential for two or more nodes to transmit simultaneously which will result in corruption of the transmission. A Media Access Control (MAC) protocol has to be used to avoid these collisions and to make sure that packets are received by the appropriate nodes.

Therefore, a MAN architecture usually consists of the physical layer, the Media Access Control, eg. IEEE 802.6 Standard. In terms of protocol architecture, this structure fits into the OSI Reference Model as shown in Figure 1.1.

1.2.2 Media Access Control Protocols for MANs

A couple of constraints should be enforced on the MACs for MANs, which are described as follows:

- Maximum Utilization

  Not all the MACs are suitable for networks with large propagation times. For example, the IEEE 802 LAN MACs have a maximum utilization given by the following ([23, 5]):

$$u \propto \frac{1}{1+ka}$$

  where

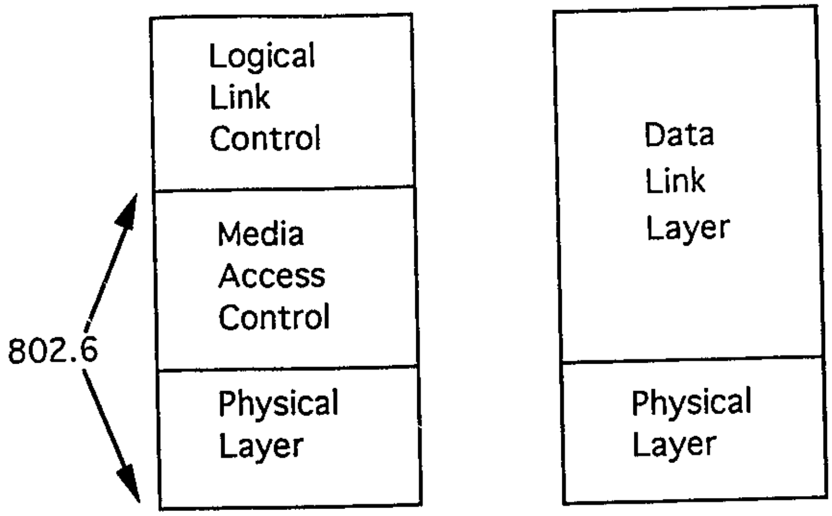

Figure 1.1: MAN Protocols and OSI Protocols

$0 < k < \infty$

$a = RD/L$

R = data rate of the medium

D = propagation time across the medium

L = length of packets sent by a node.

For this type of MAC, if the typical MAN figures of $R = 50$ Mbps medium length of 50 km is used, a packet size of 1000 bits would lead to a maximum utilization of only 0.11. Consequently, a MAC for a MAN should not have these characteristics.

- Bandwidth and Delay

  For most LAN MACs, when a node has a packet to send, the packet experiences a random delay before the MAC makes the medium available to it. Furthermore, the average delay can be so large as to reduce the effective bandwidth available to the node to an unacceptable level depending on the load offered to the network by other nodes. The MAC for the MAN must ensure that the delays between accesses to the medium stay below a fixed amount.

1.2.3 MANs on Optic Fiber

The push of advancing technology and the pull of new, more-demanding user application are stimulating the development of LANs that operate at speeds much higher than those of just a few years ago. If a LAN operates at a speed over 50 Mbps, it is called high speed LAN (HSLAN). If the HSLAN operates in the 100 Mbps range and can cover large geographic area, it will be referred to as a MAN. Typical MAN traffic is expected to include LAN interconnection, graphic and digital images, bulk data transfer, digitized voice, compressed digitized video, and conventional terminal traffic. For example, in a campus environment, video lectures are distributed and videoconferencing is supported. In a hospital, the radiologist is able to retrieve X-rays from an image database. For HSLAN or MAN networks, it is necessary to employ interconnection speeds of 100 Mbps, geographical-area coverage of up to 100 km, and network capabilities that would support hundreds of stations.

The popular physical medium of a MAN is optical fiber. Advances in lightwave technology, specifically fiber optics, enables the fabrication of relatively low-cost, low-attenuation fibers and high-quality optical transmitters, detectors, and passive components. All of these enables a high-speed backbone network to link LANs across metropolitan regions, creating a new version of "stretched" LAN, or the MAN, which has many characteristics in common with the LAN.

The Fiber Distributed Data Interface (FDDI) ([37, 36, 21]) and the IEEE 802.6 Distributed Queueing Dual Bus (DQDB) ([37, 31, 36, 21]) are emerging standards for high-speed (45 - 150 Mbps) MANs. The most common medium is optical fiber.

The early MAN was originated from a LAN architecture, specifically, the IEEE standard 802.5 token ring, by replacing the copper-based cable with single-mode or multimode fiber and implementing the FDDI. The FDDI is a 100 Mbps token-passing ring that utilizes optical fiber for transmission between stations, and has dual center rotating rings to provide redundant data paths for reliability (see Figure 1.2). The secondary ring can be used when the primary ring fails. The MAC protocol FDDI is similar to that of the token ring, the two main differences being the token passing scheme and the Timed Token Rotation Protocol used to control access to the medium.

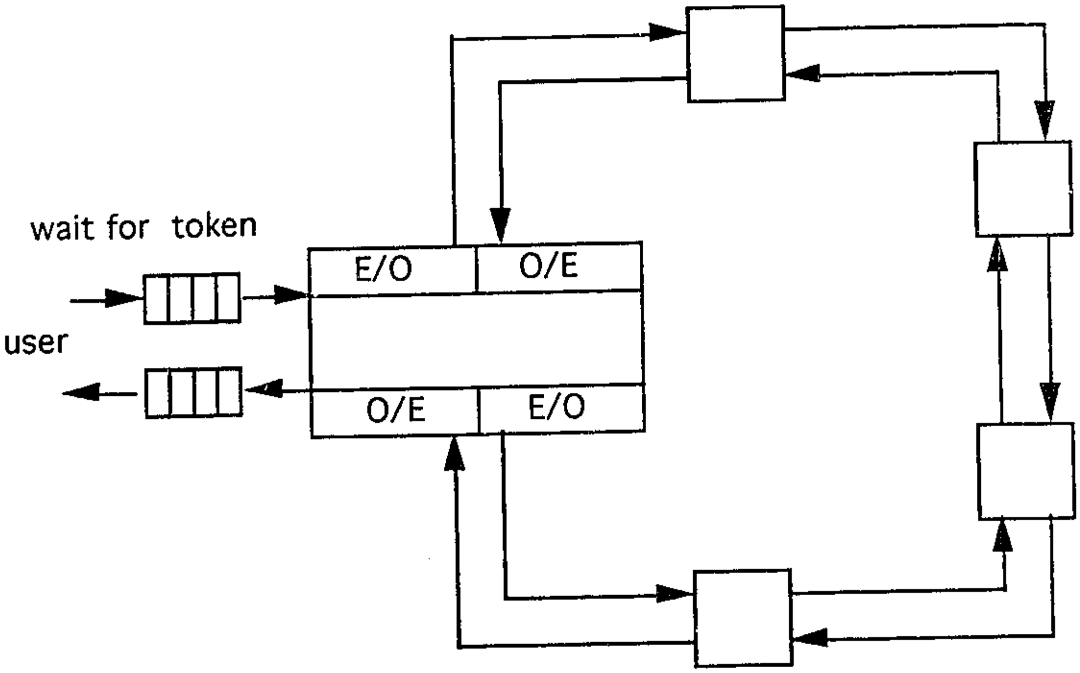

Figure 1.2: Fibre Distributed Data Interface (FDDI)

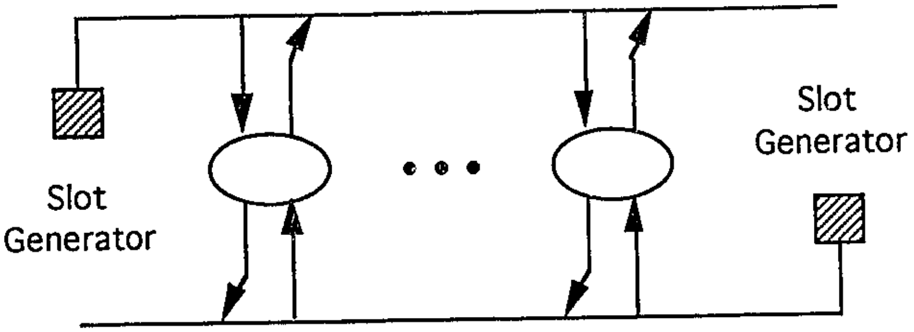

Figure 1.3: Distributed Queueing Dual Bus (DQDB)

The IEEE 802.6 standard has a dual bus architecture capable of supporting speeds up to 150 Mbps on each bus. The frame is generated at the head-ends that continuously sends fixed-length time slots down to the buses. Nodes access the time slots via a global distributed queuing algorithm. The operation of the algorithm is based on a Busy/Idle bit and a Request field in the header of each time slot. This unique MAC protocol is called the Distributed Queuing Protocol which ensures a somewhat fair access to the network by all stations. The timing structure of DQDB allows for integrated synchronous and asynchronous traffic (see Figure 1.3).

Modern MANs such as FDDI already employ transmission over multimode optical fibers based on direct detection and light-emitting diodes (LEDs). Networks can reach speeds in the Gbps range by using LEDs to transmit over single-mode optical fibers. Research proposals for future LANs and MANs are often premised on the emergence of wavelength-division multiplexing (WDM) as a means of increasing the transmission rate from Gbps range to Tbps range in the fiber networks.

1.3 Photonic Network Overview

The advent of single-mode optical fiber has presented communications engineers with the exciting dilemma of a transmission medium which has a bandwidth exceeding both the speeds at which it can be deceased by conventional means, and the aggregate information rates for which it is likely to be used. The low-loss region of single-mode fiber extends over wavelengths from roughly 1.2 to 1.6 μm, which is an optical bandwidth of more than 50 THz. An information capacity of 50 Tbps would be enough to deliver a channel of 100 Mbps to half-million destinations on a single fiber. Optical fiber has become the preferred transmission medium for telecommunications. Photonic technology is now finding applications not only for point-to-point transport, but also in distributed, packet-oriented communication network.

1.3.1 Terabit Capacity Lightwave Network

A schematic diagram for a generic, fully distributed lightwave network is shown in Figure 1.4, where the users are portrayed as connecting to the optical medium

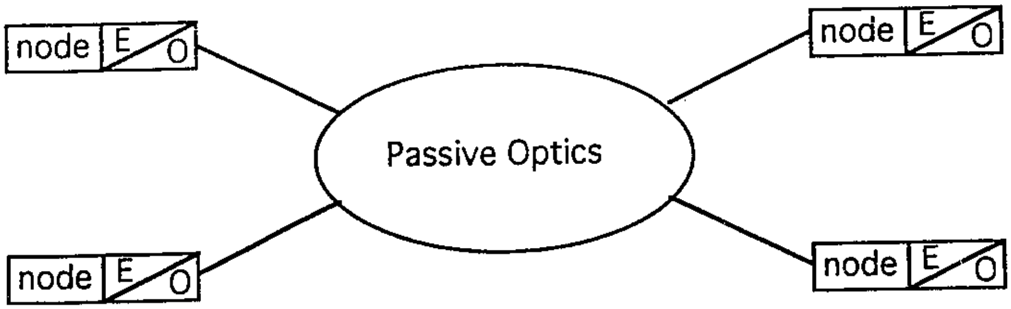

Figure 1.4: Terabit Capacity Lightwave Network

at the periphery of the network. The optical medium is entirely passive, and the only electronics appears in the distributed user access ports. It also contains the vast sea of bandwidth to be accessed, and each user is restricted to some small number of electro-optic transmitters and receivers, each of which operates at a maximum data rate set by electronic/electro-optic technology. A fundamental constraint on the architecture of this lightwave network is the electronic-optic bottleneck, or the inability to electro-optically modulate or demodulate light over a bandwidth greater than a few GHz, which is only a tiny fraction of the optical bandwidth. Thus, the network architecture must invoke some form of concurrence to tap the bandwidth of the medium, i.e., the ability to simultaneously carry a multitude of distinguishable messages. Such architecture have been studied based on wavelength, space, and time division multiplexing ([2]).

1.3.2 Wavelength Division Multiplexing

Wavelength division multiplexing (WDM) ([30]) can offer great concurrence potential. The available optical bandwidth is partitioned into a comb of narrow channels. Each receiver is assigned a unique wavelength, and a transmitter wishing to access that receiver tunes its transmitter to that receiver's wavelength and sends its packet. This approach suffers from two distinct drawbacks. First, to provide complete interconnectivity among all users (i.e., to allow any user to communicate with any other user) requires that either all user transmitters or all user receivers, or both, be able to tune to any of the wavelength channels created within the network. Fast wavelength agility, or the ability to accurately tune optical transmitters or receivers would be needed for a WDM network. Second, pretransmission coordination among all users is required, implying the need for a centralized or distributed network controller. These limitations make the implementation very difficult for MAN with high speed data rate and large number of stations.

1.3.3 Time Division Multiplexing

Time division multiplexing (TDM) ([13] is another way of gaining concurrence in the network. The TDM is generally preferred for sharing an optic fiber. Each optical transmitter is equipped with a source capable of generating a very narrow pulse of bandwidth commensurate with the overall network capacity. However, after generating such a pulse, the transmitter must wait for a prescribed time interval, commensurate with the inverse of the port speed (i.e., 1 nanosecond for a port speed of 1 Gbps) before it can generate another pulse. A mode locked semiconductor laser might satisfy these requirements. The electro-optic conversion speed bottleneck is not overcome since each transmitter is still limited to a peak rate in the order of several Gbps.

1.3.4 Space Division Multiplexing

In Space division multiplexing (SDM), the various concurrent signals are sent to a centrally located distribution point from whence they are routed to their appropriate destinations via a Space Division Switch which maintains input/output paths which are physically disjoint; the spatial paths so provided vary dynamically in response to the changing traffic patterns. Three drawbacks for SDM are considered. First, the number of optic switching elements needed is impractically large to interconnect hundreds or thousands of users for MAN. Second, since the analog switching elements are not precise, signal distortion caused by crosstalk would accumulate. Third, a large, centralized network controller is required to update the switching interconnections in response to time-varying traffic patterns.

1.3.5 Multichannel Multihop Lightwave Networks

The multichannel multihop lightwave network ([30, 24]) is one architecture which is proposed for local and metropolitan based communication networks. The advantage of the multihop architecture over other concurrence schemes is that it does not require rapidly agile optical components or dynamic coordination among the users. The multihop network consists of a physically distributed optical topology, and traffic generating and terminating nodes, each of which has been allocated some small number of transmitters and receivers. Wavelengths are assigned to these transmitters and receivers, creating several independent channels. All of which are wavelength multiplexed onto the optical medium, and defining the logical connectivity among the nodes. For a given source-destination pair a message may have to hop through several intermediate nodes, or equivalently, be transmitted successively over different wavelengths, before reaching its destination.

The logical interconnection pattern of multihop lightwave network consists of several stages connected through a perfect shuffle, which is discussed in Chapter 3 in detail. The key property of the multihop scheme turns out to be the relative independence between the logical interconnection pattern among nodes and the physical topology or fiber layout. Examples of physically distributed topologies are the bus topology, the star topology, the tree topology, and all combination of these topologies.

The multihop solution offers lower transmitter and receiver cost at each station and lower control overhead, but its broadcast and real-time traffic support are inefficient. In view of these limitations, hierarchical solutions that would combine the benefits of the two approaches are now considered in this thesis.

1.4 Future Broadband ISDN

It is commonly recognized that MANs will be forerunners of the wide area broadband networks. In recent years there has been a significant amount of interest in applying the new and developing photonic technology in telecommunications switching systems. This has been viewed as being increasingly more important as the telecommunications industry is anticipating broadband capabilities such as B-ISDN

([27, 8]).

The evolution from MAN towards B-ISDN will consist of several evolutionary steps driven by current service demands, advances in technology, and progress in standards development. FDDI and the IEEE 802.6 DQDB standard, which has a dual bus architecture capable of supporting speeds up to 150 Mbps on each bus (see Figure 1.3), are emerging standards for high-speed (45-150Mbps) MANs.

ATM has been defined by CCITT as the target transfer mode for B-ISDN. The MAN seems to be the most likely solution prior to implementation of the full ATM goal network. A proposed network, shown in Figure 1.5, LANs, MANs, and ATM switches will be integrated into the B-ISDN network. This B-ISDN network has a hierarchical architecture. The B-ISDN node consists of ATM switches which are the backbones to connect several MANs to form a Wide Area Network (WAN). From the B-ISDN point of view the IEEE MAN reduces the load offered to the B-ISDN, provides forward compatibility with B-ISDN, support broadcast and multicast connections, provides high utilization of available bandwidth, and permits automatic reconfiguration around transmission link failures and node failures. From the IEEE MAN point of view, the B-ISDN offers fast MAN interconnection at high data rates and provides functions such as billing and accounting.

One challenge for network designers and engineers who aim to interconnect MANs is always speed mismatch, which may lead to congesting at the gateways. Another challenge is the design of gateways capable of switching hundreds of Mbps.

1.5 Organization and Contribution

A topology design for MAN is presented in Chapter 2. Several existing topologies such as bus topology, ring topology, star topology, and tree topology are reviewed. The comparison of their reliability are discussed. Then hierarchical configuration such as tree-bus (Treenet) topology and star-bus topology are studied.

Chapter 3 presents a photonic network protocol design. A logical multihop lightwave interconnection network, Shufflenet, is introduced. The three hierarchical design models using ShuffleNet for Treenet are introduced.

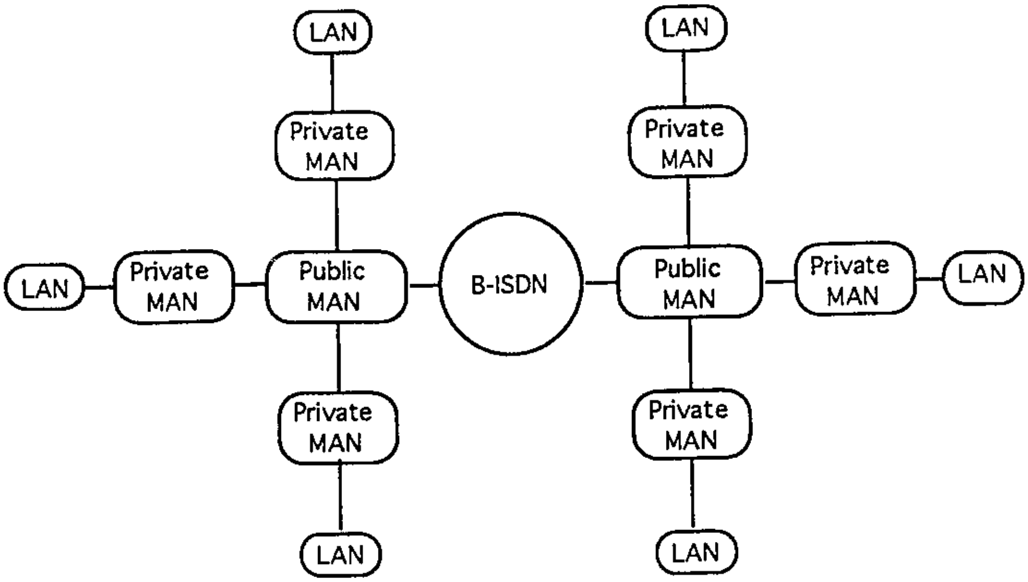

Figure 1.5: Future Broadband ISDN Network

In Chapter 4, the performance of the photonic Metropolitan Area Network is analyzed. Queueing models for Shufflenet and the three design models are presented. Analysis results such as delay and throughput are compared.

Simulation analysis are conducted in Chapter 5. The simulation results, such as queueing delay and system throughput, are presented and compared. The summary of the research work is also given in Chapter 5.

Concluding remarks for this metropolitan network design implementation and further suggestions are given in Chapter 6.

The simulation program written in Q-NAP simulation package will be provided in the Appendix.

The main contributions of this thesis are summarized as follows:

- Study of a new TreeNet topology for Metropolitan Area Network
- Implementation of the multihopping logical technology to the TreeNet topology

Chapter 2

Metropolitan Area Network Topology Design

2.1 Introduction to Today's Network Topology

Today's MAN was necessary to employ interconnection speeds of 100 Mb/s, geographical area coverage of up to 10 km, and network capabilities that would support hundreds of stations. The physical means to this end was optical fiber. Additional requirements are high availability and fault tolerance, and growth flexibility.

These requirements guide us to design the physical topology of MAN. Under the geographical location of terminals and other message sources, where shall the concentration and switching points (the network nodes) be located? How should they be connected? What form should the network take - star, tree, bus, ring, mesh, etc.? How many links (trunks or connections between nodes) are needed? The recommendation on which physical topology is best suited for the future network is dependent upon facilities (physical size, complexity, bandwidth, etc.). The target topology should be selected independently of today's existing network. The topology selection process should include alternative topology economic and technical comparisons to provide the best overall network to capture feature revenue opportunities. Once an optimal topology is selected it is then appropriate to determine how, and to what extent, to use existing network equipment to attain that selected topology.

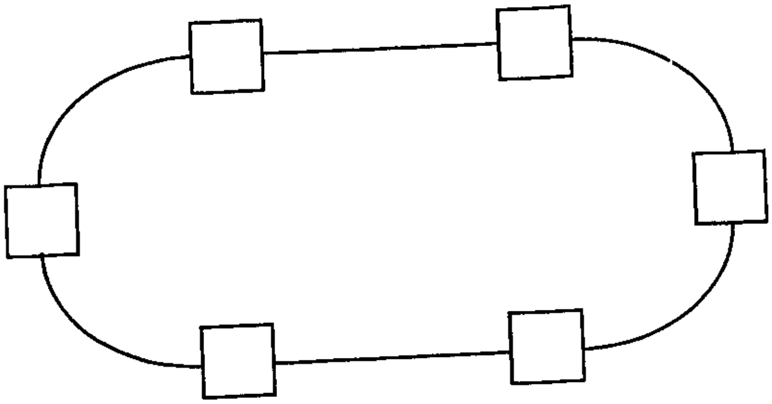

Figure 2.1: Ring Topology

2.2 Basic Physical Topologies Overview

There are four basic topologies which are the building blocks for MAN. They are: ring, bus, star and tree. They will be described in the following section. The comparison of them will be discussed.

2.2.1 Ring Topology

Figure 2.1 shows the ring topology ([36]). Ring topology has nodes interconnected in the shape of a ring. When data is transmitted, it is sent from node to node in one direction. Ring topology appears to be initially easier to deploy well suited to fiber optics, requires lower initial investments, is easier to expand. Ring implementations require active repeats and substantial logic working at channel speed (e.g., address recognition and flag setting) at each station. Their cost and reliability, in spite of some failure node proposals, may set a limit to the use of ring interfaces at very high speed. The ring topology has long been a favorite structure for MAN's. FDDI ([37, 36, 21]) is an example of ring-structured networks.

2.2.2 Bus Topology

Bus topology ([39, 9]) is formed by connecting network nodes to a single transmission media. The bus cable is centrally routed through an area with smaller cables

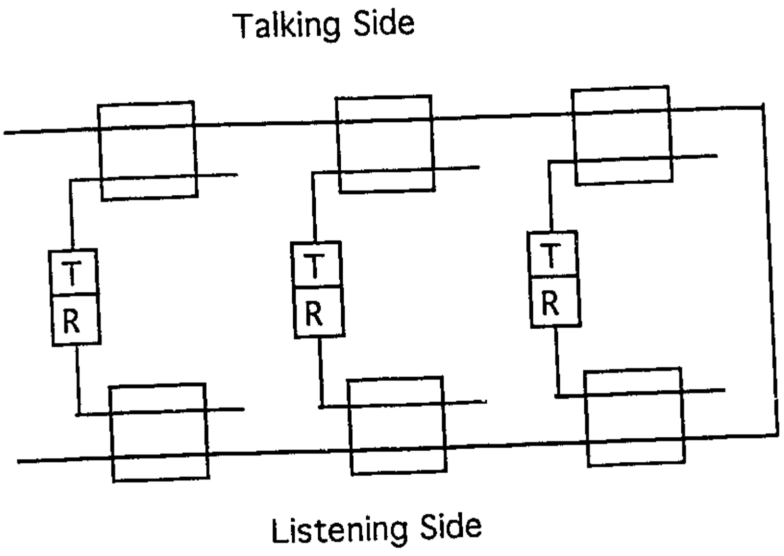

Figure 2.2: Bus Topology

from the main bus to the actual network nodes. Nodes also may be added by tapping directly into the bus. Each node listens to the cable and waits to transmit its data when the cable is quite or idle. Data that is transmitted by one node is simultaneously received by all other nodes on the bus. Data is ignored by all except the node to which it was addressed. The bus topology is based on a linear bus topology shown in Figure 2.2. Stations are coupled via passive interfaces. Each station's traffic is first injected into the "talk side" of the bus, via directional couplers, then broadcast to the receivers on the "listen side", via a second set of couplers. Since optical couplers are intrinsically unidirectional, a unidirectional bus system (UBS) must be used. Within the family of unidirectional bus structures, we may distinguish two classes: the token schemes (Express-net, D-net, and U-net), and the random access schemes (Ethernet).

2.2.3 Star topology

The first basic configuration to implement a photonic network is star topology ([34, 12, 18]). Star topologies can be built by connecting several stations via a star coupler (see Figure 2.3). A star has a major advantage in that the outlying nodes can

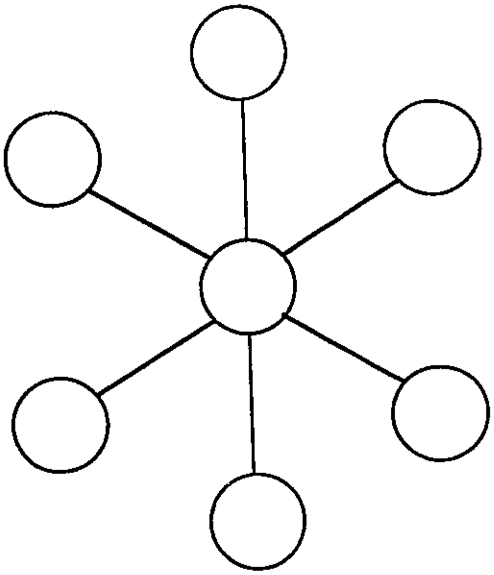

Figure 2.3: Star Topology

be connected to the network through very simple interfaces without any address recognition or repeater circuitry, since the nodes can assume that any message appearing on their dedicated link was intended for them. Furthermore, a star topology can offer some security since all communications between nodes are subject to approval and supervision by the central node. Star configured networks have been implemented using either passive or active components.

2.2.4 Tree Topology

The tree topology ([16, 6, 10]) shown in Figure 2.4 can be viewed as improved star type system in which cabling requirements are reduced by using repeater nodes to replay messages to and from other nodes. The tree configuration differs from the star configuration is that the number of stations in the network should be a power of 2. The advantage of a tree over a star is that some nodes can still communicate with others even though the central node is failed. As in all topologies involving repeater nodes, the failure of a repeater can cripple all or part of the network. The usual countermeasures include robust, highly reliable repeater designs which "never"

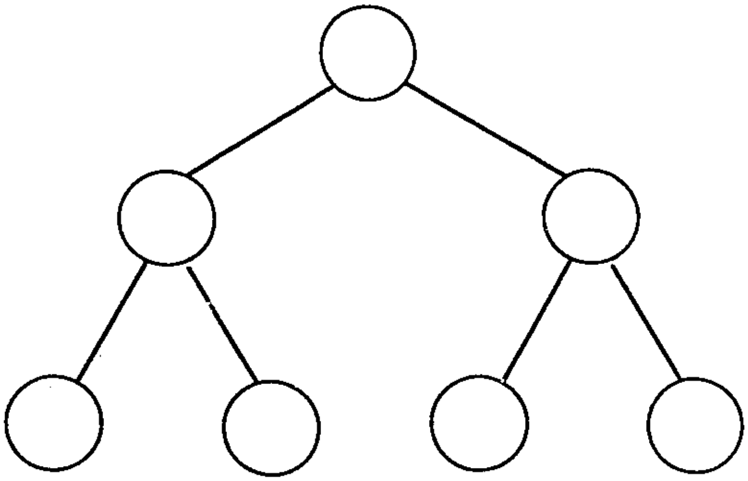

Figure 2.4: Tree Topology

fail, fail-safe bypassing circuit to automatically bypass failed repeaters, or extra links which can be used to establish alternate paths to all functioning nodes after a repeater fails. The tree topology is particularly well suited for this approach.

2.2.5 Comparison between the basic topologies

Ring, Bus, Star and Tree are the most popular LAN and MAN topologies. In fact, each type of topology is capable of adapting to different applications. In this section, we will compare these four topologies. The optimum topology is being reviewed which can best accommodate the demands of a Metropolitan Area Network.

Studies on the advantages and disadvantages of ring and bus networks do not draw a conclusion favoring one or the other. Here we will present the advantages that a bus network compared to the ring network.

- A bus is generally more reliable because usually the station-to-cable connection is passive.
- A bus induces much less propagation delay than a ring because each ring station may have to examine the address field of a message to determine whether the

message is to be taken off the ring (or else the message would go around the ring endlessly) or retransmission to the next station.

- A bus has a better distributed failure recovery characteristics.

A drawback of the star and tree topologies compared to the bus and ring topologies is that it may result in higher wiring costs. However, star and tree topologies have the following advantages over other topologies: ready suitability for optical fiber based implementations because of the point-to-point links from end nodes to the central hub, high achievable throughputs, and simpler end stations. In comparing with star topology tree topology offers some favorable advantages ([14]):

- saving of fibers
- better usage of the fiber transmission capacity.
- the transmitter and receiver in the exchange is shared by several subscribers (coat reduction).
- during the introduction period additional subscribers can be connected to the network with little effort.
- some nodes can still communicate with others even though the central node is failed.

Disadvantages of tree networks are the more complicated transmission technique and the power budget restrictions mainly due to the power splitting factor of the couplers.

Since every topology has its own advantage over other topologies, we can combine several basic topologies together to form a hierarchical architecture. For example we can have Tree-bus and Star-bus shown in Figures 2.5 and 2.6.

The hierarchical architecture is essentially an architecture with two levels of nodes in which distributed node is used at the lower level and centralized node at the upper level. The MAN hierarchical architecture presented here offers a hybrid architecture which merges and perceives the advantages of both. The central node provides

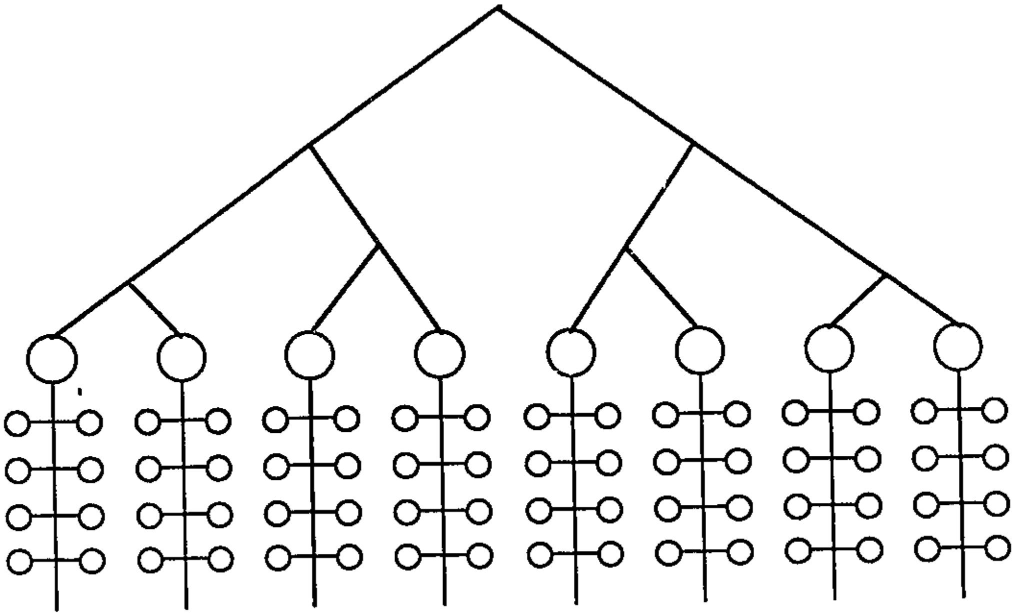

Figure 2.5: Tree-Bus Topology

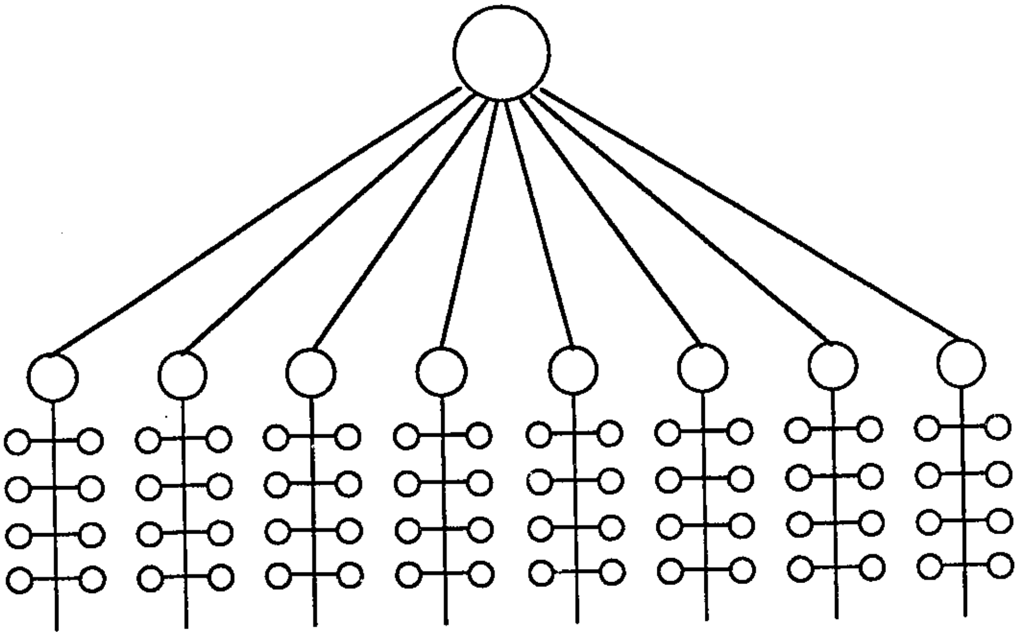

Figure 2.6: Star-Bus Topology

fast interconnectivity between networks, removes the bandwidth limitations of shared medium architecture and ensures security. The shared medium approach efficiently utilizes the available bandwidth, provides mechanisms for congestion control, supports broadcast and multicast services, and reduces the load offered to the central node.

In this paper, we propose a hierarchical architecture: Treenet which can be used in MAN applications and can have some advantages of both tree and bus topologies.

2.3 TreeNet: a Hierarchical Configuration

A Tree network has an advantage of dynamic expansion to meet future growth. In the tree network, clusters of communities of interests can be formed. Depending on the capability of intermediate nodes, forming these clusters could increase the number of concurrent transmissions and the aggregate throughput of the network. Clusters can be formed by connecting stations with a high percentage of intertraffic to the same intermediate node. Depending on the traffic flow among clusters, the tree may be arranged to have the proper ancestor of some clusters at a lower level of the tree.

In a tree network, different routing possibilities exist in establishing a connection between two stations. With simple intermediate nodes, packets from leaf stations may be forwarded on uplinks to the root, then broadcasted on downlinks to all the leaf stations. With more intelligent intermediate nodes, it is possible to have a packet travel up until the proper ancestor of its source and destination is reached. Then the packet may either by broadcasted on all downlinks or routed through the appropriate links to its proper destination. These different approaches have their own merits and demerits in terms of performance and cost of implementation. Generally, with simple nodes, the higher packet delay and lower aggregate network throughput can be achieved. With more complex nodes with switching capabilities, generally higher aggregate throughputs and lower average packet delays can be expected (because of increased concurrence of transmission), but the overall cost could very well be higher.

Several previous papers discussed about Treenet. In the following sections, we will review the Treenet in the literature. Then we will propose a active Treenet as

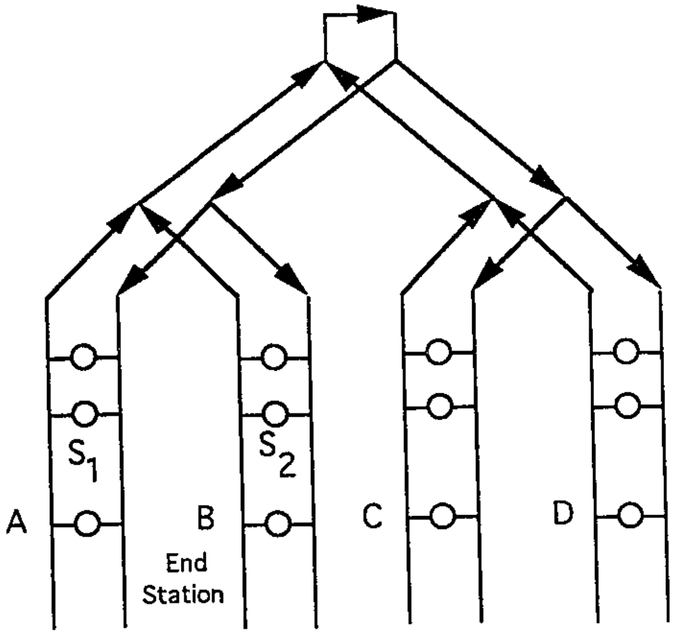

Figure 2.7: Passive Treenet

the topology for MAN.

2.3.1 Review of the Treenet

In the simplest implementation of Treenet Gerla and Fratta presented ([10]) (see Figure 2.7), no active components are present on the network path connecting any two stations, thus protecting the system from active component failures. This also implies that the signal transmitted by one station is broadcast to all other stations, with no filtering nor store-and-forward processing at any intermediate node (i.e., gateway). This permits to operate the network at very high aggregate data rate, without suffering bandwidth limitations imposed by the gateways.

Passive taps, full broadcasting and very high aggregate data rates are well advertised advantages of linear bus networks. While sharing these advantages, Treenet offers additional features which permit to overcome some of the traditional drawbacks of the linear bus architecture. First, Treenet extends the number of stations that can be supported by an order of magnitude (from tens to hundreds, say). Secondly, the tree topology is better suited to cover a large geographical area (campus, industrial park, metropolitan area, etc.) than a linear topology. Thirdly, in the tree structure, the problem of transmitter and receiver calibration is simpler than in the linear bus. Finally, the optical couplers used to build the tree are simple 3-dB couplers, while the couplers in the linear bus may need to be "tuned" depending on their position on the bus (i.e., the coupling ratio is adjusted to optimize the power budget). All the above advantages do not come, of course for free. Treenet has also some drawbacks. A token protocol is used for packet transmission proposed by Gerla and Fratta. That is, the end station in the first bus issues a token, upon detecting the token, the station can transmit packet. Since the token is shared by all the stations in the Treenet, the delay is very large (the delay being equal to the cycle time taken by the token through the tree). To overcome the drawback of large latency delay for passive Treenet, we propose an active Treenet which has active nodes between the tree and the linear buses.

The paper presented by Ibe and Chen ([16]) describes the architecture and performance of Treenet. The users are divided into geographical subgroups, and any two subgroups are assigned a fixed set of channels. Different types of traffic are carried on different channels and access to the network is made a distributed manner. The control channel is shared by all users. The transmitting user first sends a control packet over the control channel, consisting of the transmitter address bits, the user address bits and the wavelength to be used for data transmission. The data channel is set up after the control packet is received. There are two drawbacks of this proposed Treenet. One is wavelength-agile transmitters or receivers are required. Another one is that pretransmission coordination between two users wishing to communicate is also required. Rapidly-tunable, wavelength-agile optical transmitters and receivers have not yet emerged from research laboratories. The complexity and associated delay of the pretransmission coordination protocols make especially cumbersome at the high rates at which we envision.

2.3.2 Active Treenet Architecture

To overcome the drawbacks of the above Treenets, we present the active Treenet in the section. Figure 2.5 shows the active Treenet. It is similar to passive Treenet except there are active nodes between the tree and the linear buses. Each bus can be viewed as a local area network. All the buses are interconnected together by the active nodes to form the tree. The active nodes are used for traffic filtering. Its function is to inspect every packet coming up from the leaves and to extract and forward to other active nodes the packets with a "foreign" address. Each packet has the destination address on the header. If the packet is local traffic in the same bus, the packet will be transmitted to the active node on the top of this bus, then directly down to the destination station of this bus. One advantage of the active Treenet over the passive Treenet is the traffic intensity on the top tree is reduced since the local traffic is not broadcasting to all other buses. The local traffic can still go through even though the link or node on the top tree is failed. Another advantage of active Treenet is that the average delay is improved. The disadvantage of using active Treenet is that the system reliability is reduced since the active components are easy to break.

In this thesis, our protocol is based on Wavelength Division Multiplexing (WDM) technology. We will show in chapter 3 that active Treenet using WDM can overcome the drawbacks of the Treenet in the literature.

2.3.3 Extension of the Treenet Topology

A tree network has an advantage of dynamic expansion to meet future growth. Whenever a new station is added, it is connected to the nearest bus. Whenever a active node is added, its main links are connected to the tree and a new bus is created to connect to this active node. Links are chosen so as to minimize the physical length of both tree and bus. Two approaches are presented to form a Treenet in a metropolitan area.

First, no planned expansion is required. Here every client is given a new access node and connected to a branch closest to it. Second, a network grows is planned stages to meet the anticipated development of a metropolitan area. A new client is connected to the nearest MAN node owned by the network company. Rearrangement of the network is significant improvement of network performance such as reliability, traffic handling capacity, saving of address space, material cost and other cost factors ([33, 26]).

The coverage of a metropolitan area may require several Treenets to improve the total number of stations. A very effective way is interconnecting several Treenets with modular star, as shown in Figure 2.8 described as a multilevel categorization of the high-speed metropolitan area hierarchy. At bottom level (level 0), we include the user premises stations, terminals and local area networks. Use-base bus LAN is typically employed. The second lowest level (level-1) network hierarchy provides a backbone for the interconnection of the level-0 user networks; it also serves as a subscriber local access network used to higher level communications networks and nodes. Here, the level-1 is the active nodes serving as bridges or gateways. Level-2 network architecture can be characterized in a manner similar to that used for level-1 networks. A level-2 network can provide interconnection and access to level-1 networks through the use of proper pathways and backbone networks that are configured as a star used to interconnect several Treenet; such a star configuration must be properly backed up by other star nodes to provide the desired availability and reliability features. Finally, Tree-Nuts can be connected to each ether or to a Wide Area Network via gateways (level 3).

2.4 Summary

We have proposed a family of fiber optics topologies for HSLAN and MAN implementation. The advantages and disadvantages of each topology are compared. The TreeNet topology is considered the best topology for MAN. It can easily support a large number of stations and cover a large geographical area. The tree topology is more reliable than star topology.

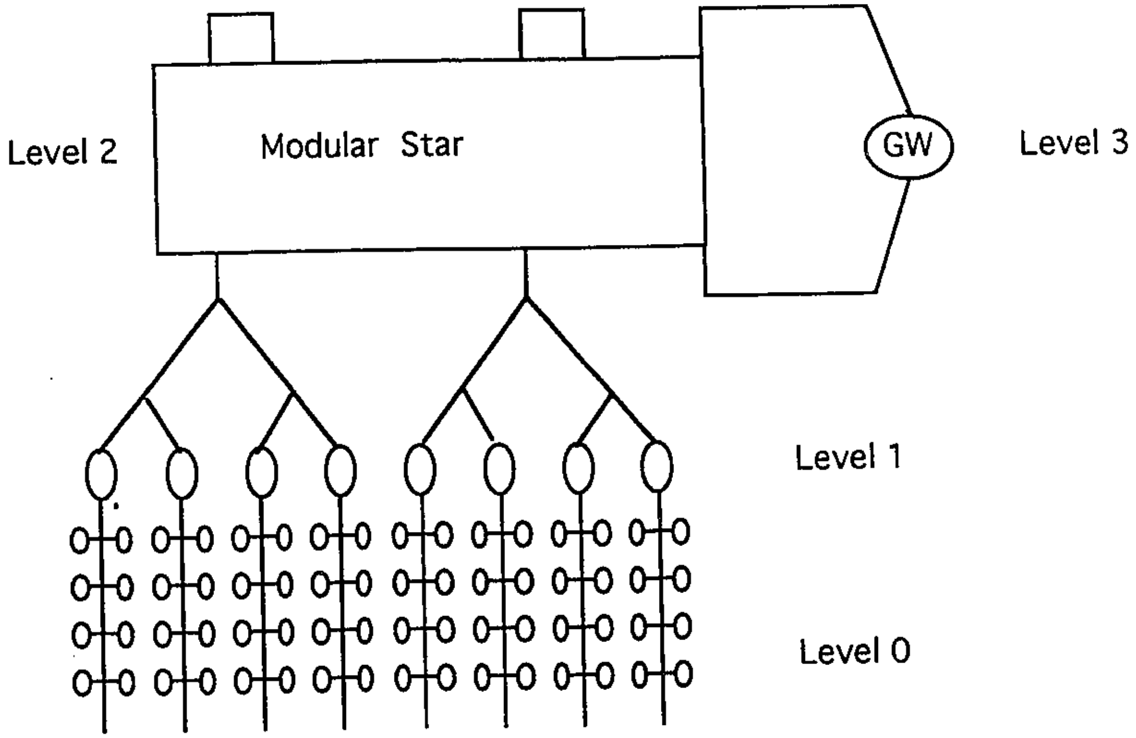

Figure 2.8: Extended Treenet

Chapter 3

Protocol Design for Treenet

3.1 Introduction and Motivation

In Chapter 2, the Treenet was proposed as a physical topology for Metropolitan Area Networks. Physical topology means the actual underlying network topology. On any underlying physical topology, one can impose a carefully selected connectivity pattern that provides dedicated connections between certain pairs of users. Traffic destined to a user that is not directly receiving from the transmitting user must be routed through intermediate users. This overlaid topology is referred to as the logical topology. The term *multihop* is used to refer to any network where traffic may have to be routed through intermediate users. The multihop network is considered as a logical topology protocol in this thesis.

There are several advantages to use a logical topology on top of a physical topology:

- It may be possible to create the logical topology to suit the traffic patterns in the network.
- Although the physical topology may be arbitrary, the logical topology can be made regular, so that routing and flow control are simpler and traffic balancing is improved.

Here, the logical topology can be implemented using only fixed-tuned transmitters and receivers. In the absence of a logical topology, one must use tunable transmitters or receivers to provide direct any-to-any connectivity between users in the network. The technology on these tunable components is less mature than that of their fixed-tuned counterparts. More importantly perhaps, one avoids the problem of dealing with collisions in the shared-medium topology consisting of point-to-point links over the broadcast medium.

In order to create a logical topology over a physical Treenet, a multichannel multihop lightwave networks called ShuffleNet ([15, 40, 4, 3]) is proposed. Multichannel multihop lightwave networks are optical-fiber-based packet communication systems for multiuser applications, such as LANs and MANs. In the next section, we will use a multichannel multihop network as a logic topology to the Treenet.

3.2 A ShuffleNet Architecture

3.2.1 Network Description

In a multichannel multihop lightwave network, packets destined from one user to another may be routed via intermediate users in a sequence of "hops" on the fiber, each hop using a different fiber channel. A user (node) in such a network has a small fixed number of incoming and outgoing links, independent of the total number of users. Users interface to the optical medium through network interface units (NIU).

The ShuffleNet is a multichannel network that may be applied to the Treenet. It is an attractive logical topology for Treenet physical topology and provides a regular topology with simple addressing and self-routing. Here, both transmitter and receiver lasers are fixed. ShuffleNet avoids two serious drawbacks of standard multichannel approaches:

(1) the requirement of wavelength-agile transmitters or receivers.

(2) pretransmission coordination between two users wishing to communicate.

An example of the ShuffleNet is shown in Figure 3.1, which is drawn for an 8-port ShuffleNet bus topology. Sixteen WDM channels are created within the fiber bus. The users interface to a unidirectional multiwavelength optical bus through a set of Network Interface Units (NIUs), each having two fixed wavelength transmitters

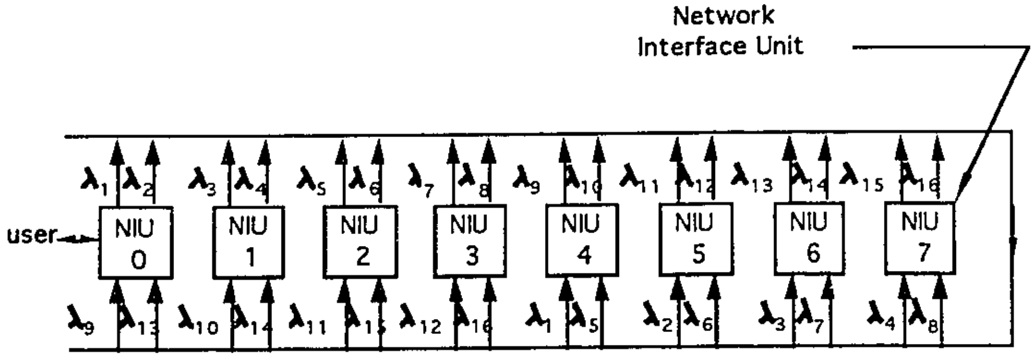

Figure 3.1: 8-Port ShuffleNet Bus

and two fixed wavelength receivers. A packet can go from any input to any output port in a path of three "hops" or less. The NIUs functionally appear as two-input two-output electronic packet switches. Thus, a packet from the input port at NIU-1 desiring to go to the output port at NIU-3, may first be transmitted on to NIU-6 and then to NIU-3. It is shown in [10] that there is a routing algorithm delivering packets from a given source to any given destination in the minimum number of hops possible for that source/destination pair. The routing algorithm is attached in Appendix A. A feature of the ShuffleNet architecture is that there are several paths between each pair of input and output ports. In the above example, if the illustrated path was congested, an alternate path of hops from NIU-1 to NIU-3 could be used, for example, transmitting to NIU-5, then NIU-2, then NIU-8 and finally to NIU-3, while this is not a minimum path, it is still possible to get to output port at NIU-3, even if congestion (or failure) made communication through NIU-6 impossible.

Figure 3.2 shows the connectivity graph associated with the example of Figure 3.1. The eight NIUs are placed in two columns of four NIUs each. Each NIU has two fixed wavelength transmitters and receivers. The assignment of transmit and receive wavelength is such that with the right side of the second column as if the entire graph were wrapped around a cylinder. In general, the ShuffleNet connectivity graph consists of $N = kp^k(k = 1, 2, \cdots; p = 1, 2, \cdots)$ NIUs in k columns of p^k NIUs each. Moving from left to right, successive columns are connected p^{k+1} by directed arcs,

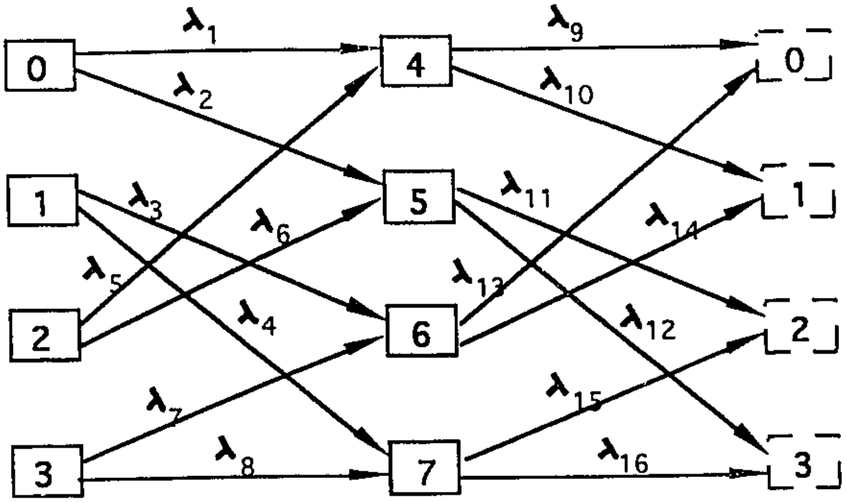

Figure 3.2: 8-Port ShuffleNet Connection

arranged in a fixed ShuffleNet pattern, with the last column connected to the first as if the entire graph were wrapped around a cylinder. Each of the p^k NIUs, in a column has p arcs directed to p different NIUs in the next column according to p-Shuffle interconnect pattern. Figure 3.3 shows the connectivity for an 18-port ShuffleNet ($p = 3, k = 2$).

Figure 3.4 shows a functional block diagram of a two-transmitter, two-receiver NIU. There are three inputs. The first one is the electronic interface to the end users. The other two are optical, and correspond to the two channels to which the NIU is permanently attached. The incoming optical signals are converted to baseband electrical signals, and together with the generated user data, are processed by a three-by-three fully interconnected electronic switch. Although the data rate of all electronic signals is the channel speed (e.g., 1 Gb/sec), the bits can be demultiplexed and processed in parallel by slower speed electronics. The destination address in the header of each arriving packet determines to which of the three outputs the packets is to be routed. Since multiple packets may arrive on different inputs destined for the same output, each of the three outputs is served by a shared buffer that stores arrivals on a first-in-first-out basis. Two of these outputs drive optical transmitters to

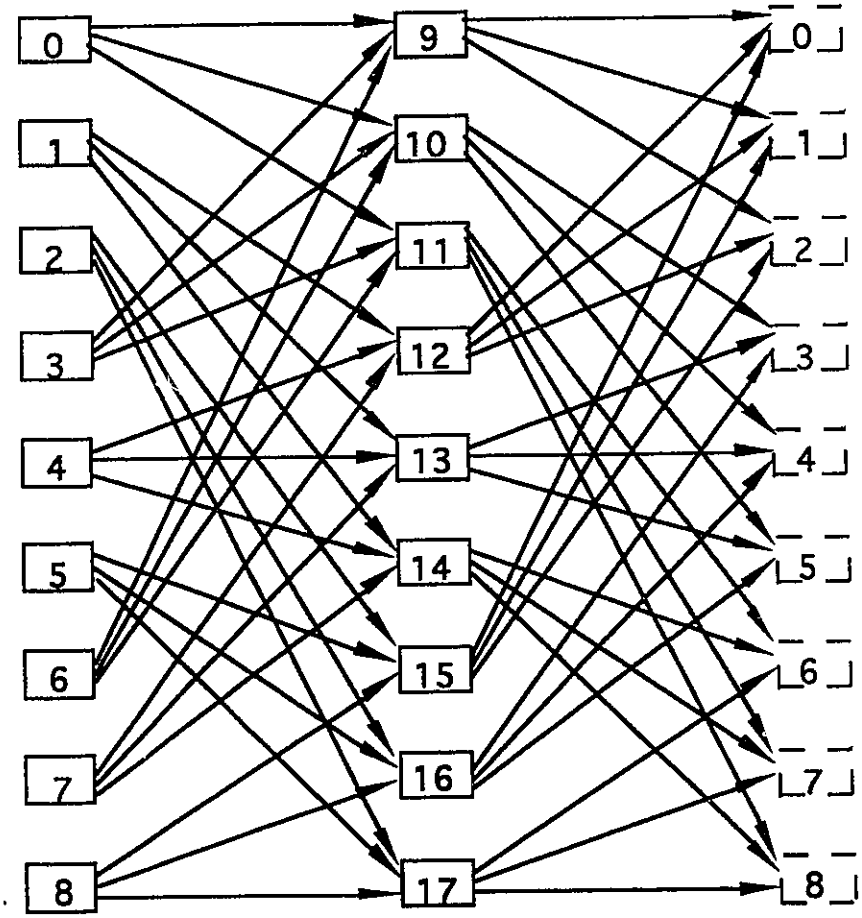

Figure 3.3: 18-Port ShuffleNet Connection

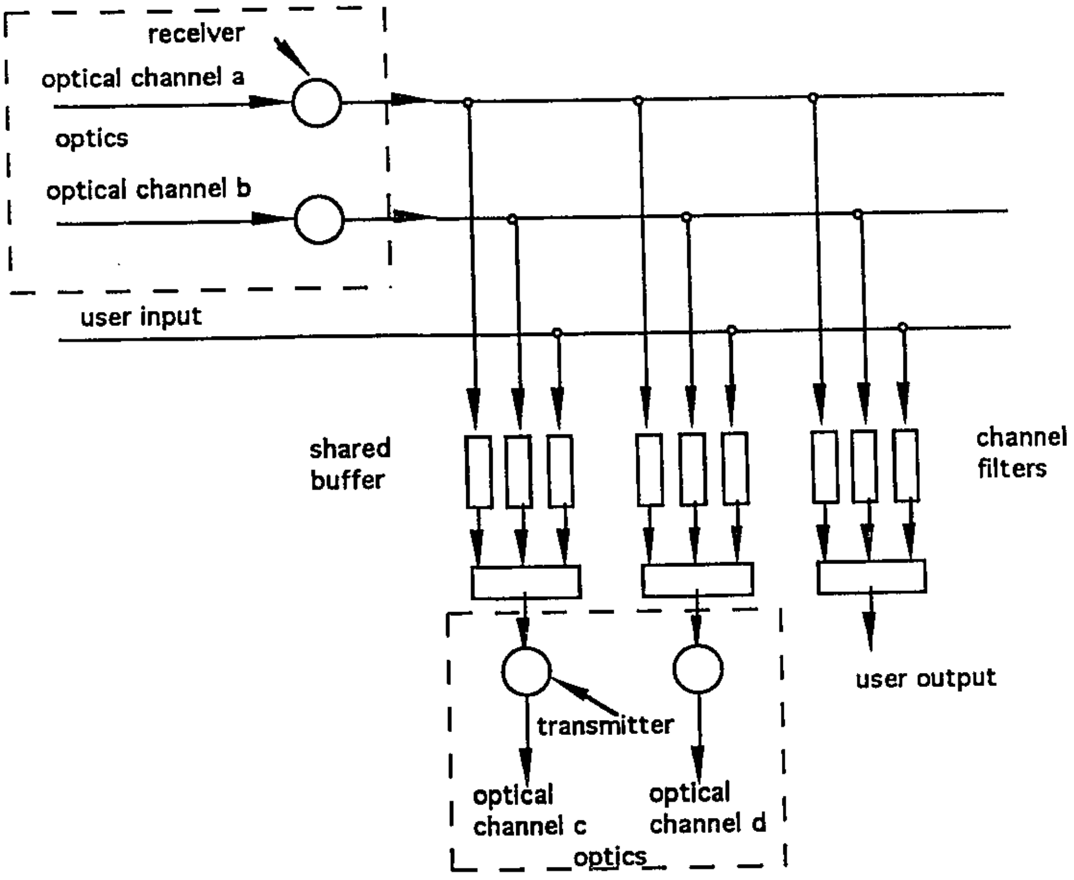

Figure 3.4: Block diagram of two-transmitter, two-receiver node

place the signals on the appropriate channels of the transmit bus. The third output goes directly to the end user.

3.2.2 Network Capacity for ShuffleNet

In the multihop environment, only a portion of the network capacity is used for newly generated traffic. A certain amount of network capacity is taken up by regenerated traffic as packets hop from one node to another in order to get to their destinations. The overall capacity of the network C is inversely proportional to the average number of hops that a packet takes to get to its destination. Thus,

$$C = \frac{1}{E[\text{Hops}]}$$

Since there are N nodes and each node has two links running at speeds of C bits/s, the total network capacity is given by:

$$\gamma = \frac{2NS}{E[\text{Hops}]}$$

It is necessary to calculate the average number of hops from a source to a destination. The minimum-hop path from a given source to each destination in the network can be represented by a directed spanning tree routed at the source. Figure 3.3 illustrates an 18-NIU ($p = 3, k = 2$) ShuffleNet that one might use to assign routes to packets generated by user one destined for the other users in the network. The routing of packets in the network can be specified by a set of N such spanning trees, one routed at each NIU in the network.

It is not enough, however, to select minimum-hop routes in the network. The capacity of the network may be unnecessarily limited if the routing is not selected to load balance the traffic on the WDM channels. That is, many minimum-hop routing paths may pass through the same WDM channels, making it a bottleneck in the network.

Note from Figure 3.3 that the depth of the spanning tree (i.e., the maximum number of hops from a source to a destination) is $2k - 1$, which is proportional to approximately the logarithm of the number of NIUs. More specifically,

$$\text{Number of NIUs } h \text{ hops from source} = \begin{cases} p^h & h = 1, 2, \cdots, k-1 \\ p^k - p^{h-k} & h = k, k+1, \cdots, 2k-1 \end{cases}$$

so that the expected number of hops between two randomly selected NIUs is given by ([4]):

$$E[\text{Hops}] = \frac{kp^k(p-1)(3k-1) - 2k(p^k-1)}{2(p-1)(kp^k-1)}$$

The channel efficiency is determined solely by the connectivity graph and not the particular implementation, such as whether the channels are dedicated or shared.

$$C = \frac{1}{E[\text{Hops}]} = \frac{2(p-1)(kp^k-1)}{kp^k(p-1)(3k-1) - 2k(p^k-1)}$$

The main concern, however, is not with the channel efficiency, but with achieving concurrence in the network as measured by the network throughput. Here only one performance comparison will be presented to illustrate the large capacity of multihop lightwave networks. For a (p, k) ShuffleNet with dedicated channels (requiring p transmitters and p receivers per NIU), the total network throughput and the throughput per NIU are $C^k p^{k+1}$ and Cp, respectively, normalized to the channel transmission rate. For a 1 Gb/s transmission rate, Figure 3.5 shows the throughput per NIU as a function of the values of p, the number of transmitters and receivers per NIU. Also shown, for comparison, is the throughput per NIU of a single channel network operating at 100% media access efficiency. The throughput per NIU for the single channel network decreases as network. Note that aggregate throughputs in the 100's and even 1000's of Gb/s are achievable with a multihop lightwave network, despite the fact that each NIU is limited to 1 Gb/s.

The multi-hop approach ShuffleNet can be used as the logic topology for bus, ring ([20]), star ([17]) and tree ([11]) topologies. In the next section, the ShuffleNet will be implemented in the tree network. Neither centralized nor decentralized network control is required. Except for the fixed wavelength laser sources located near the user's network interface unit, all lightwave components are completely passive (i.e, fiber and couplers only).

3.2.3 ShuffleNet Implemented in the Tree Network

The tree topology has been presented in the previous chapter. Here the physical tree topology is divided into two classes: the basic tree topology shown in Figure 3.6, and the extended tree topology shown in Figure 3.7. In the next section, ShuffleNet will be used as a logical topology to tree topology, and discùssion will be made on how to implement ShuffleNet io the basic and extended tree topologies separately.

Basic Tree Network

Figure 3.6 shows the basic tree topology, which is a binary tree. The users are located on the bottom of the tree. It is assumed that the tree level is k_1, then the

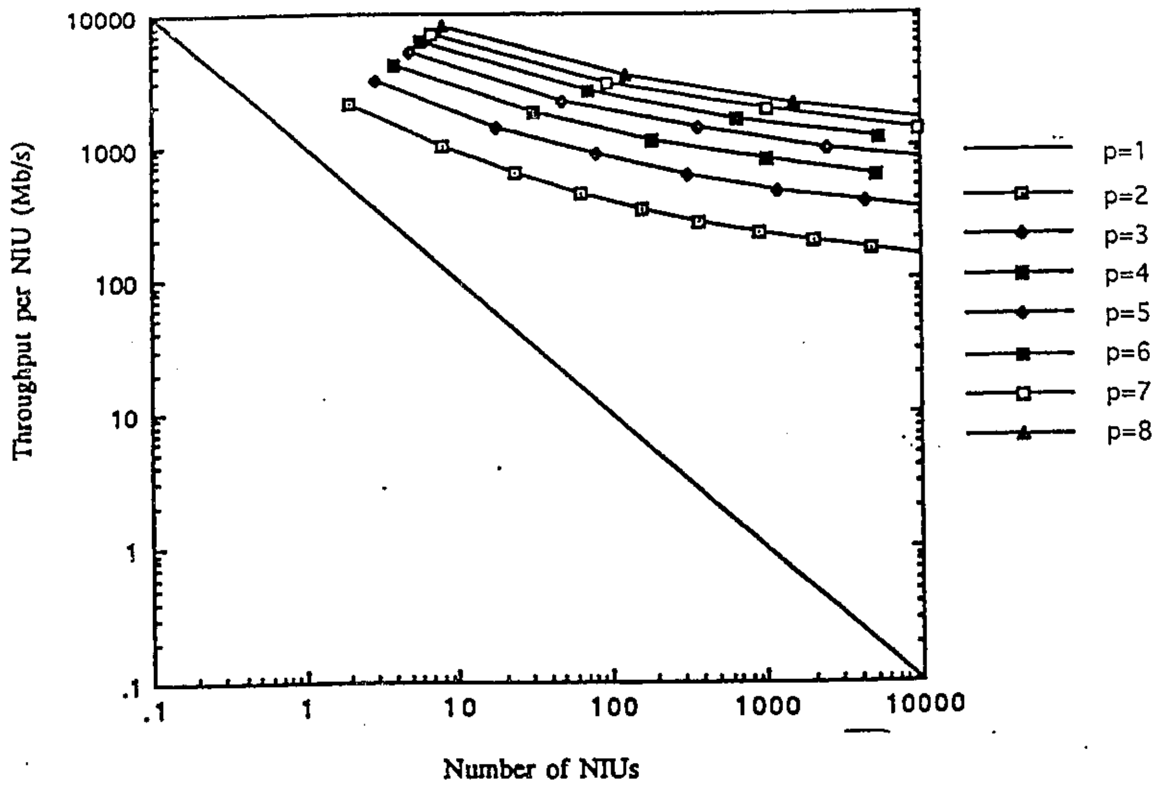

Figure 3.5: Throughput per NIU for ShuffleNet

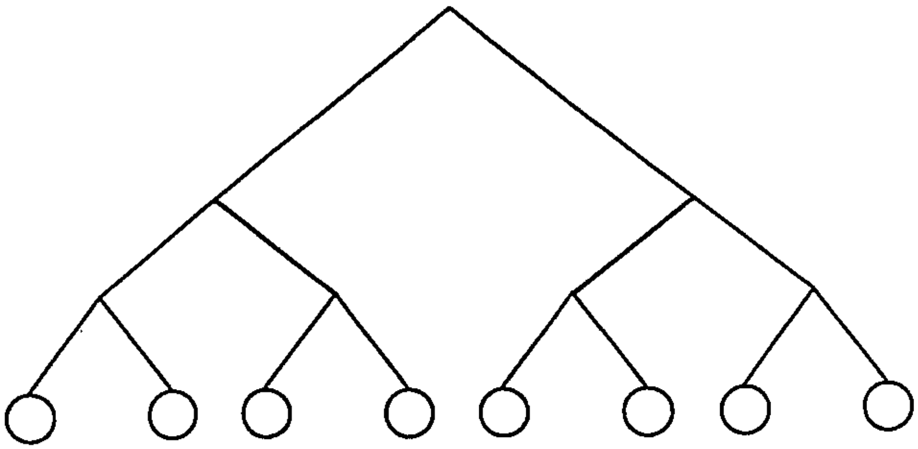

Figure 3.6: Basic Tree Network

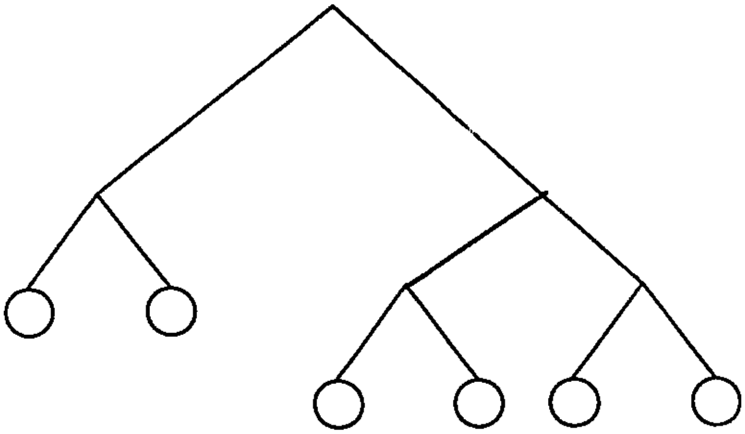

Figure 3.7: Extended Tree Network

Tree Levels (k_1)	1	2	3	4	5	6	7	8
Number of Users	2	4	8	16	32	64	128	256

Table 3.1: Number of users in the binary tree network

number of users on the bottom level is equal to 2^{k_1}, $k_1 = 0, 1, 2, \cdots$. Table 3.1 shows the result of the number of users for the tree topology, and Table 3.2 shows the result of the number of NIUs which is equal to $k_2 p^{k_2}$ for the perfect ShuffleNet ($p = 2$) when k_2 varies from 1 to 8. If the tree topology is given, then the number of users is fixed. The question is how the ShuffleNet can be used to implement the tree topology. There three options described as:

(1) Using ShuffleNet with $p = 1$:

From Tables 3.1 and 3.2, it can be seen that the perfect ShuffleNet connectivity can not match the basic tree network exactly. In order to match the number of users on the bottom level of the tree, a ShuffleNet connectivity called the logic ring ShuffleNet connectivity ($p = 1$) is used to implement the tree topology.

k_2	1	2	3	4	5	6	7	8
Number of NIUs	2	8	24	64	160	384	896	2048

Table 3.2: Number of NIUs in the ShuffleNet

Tree Levels (k)	2	4	8	16	32	64	128	256
Number of NIUs	2	4	8	16	32	64	128	256

Table 3.3: Number of NIUs in the logic ring ShuffleNet

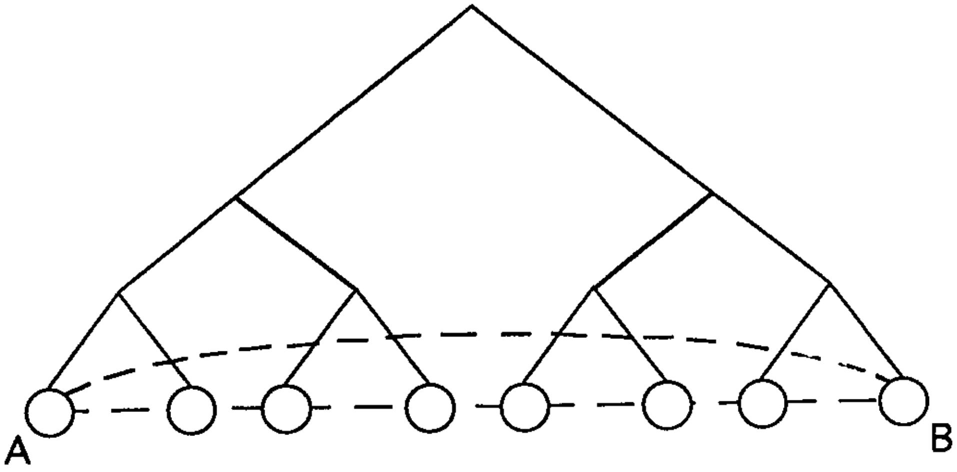

Figure 3.8: Logic Ring ShuffleNet

A logic ring ShuffleNet applied to tree topology is shown in Figure 3.8. A user is connected with the next user using one wavelength. A packet can go from any source to any destination in one hop to maximum seven hops. Table 3.3 shows the number of NIUs for logic ring ShuffleNet for variable k. The logic ring ShuffleNet can be easily implemented in the basic tree network. One drawback of the logic ring ShuffleNet is the poor reliability since there is only one wavelength between two users. Another drawback is the large propagation delay between the transmission from farthest nodes A to B since the maximum hops are required for the packet transmission (see Figure 3.8).

(2) Using Perfect ShuffleNet with dedicated wavelengths:

In the perfect ShuffleNet, each user has two fixed wavelength transmitters and two fixed wavelength receivers. From Tables 3.1 and 3.2, it is known that the perfect ShuffleNet cannot be used for tree network exactly. If the level of the tree is k_1, the number of users for the tree network is 2^{k_1} a perfect ShuffleNet

k_1	k_2	2^{k_1}	$k_2 2^{k_2}$	k_1	k_2	2^{k_1}	$k_2 2^{k_2}$
1	1	2	2	11	8	2048	2048
2	2	4	8	12	9	4096	4608
3	2	8	8	13	10	8192	10240
4	3	16	24	14	11	16384	22528
5	4	32	64	15	12	32768	49152
6	4	64	64	16	13	65536	106496
7	5	128	160	17	14	131072	229376
8	6	256	384	18	15	262144	491520
9	7	512	896	19	16	524288	1048576
10	8	1024	2048	20	16	1048576	1048576

Table 3.4: Using the ShuffleNet for the tree network

can be chosen and implemented in this tree topology. It can be assumed that the number of NIUs for the perfect ShuffleNet is $k_2 2^{k_2}$. Since the number of NIUs in the ShuffleNet should be greater than the users on the tree network, the relationship between the number of users in the tree network and the number of NIUs in the perfect ShuffleNet is as follows:

$$2^{k_1} \leq k_2 2^{k_2} \Longrightarrow k_2 + \log_2 k_2 \geq k_1$$

For a given tree network, the tree level k_1 is given. The choice of k_2 should satisfy the relationship $k_2 + \log_2 k_2 \geq k_1$. Since k_1 and k_2 can only be integers, Table 3.4 shows the result of the number of users in the tree network(2^{k_1}), ShuffleNet connectivity level k_2 and the number of NIUs ($k_2 2^{k_2}$).

The number of NIUs in the ShuffleNet is larger than the number of users in the Treenet. For example, if the ShuffleNet is used to implement a 4-level tree which has 16 users on the bottom, a 24-port ShuffleNet will be used. Within this 24 NIUs, 16 NIUS will be used on the tree network, the rest 8 users will be redundant. The physical connection is shown in Figure 3.9. The 4-level subtree is the main tree. The nodes on this tree are end users. The 3-level subtree is the redundant tree. The nodes on this tree are intermediate nodes, not end users. The 24 nodes on the tree form a 24-port ShuffleNet. Since the 3-level tree is a

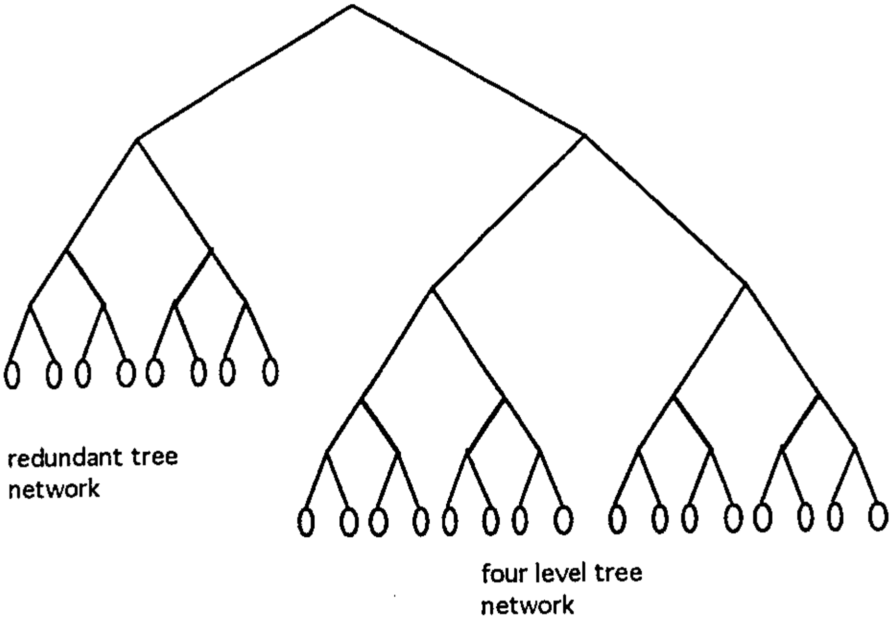

Figure 3.9: 24-Port ShuffleNet for the Tree Network

redundant tree, nodes on the redundant tree can be used to replace nodes on the main tree if nodes on the main tree are broken. The redundant tree can be used to replace one of the subtrees in the main tree if the subtree is down.

The advantage of this option is that there is a redundant tree for the main tree, which increases the system's reliability. The disadvantage is that the increase of wavelengths, transmitters and receivers will increase the cost.

(3) Using Perfect ShuffleNet with Shared wavelengths:

In option (2), the perfect ShuffleNet with separate wavelength is implemented in the tree network. Its drawback is that the number of wavelengths used in the perfect ShuffleNet with separate wavelengths is $2k2^k$, which is large amount of wavelengths. To reduce the number of wavelengths, a perfect ShuffleNet with shared wavelengths can be used, whose connectivity graph is shown in Figure 3.10. It can be noticed that users in the first column are sharing wavelengths λ_1 and λ_2, users in the second column are sharing wavelengths λ_3 and λ_4.

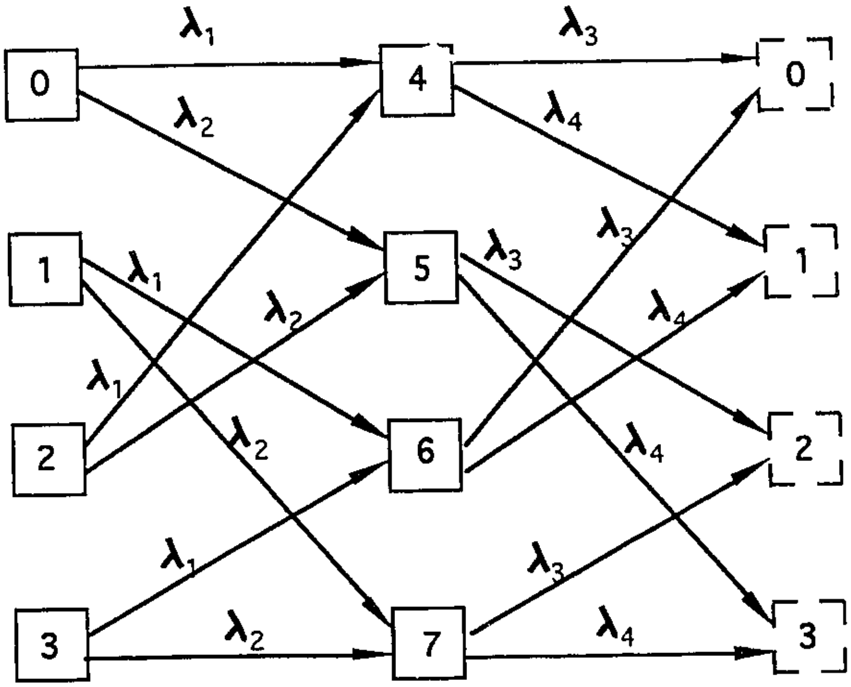

Figure 3.10: 8-Port ShuffleNet with Shared Channels

The number of wavelengths in the perfect ShuffleNet with shared wavelength is reduced to 4 compared with the 16 wavelengths in the perfect ShuffleNet with dedicated wavelengths. Since multiple users are transmitting on a common channel, Time Division Multiplexing Access(TDMA) protocol is used. This gives each user a fixed allocation of the capacity of a single channel.

This approach can be implemented in the tree network. For example in Figure 3.9, there are three 8-node trees. We use 8-port ShuffleNet with shared wavelengths on each tree, so that the total number of wavelengths used is reduced from 64 to 16.

Extended Tree Network

In the extended tree network, the number of nodes is not necessarily a power of 2. In this case, the tree network can be extended to make the number of users a power of 2, so that the perfect ShuffleNet can be used for the implementation. For example in Figure 3.7, the number of users is 6. By adding 2 more users, an 8-port perfect ShuffleNet can be used for its implementation. The two added users are not end users but redundant users to be used only when the other 6 users are broken.

3.2.4 Protocol Design for Treenet using ShuffleNet Interconnectivity

In the previous section, the ShuffleNet has been used as the logic topology for the implementation of the tree topology. In this section, three protocol design models for Treenet topology using ShuffleNet interconnectivity are presented. A Synchronous Time Division Multiplexing (STDM) technology is introduced. The physical topology of Treenet has been shown in Figure 2.5. Separate logical protocols will be used for the upper and lower level. The three hierarchical schemes are as follows:

1) ShuffleNet in the upper level and Synchronous Time Division Multiplexing (STDM) in the lower level;

2) ShuffleNets in both upper and lower levels with dedicated channels between users and active nodes;

3) same as 2) except that there are no dedicated channels between users and active nodes.

STDM

Although the type of framing is often used to discriminate between asynchronous and synchronous transmission, the fundamental difference between the two methods is that with asynchronous transmission, the transmitter and receiver clocks are unsynchronized while with synchronous transmission both clocks are synchronized.

In a STDM system ([13]), each user is assigned a fixed time duration (a time slot) on the communication channel for the transmission of its packets. Each station has a clock. After one user's time duration has elapsed, the channel is switched to another user with a synchronous operation. Packets from each of the users connected to the multiplexer are transmitted in the order of arrival using the full capacity of the transmission line. In order to distinguish among messages going to different users sharing the same line, addressing information for destination must accompany these messages.

It is important to maintain framing synchronization since, if the source and destination are out of step, data on all channels are lost. The most common mechanism for framing is known as added-digit framing. In this scheme, typically, one control bit is added to each TDM frame. An identifiable pattern of bits, from frame to frame, is used on this "control channel". Thus to synchronize, a receiver compares the incoming bits of one frame position to the expected pattern. If the pattern does not match, successive bit positions are searched until the pattern persists over multiple frames. Once framing synchronization is established, the receiver continues to monitor the framing bit channel. If the pattern breaks down, the receiver must again enter a framing search mode.

Shu-STDM

In this model, the ShuffleNet is used in the upper level and STDM scheme in the lower level. Implementation using ShuffleNet in the tree topology has been discussed in the previous section. For example, in Figure 2.5, eight active nodes are physically connected together to form a tree. The logical connection of them is an 8-port perfect ShuffleNet, that is, each active node can transmit and receive two wavelengths with other active nodes. There are buses in the bottom level. The STDM scheme is used on each bus. The end user of each bus issues a frame. The number of slots in the frame equals the number of users in one bus and the slot length equals the packet length. Each user occupies one slot. When a packet arrives at a user, it is stored in the buffer of this user. Then the packet is transmitted to the active node using a fixed slot in the frame. The destination address of the packet is checked by the active node. If the packet is for another user of the same bus after the destination address is checked, it will be transmitted back to the bus by the active node quickly. If the packet is for the user of another bus, it will be transmitted to the active nodes of the destination bus by passing several active nodes of the ShuffleNet. Finally, the active node of the destination bus sends the packet to the destination user. On the receiving side, the receiver of the user receives the packet according to the destination address.

Shu-Shu1

In this model, the upper level consists of a ShuffleNet called outgoing ShuffleNet. This upper level ShuffleNet is the same as in the first model. In the lower level, each physical bus is configured to be a logical ShuffleNet which is called local ShuffleNet. There is a dedicated wavelength for transmitting and one for receiving between each user in a bus and active node on the top of this bus. For example, in Figure 2.5, the logical Treenet consists of a outgoing ShuffleNet and eight local ShuffleNets. Each user in a bus is a node in the local ShuffleNet. When the packet arrives at a user, the destination address of this packet is checked (the destination address consists of the local address and the foreign address). If this packet belongs to the local traffic, it is transmitted to the destination user using the local ShuffleNet of the same bus. If the destination address is for a user outside the local bus, the packet is transmitted directly to the active node on the top of this bus using a dedicated wavelength. By transmitting through the outgoing ShuffleNet next, the packet can reach the active node of the destination bus. Finally, the packet is transmitted to the destination user by a dedicated wavelength. Since the packet transmission within one bus does not affect the packet transmission in another, the same wavelength can be used in every bus.

Shu-Shu2

The third model, called Shu-Shu2, is similar to the second model except that there is no dedicated channel between a user and its active node. Rather, an active node is shared by both the local ShuffleNet and the outgoing ShuffleNet. If a packet belongs to outgoing traffic, it has to go through the local ShuffleNet, usually in multihops in order to get the active node. From there it is transmitted through the outgoing ShuffleNet to the active node of the destination bus. Finally, it is sent to the destination user through the local ShuffleNet, again in a multihop path.

3.2.5 Summary and Remarks

An multihop photonic network, the ShuffleNet logical interconnections, has been described. Three design models for Treenet using ShuffleNet have been presented. There are two problems with the ShuffleNet implemented in Treenet physical topology.

- The ShuffleNet is only defined if the number nodes equals to kp^k. At present, no way has been found to add one node at a time changing a small number of connections and moving from a network with $p - i + 1$ nodes.
- The alternate paths between a source and a destination are not good. If the preferred path is blocked or inoperable, the alternate paths are much longer.

Although there are some limitations, ShuffleNet is still very attractive for Treenet implementations because of the advantages we discussed above. In the following chapter, performance analysis will be conducted for ShuffleNet, STDM and the three proposed design models.

Chapter 4

Performance Analysis of Photonic MAN

4.1 Introduction

In this chapter, the performance of ShuffleNet and three design models as measured by delay and throughput will be considered. First, the performance of Shufflenet will be analyzed. The delay and throughput performance for Treenet will be presented. A comparison of the performance for the three design models will be discussed.

4.2 Performance Analysis for ShuffleNet

4.2.1 Modeling and Assumption for ShuffleNet

Due to the node-to-node dependency in a tandem-node network, the Independent Assumption ([22]) will be used, so that each node can be analyzed independently. Furthermore, since the ShuffleNets are symmetrical, the discussion will be focused on one user output node. The queueing model of the user output for the perfect Shufflenet is shown in Figure 4.1. There are three arrival streams to each node: an external arrival and two internal arrivals from two optical channels. We assume the external arrival process is Poisson. The internal arrival process is assumed Poisson because the node is independent of other nodes and the number of arrivals to this node is large. By assuming all arrivals to be Poisson, and packet lengths to be fixed, the output user buffers to be infinite in size, the queueing system associated with

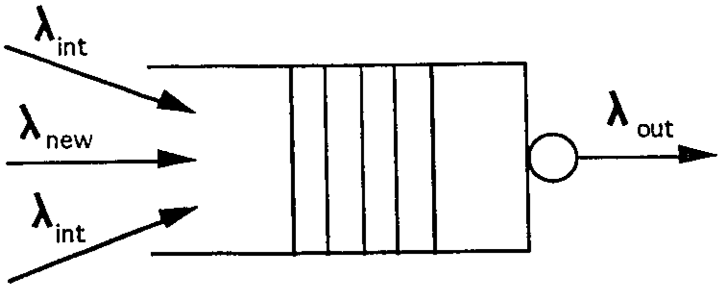

Figure 4.1: Queueing Model for ShuffleNet

a given node can be modeled as an M/D/1 queue. Note that the internal arrivals are outputs of M/D/1 queues and are in general not Poisson. However, they will be treated as Poisson and be verified by simulations to see how good this assumption is.

With the probability P_{out}, a packet arriving from an optical channel is to be disregarded by the NIU, and with probability $(1 - P_{\text{out}})/2$, a given packet must be routed to a given one of the NIU outputs (for each packet, two optical outputs are equally likely to be selected). Also, a packet arriving from the user is equally likely to go to one of the two optical channels.

Since a packet takes on the average $E[\text{Hops}]$ to reach its destination, the probability that a packet at a given node would depart the network (i.e., arriving at its destination) is $P_{\text{out}} = 1/E[\text{Hops}]$.

4.2.2 Queueing Delay Analysis

For the purpose of analysis and verification by simulations, ShuffleNet with $p = 2$ will be considered in the following section. When $p = 2$ (Perfect ShuffleNet), the expression for $E[\text{Hops}]$ is given by:

$$E[\text{Hops}] = \frac{(3k^2 - 3k)2^{k-1} + 2k}{k2^k} \tag{4.1}$$

For a stable system shown in Figure 4.1, the average arrival rate to any user is equal to the average departing rate, which is:

$$\lambda = \lambda_{\text{new}} + 2\lambda_{\text{int}} = \lambda_{\text{out}} \tag{4.2}$$

where λ, λ_{new}, λ_{int} and λ_{out} are the total arrival rate, the external arrival rate, the arrival rate from the optical channel and the outgoing rate respectively. $\lambda_{\text{int}} = \frac{(1-P_{\text{out}})}{2}\lambda_{\text{out}}$ accounts for one internal arrival that must be routed to the user outputs (for each packet, both optical outputs are equally likely to be selected). From Equation 4.2,

$$\lambda = \frac{1}{P_{\text{out}}}\lambda_{\text{new}} = E[\text{Hops}]\lambda_{\text{new}} \tag{4.3}$$

where $P_{\text{out}} = 1/E[\text{Hops}] = 1/2$ for the perfect Shufflenet, and the total arrival rate is $\lambda = 2\lambda_{\text{new}}$.

Let τ be the service time of a packet, then the system will be stable provided $\lambda\tau < 1$. Let $\rho_1 = \lambda\tau$ be the normalized traffic load presented to a given user, and d_1 be the queueing delay of a packet. Since the packet length is fixed, the queue model is a M/D/1 queue, according to [22, page 191]:

$$d_1 = \tau + \frac{\rho_1}{2(1-\rho_1)} \tag{4.4}$$

From this, the mean queueing delay in multihopping through the network is the sum of the delays experienced for each hop. The mean path delay d through all these hops is given by:

$$d = E[\text{Hops}] \tag{4.5}$$

where $E[\text{Hops}]$ and d_1 are given in Equations 4.1 and 4.4 respectively.

Results are plotted in Figure 4.2. The load in these illustrations is that offered directly by the end users. In Figure 4.2, the mean delay is normalized with respect to the average number of packet transmission time τ. It can be seen from these curves that, as the number of users and permissible offered load increase, more delay is encountered since the average number of hops taken by a representative packet increases. However, for all cases considered, the queueing delay is less than about 20 packet transmission times if the offered load is less than 80% of the achievable end-user capacity.

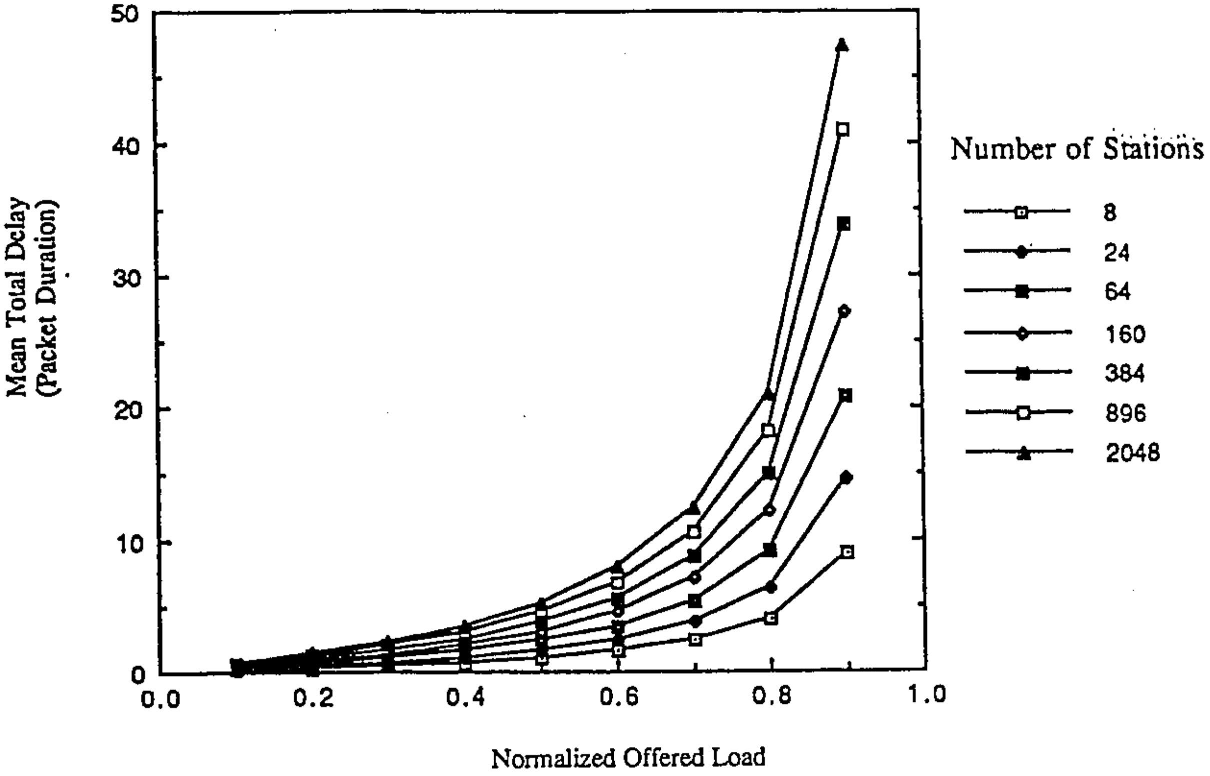

Figure 4.2: Delay versus Normalized Offered Load per NIU for Shufflenet

4.2.3 Throughput Performance

Another performance measure is the information throughput. Information throughput can be defined as the total number of information bits transmitted per unit time. Even though a certain number of overhead bits for addressing purposes are transmitted in addition to the information bits, these overhead bits are included in calculating the information throughput.

Let's now introduce some merit parameters. The normalized throughput is S. Throughput and the average distance are mutually correlated. Since the probability that a packet at a given node would depart the network is $P_{\text{out}} = 1/E[\text{Hops}]$. The normalized throughput per user is $S = \rho'/\rho$. Where ρ' is the average number of packets that can get out of the network and $\rho' = \rho/E[\text{Hops}]$, ρ is the average number of packets presented to a given user, and $\rho \leq 1$. The maximum normalized throughput per user is:

$$S_{\text{max}} = \frac{1}{E[\text{Hops}]} \tag{4.6}$$

Numerical results are shown in Table 4.1 for the index k, the corresponding number of regenerative users N, the number of channels needed w, the normalized throughput per user S, and the normalized throughput of the network. It is noted, for example, that with $k = 7$, a network serving 896 users can be constructed. A total of 1792 distinctive channels are needed. The normalized throughput per user is only 0.11, but this is entirely acceptable in view of the fact that the throughput of network increases when the number of users increases.

4.3 Performance Analysis for Three Design Models

4.3.1 Notations

The performance of the three design models as measured by delay and throughput is considered now. This delay encompasses queueing, transmission, propagation, and processing delay. Since the Treenet is hierarchical in architecture, the queueing delay is the sum of the delays on both the upper and lower level. In all the performance

k	Number of Users N	Number of Channels W	Throughput of the network	Throughput per user
2	8	16	4	0.5
3	24	48	7.44	0.31
4	64	128	14.08	0.22
5	160	320	27.2	0.17
6	384	768	49.92	0.13
7	896	1792	98.56	0.11
8	2048	4096	184.32	0.09
9	4608	9216	368.64	0.08

Table 4.1: Numerical Parameters for Dedicated Channels

analysis, the overhead bits are included in the packet format since the size of the overhead is small. In order to calculate the delay and throughput performances, the following symbols will be used for three design models:

SHU-STDM

D total delay including queueing, transmission and propagation delays

D_1 queueing delay for a packet in a station on the lower bus level

D_2 queueing delay for a packet in a active node on the upper tree level

D_{p1} propagation delay in the lower bus level

D_{p2} propagation delay in the upper tree level

N_1 number of stations in a bus

N_2 number of active nodes in the upper tree level

λ_{new} new packet arrival rate to a station

T slot time length dedicated to a station

T_F frame time length dedicated to a bus

λ_2 total arrival rate to a active node including the arrival rate from the bus and the arrival rate from two optical channels

τ_2 service time for a packet in the active node

ρ_1 utilization factor in the lower level

ρ_2 utilization factor in the upper level

$E_2(\text{Hops})$ average number of hops in the upper level Shufflenet

S_1 throughput per station in the lower level

S_2 = throughput per active node in the upper level

SHU-SHU1 and SHU-SHU2

D total delay including queueing, transmission and propagation delays

D_1 queueing delay for a packet in a station on the lower bus level

D_2 queueing delay for a packet in a active node on the upper tree level

D_{p_1} propagation delay in the lower bus level

D_{p_2} propagation delay in the upper tree level

N_1 number of stations in a bus

N_2 number of active nodes in the upper tree level

λ_{new} new packet arrival rate to a station

λ_2 total arrival rate to a active node including the arrival rate from the bus and the arrival rate from two optical channels

τ_1 service time for a packet in a station

τ_2 service time for a packet in a active node

ρ_1	utilization factor in the lower level
ρ_2	utilization factor in the upper level
$E_1(\text{Hops})$	average number of hops in the lower level Shufflenet
$E_2(\text{Hops})$	average number of hops in the upper level Shufflenet
S_1	throughput per station in the lower level
S_2	throughput per active node in the upper level
μ	proportion of the outgoing packets out of the total arrival packets in a station

4.3.2 Performance Analysis for SHU-STDM

In order to calculate the delay performance, it is assumed that the output of the M/D/1 system is Poisson. In this model, the lower level is STDM, where the capacity in the form of recurring time slots is dedicated to each of the data sources sharing the transmission facility. Flow on the line is blocked into fixed-length frames. The frame is allocated so that periodically recurring slots carry only the output of a particular source. Messages arrive at a rate of λ_{new} messages per second. These messages are stored in buffers until the portion of a frame dedicated to a source is available. For purposes of explanation it is helpful to visualize TDM as being implemented by a sort of commutator (see Figure 4.3). When it is the turn of a particular source to empty its buffer the can be imagined being visited by a server. The server stays at each buffer for a fixed amount of time, removing at most a specified maximum amount of data. Since the action of the server does not depend on the traffic from the sources, the time required for the server to cycle through all sources is constant and the sources may be analyzed independently.

The analysis is focused on a particular source. The beginning of a TDM multiplexing frame is taken to be the point where the server arrives at the source's buffer (see Figure 4.4). It is assumed that the duration of the frame is T_F seconds, where $T_F = N_1 T$, N_1 is the number of slots in the frame, and T is a slot time dedicated to a

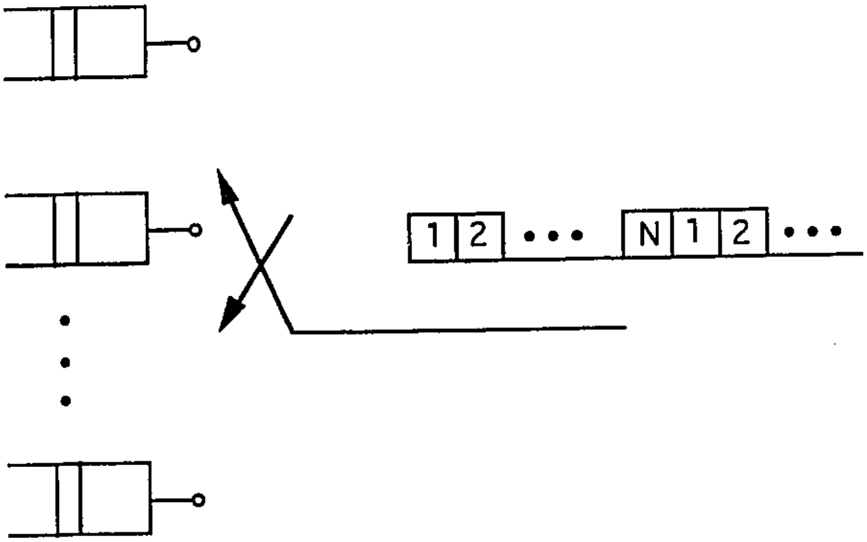

Figure 4.3: Commutator Model of Synchronous Time Division Multiplexing

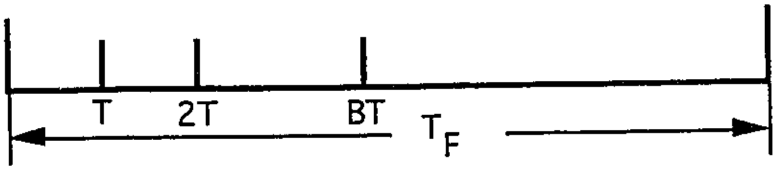

Figure 4.4: Time Division Multiplexing Frame

station. Then using an imbedded Markov chain analysis the average queueing delay is given by ([13]):

$$D_1 = \frac{\lambda_{\text{new}} {T_F}^2}{2(1-\rho_1)} + \frac{T_F}{2} + T \tag{4.7}$$

where $\rho_1 = \lambda_{\text{new}} T_F$ and λ_{new} is the average external packet arrival rate for a station.

Since the upper level is a ShuffleNet, the equation of mean queueing delay D_2 for the upper level has been introduced in the previous section. The total delay in the upper level is the sum of the delay of the all active nodes that a packet passes when it is transmitted from the source to destination station. The discussion is only focused on one active node. The total arrival rate from the bus to the active node is $N_1\lambda_{\text{new}}$. In order to simplify the calculation, it is assumed the output process of STDM is

Poisson process, and the arrival process to the active node is Poisson distributed. It is also assumed that the service time for a packet in the active node is τ_2. According to the analysis of the queueing delay of M/D/1 queue, The mean queueing delay for an active node in the upper level is:

$$D_2 = \tau_2 + \frac{\rho_2}{2(1-\rho_2)} \tag{4.8}$$

where $\rho_2 = \lambda_2\tau_2$, λ_2 is the total arrival rate to the active node. It is assumed that a packet need to pass average $E_2[\text{Hops}]$ hops from the active node on top of the source station to the active node on top of the destination station. Since the active node is not end station, the packet needs to experience $E_2[\text{Hops}] + 1$ times delay of active nodes. We have:

$$\lambda_2 = (E_2[\text{Hops}] + 1)N_1\lambda_{\text{new}}$$

The mean total delay in multihopping through the network is the sum of node delays in the upper and lower levels and the path propagation delay, which can be expressed as follows for two different cases:

- If the packet is local on the bus, this packet is transmitted to the active node on top of this bus and directly to the destination station from the active node. The mean total delay is given by:

  $$D = D_1 + D_2 + D_{p_1}$$

  where D_{p_1} is the average propagation delay for a packet transmitted from a source station to a destination station within the bus. D_1 and D_2 are given by Equations 4.7 and 4.8, respectively.

- If the traffic is outgoing, the packet has to suffer the delay of the local traffic plus the delay in the upper level. The mean total delay is given:

  $$D = D_1 + D_2(E_2[\text{Hops}] + 1) + D_{p1} + D_{p2} \tag{4.9}$$

  where D_{p2} is the average propagation delay for a packet transmitted from the active node on top of a source station to the active node on top of the destination station.

Since the network is hierarchical in architecture, the total throughput is counted in both layers. By assuming that the arrival rate to each station in the bus is λ_{new}, the throughput per user is $S_1 = \lambda_{\text{new}}$. The throughput per active node is

$$S_2 = \frac{N_1 \lambda_{\text{new}}}{(E_2[\text{Hops}] + 1)}$$

.

4.3.3 Performance Analysis for Shu-Shu1

In this model, both the upper and lower level are ShuffleNets. The following symbols are used:

Assuming that the number of stations in a bus is $N1 = k_1 2^{k_1}$, the queueing delay for the station in a bus is given by:

$$D_1 = \tau_1 + \frac{\rho_1}{2(1-\rho_1)} \tag{4.10}$$

where $\rho_1 = E_1[\text{Hops}]\lambda_{\text{new}}\tau_1$.

In the upper level, the total arrival rate to an active node is $N_1 \mu \lambda_{\text{new}}$. Assuming that the number of active nodes is $N_2 = k_2 2^{k_2}$, the service time for a packet is τ_2, the queueing delay for an active node is given by:

$$D_2 = \tau_2 + \frac{\rho_2}{2(1-\rho_2)} \tag{4.11}$$

where $\rho_2 = (E_2[\text{Hops}] + 1) N_1 \mu \lambda_{\text{new}} \tau_2$.

The mean total delay can be expressed as follows for two different cases:

- If the traffic is local on the bus, this packet is transmitted to the active node on top of this bus and directly to the destination station from the active node. The mean total delay is given by:

  $$D = E_1[\text{Hops}] D_1 + D_{p1}$$

  where D_{p_1} is the average propagation delay for a packet transmitted from a source station to a destination station within the bus. D_1 and D_2 are given by Equations 4.10 and 4.11, respectively.

- If the traffic is outgoing, the packet has to suffer the delay in both the upper and lower levels. The mean total delay is given by:

$$D = D_1 + D_2(E_2[\text{Hops}] + 1) + D_{p1} + D_{p2}$$

Since both of the upper and lower levels are ShuffleNets, the throughput per station is $S_1 = \lambda_{\text{new}}/E_1[\text{Hops}]$. The throughput per active node is:

$$S_2 = \frac{N_1 \mu \lambda_{\text{new}}}{(E_2[\text{Hops}] + 1)}$$

4.3.4 Performance Analysis for Shu-Shu2

In this architecture, both the upper and lower levels are ShuffleNets. The active nodes are shared by both levels. When a packet arrives at station 1 in a bus, if this packet is inside the traffic of this bus, it will be transmitted to the destination station through the local ShuffleNet in this bus. If this packet is outgoing traffic outside the bus, it will be transmitted to the active node on top of this bus through the local ShuffleNet in the bus, then to another active node on top of the destination station through the upper level ShuffleNet, finally to the destination station through the local ShuffleNet in the bus which the destination station belongs to.

- If the traffic is local, the end-to-end delay is given by:

$$D = E_1[\text{Hops}]D_1 + D_{p_1}$$

- If the traffic is outgoing, the end-to-end delay is given by:

$$D = E_1[\text{Hops}]D_1 + (E_2[\text{Hops}] + 1)D_2 + E_1[\text{Hops}]D_1$$

The throughput per user is $\lambda_{\text{new}}/E_1[\text{Hops}]$. The throughput per active node is $N_1 \mu \lambda_{\text{new}}/(E_1[\text{Hops}])(E_2[\text{Hops}] + 1)$.

4.4 Summary

In this chapter, the delay and throughput performance are analyzed for the three design models proposed in Chapter 3. The analysis of performance evaluation is based on certain assumptions. It is assumed that the arrival process for all nodes is Poisson process although some arrival processes are the output of M/D/1. By assuming the service time is fixed, each node is simple M/D1 queueing system. The queueing delay formula is achieved by introducing M/D/1 queueing delay equation. In order to verify the results of the performance analysis, Comparison with the simulation results will be made in the next chapter.

Chapter 5

Analysis by Simulation

5.1 Introduction and Motivation

With the fast development of the telecommunication technology, it becomes more and more difficult to perform test on the performance of a system. The evaluation of the system performance relies on the analysis, modeling and simulation of the proposed system. With the correct method of modeling and simulation, the evaluation of the system performance can be conducted with much less cost, yet achieving very accurate performance evaluation. There are two main advantages of using computer simulation. First, simulation frees analyst from a great deal of repetitive work involved in substituting numbers into formulas and enables the analyst to concentrate on the results. Second, simulation is the insight into system performance provided by both the modeling process itself and the experience gained from simulation experiments.

In the previous chapter, we presented the queueing analysis for ShuffleNet and three design models. In order to validate the assumption introduced in the analysis, we run the simulation using QNAP2 simulation package to run the simulation.

QNAP2 simulation package can be defined as a system for describing, handling and solving queueing network models. It is comprised of a specification language which is used for the description of the models under study and the control of their resolution, and of several resolution models, or solves, implementing the different techniques currently available. In QNAP2, a queueing network consists of a set of stations (one or several servers and one queue) through which circulate customers according to given routing rules; the customers may be distributed into several classes characterizing different behaviors and different processing in the stations. The processing done by a station may be described by a simple time duration defined by its probability distribution or by a complex algorithm which may include synchronization operations.

In the following sections, we will present the simulation models and results for ShuffleNet and three design models. Certain assumptions are made in order to modeling the system.

5.2 Simulation for ShuffleNet

5.2.1 Simulation Models

Since the nodes in ShuffleNet are symmetrical, We only need to focus on one user node only. The queueing model for a 8-port ShuffleNet is shown as in Figure 5.1. There is a queue in every station. When packets (including new packets and packets from two other stations) arrive at a station, they will wait on the queue. The queues are FCFS and have the infinite length.

5.2.2 Assumptions

In order to run the simulation, the following assumptions are made:

- Every station in the ShuffleNet is symmetrical.
- The queue in every station has the infinite queue length. There is no packet loss in the network. The queue scheme is FCFS.
- The traffic in the system is uniform traffic.
- Every packet has the fixed length.
- The new arrival process at a station is Poisson distributed. The service time of each queue in the station is fixed.
- For each packet transferred from one station to another station a routine is fixed.

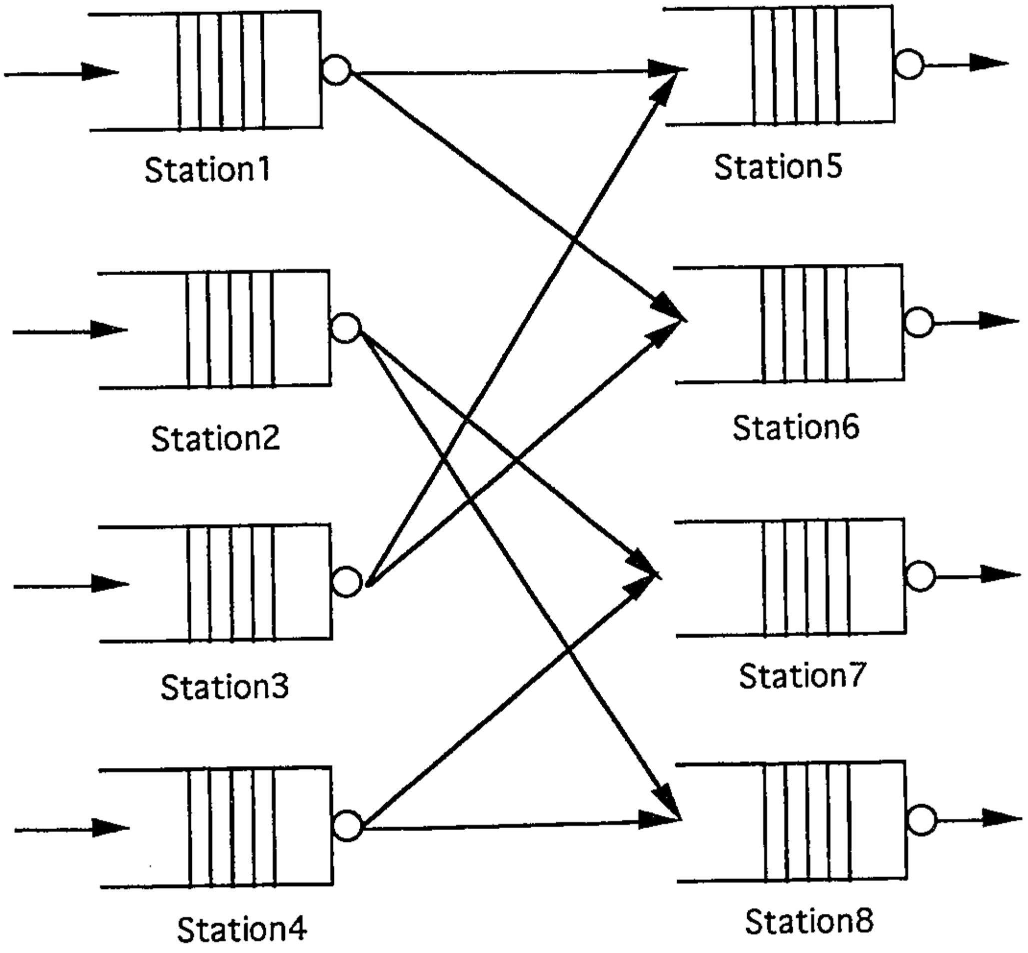

Figure 5.1: Simulation Model for ShuffleNet

- The distance between every two stations is very close. The propagation delay is ignored.

Simulation has been run using the QNAP2 simulation package. The QNAP2 simulation program for ShuffleNet is shown in Appendix B. For each simulation run, the simulation time is chosen to have a convergent result, and a 95% confidence interval is also provided automatically at the end of the simulation.

5.2.3 Simulation Results

Simulation results are shown as follows. Since the 95% confidence intervals are found to be very small, they are not shown for clarity purpose. Figures 5.4 and 5.5 show the mean and variance of queueing delay for packet transmissions with respect to normalized throughput. The results from the analysis and simulation agree with each other very well. From these two examples, we observed that the proposed M/D/1 analytical approach for ShuffleNet gives a very close performance of ShuffleNet.

A more detailed discussion of results are as following. Figure 5.6 shows the mean packet queueing time in one station versus normalized throughput for different size of stations. We can see that the delay increases dramatically when the size of the station increases. The maximum normalized throughput per station decreases from 0.5 to 0.2 when the size of the station change from 8 stations to 64 stations. This is because more hops are required for a packet transmission from a source to a destination when the size of the station increases. The packet has to wait longer time to transmit in the queue of the station since more packets are waiting in the station. The normalized throughput for one station which equals to $1/E[\text{Hops}]$ decreases when more hops are required for packet transmission.

In the next section, some results on three design models will be presented. The delay and throughput performance will be compared.

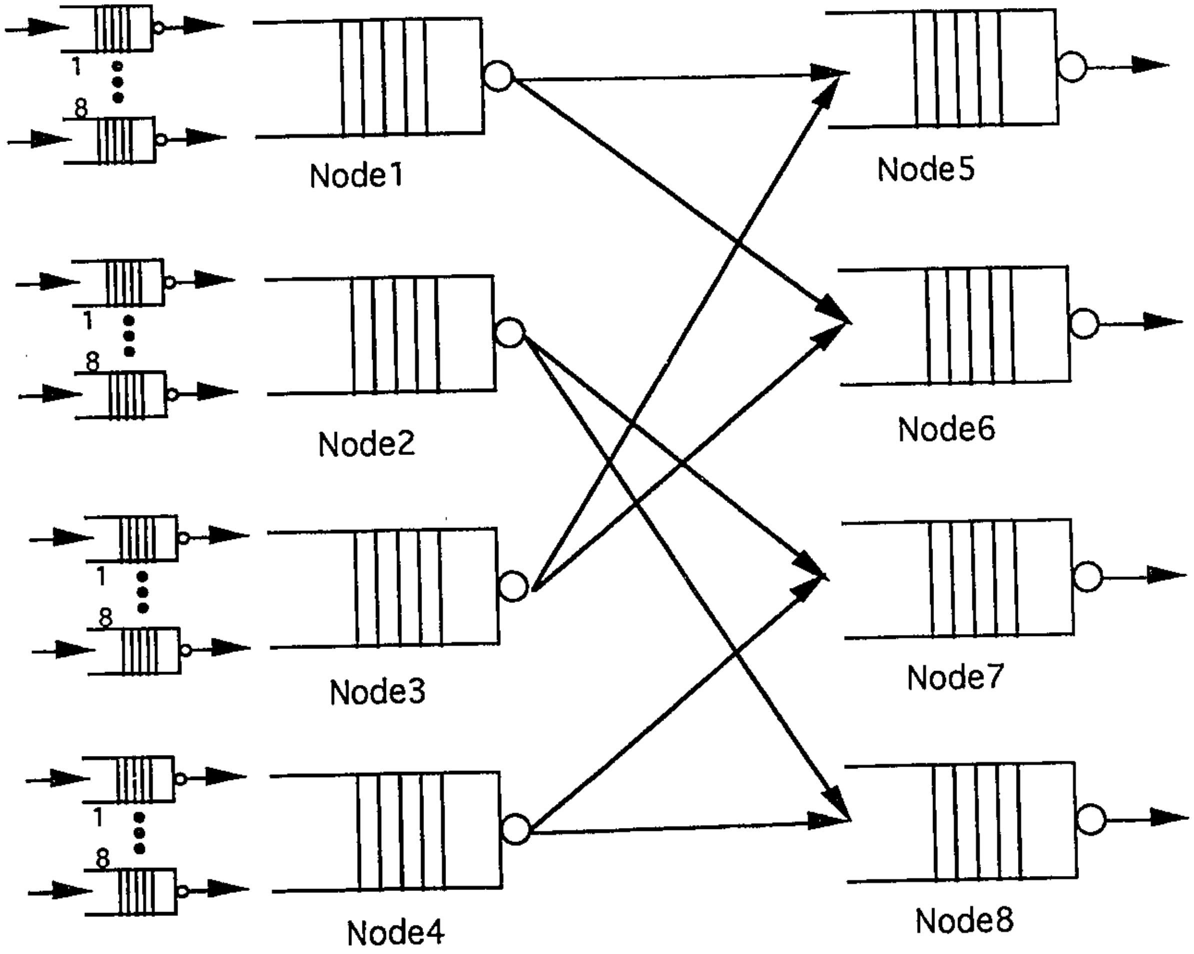

Figure 5.2: Simulation Model for Treenet

5.3 Simulations for Three Design Models

5.3.1 Simulation Models

The Treenet we proposed in the paper is a hierarchical architecture with different logic topologies on the upper and lower levels. The simulation model for Treenet with 64 stations is shown in Figure 5.2. There is a queue in every station and every active node. The input of the queue in the active node is the outputs of the queue from eight stations in the bus.

In the first model, ShuffleNet is used in the upper level. The simulation model of

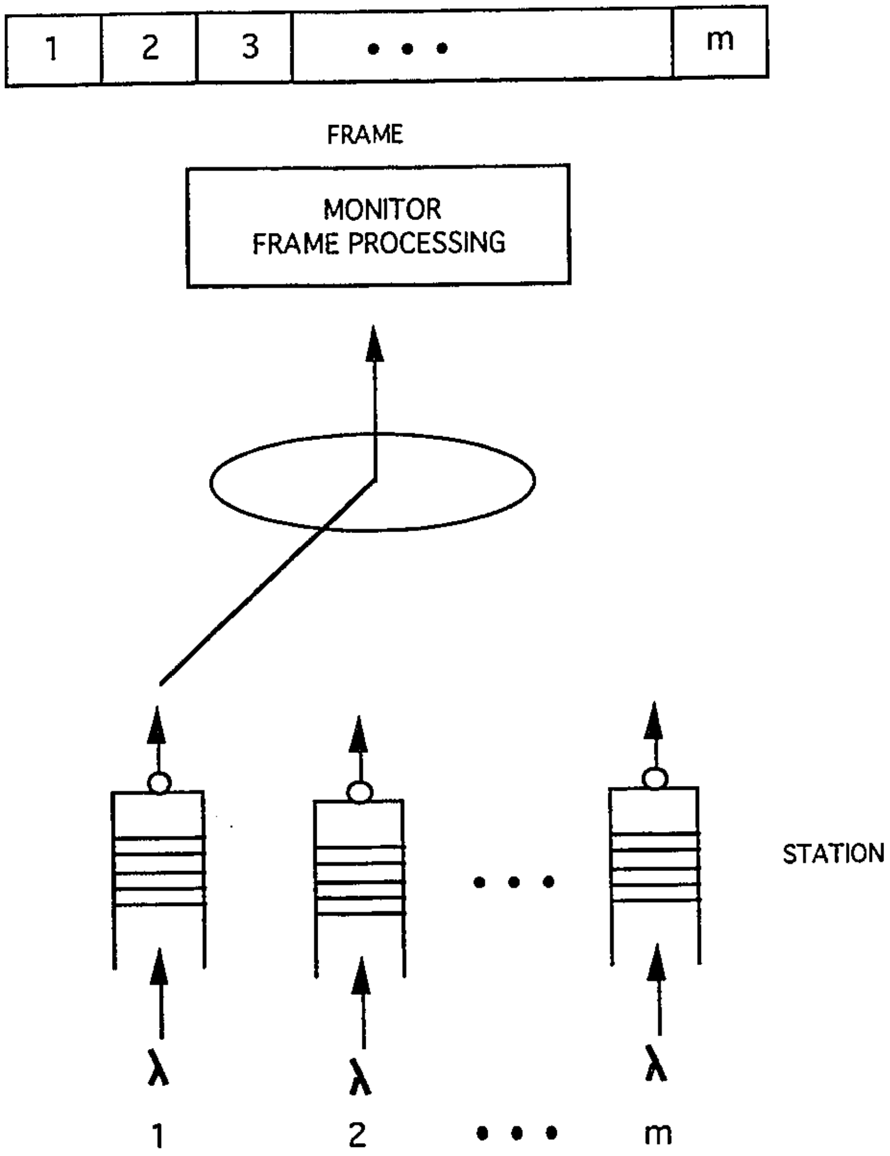

Figure 5.3: Simulation Model for STDM

the ShuffleNet is shown in Figure 5.1. STDM is in the lower level of the first design model. Figure 5.3 shows the simulation model for STDM. The generation of the frame is modeled by the use of a single server successively passing the transmission permission to every station. When a station receives the permission and has packets queued, it transmits right away. When a station receives its permission and has no packet in its queue, it lets the time slot pass by without being used. Packets arriving during this time slot are not allowed to transmit. After its slot has gone by the station, no matter whether it has transmitted or not, must wait for the same slot in the next frame. After having passed the slot to every station (i.e., from 1 to M), the single server station will have to restart a frame by passing the transmission permission to station again.

In the second model, ShuffleNet is used in the lower level. There is a dedicate channel between each station and the active node on the top of the station. Only the local traffic is waiting in the queue of the station. When a outgoing traffic arrives to a station, it is transmitted to the active node immediately. The outgoing traffic does not suffer the queue delay in the local ShuffleNet. There is no queueing delay for a packet from this active node transmitted to the destination station since the packet can be transmitted to the destination station using the dedicated channel.

In the third model, there is no dedicated channel between the active node and the station. The transmission of the local traffic is the same as in the second model. The difference from the second model is that the outgoing traffic has to suffer of the queueing delay of the local ShuffleNets in the buses which the source station and the destination belong to.

5.3.2 Assumptions

The simulation has been run for the three design models according to the following assumptions:

- The size of the system is 64 stations. There are eight active nodes in the tree level and eight stations in each bus.

- The queue in every station has the infinite queue length. The traffic in the system is uniform traffic. Every packet has the fixed length.
- The arrival process to a station is Poisson distributed. The transmission rate of the active node is 1.6 Gbits/sec, transmission rate of the station is 200 Mbits/sec.
- Maximum station-node-station distance is 50km and maximum propagation delay is 250 μsec.

5.3.3 Simulation Results

Figure 5.7 shows the queueing delay for a packet waiting in one station verses the normalized throughput per station. We can see that the simulation results and the analytic results are matched very well. So the analytic model for STDM proposed in chapter 4 is proved the closed design model for STDM. Figure 5.8 shows the mean queueing delay for a packet waiting time in a active node verses the normalized throughput per active node. It can be seen that good agreement between the analytic and the simulation results can be achieved. This enhances the confidence of the assumption of M/D/1 queueing model for a active node. Figure 5.9 shows the end-to-end delay for a packet transmission from a source station to a destination station verses the normalized throughput per station. Again the analytic and simulation results agree with each other very well. The delay and throughput results of analysis and simulation for Shu-Shu1 are shown from Figures 5.10 to 5.12. The delay and throughput results of analysis and simulation for Shu-Shu2 are shown from Figures 5.13 to 5.15. From all these examples we observed that the proposed analytic approach in chapter 4 gives an exact performance of the Treenet systems.

More results are shown as following to compare to the performance of three design models. Figure 5.16 shows the End-to-End delay of local traffic for three design models. We can see that the End-to-End delay of local traffic transmitted from a source to a destination in SHU-STDM is much lower than that of the two other models. This is because the local packets have to go through the local ShuffleNet in the second and the third model when they are transmitted from a source station to a destination station. The queueing delay becomes longer in the local ShuffleNet. Figure 5.17 shows the End-to-End delay of outgoing traffic for three design models. The delay for the SHU-SHU2 increases much faster as the traffic load increases. The reason is that the queueing delay in the local ShuffleNet becomes longer since both the local and outgoing traffic must go through the local ShuffleNet in SHU-SHU2. From these figures, we can see that the delay performance for the SHU-STDM is better than others.

Figures 5.18 to 5.21 depict the mean packet vs. input traffic per station (packet arrival rate) for the packet length from 1000 *bits* to 50000 *bits*. Again we can see that the maximum throughput per node for SHU-SHU2 is much lower than that of SHU-SHU1 and SHU-STDM. The delay for the SHU-SHU2 increases dramatically in the heavy traffic load. Figure 5.22 shows the mean delay vs. packet length for SHU-SHU1. The delay increases when the packet length increases. From these figures, we can see that the delay performance for the SHU-STDM is better than others. The reason might be that the queueing delay becomes longer in the local ShuffleNet because of the extra transit traffic for SHU-SHU1 and SHU-SHU2.

The previous results are based on the uniform traffic. Figure 5.23 shows the end-to-end delay for packets transmission under nonuniform traffic. Here, only the arrival rate in one station is changed and the arrival rates in the rest of the stations are not changed. The delay increases dramatically when the packet arrival rate changes for one station in each bus. We can see that the small change of the traffic load affects the mean end-to-end delay of the packet.

5.4 Summary

In this chapter the queueing models foe ShuffleNet, STDM and three design models are presented. The performances of the ShuffleNet and three design models have been analyzed. The analytical results were validated by simulations. Comparison between three design models are made. From the simulation results, the SHU-STDM is found to be the best model for Treenet.

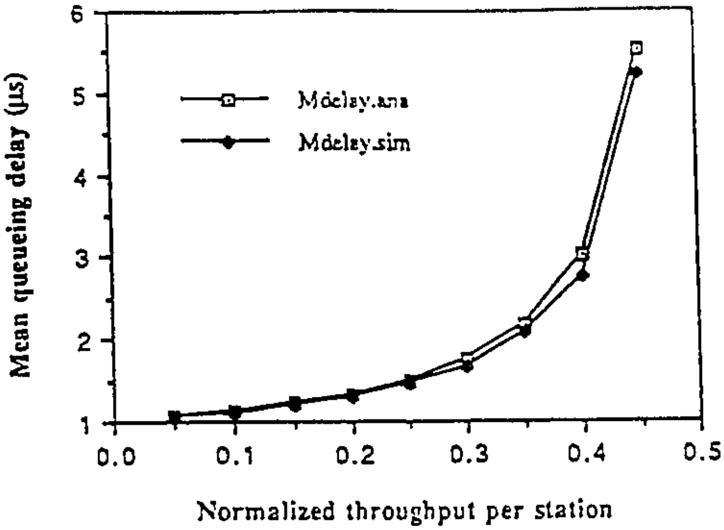

Figure 5.4: Mean queueing delay versus normalized throughput for 8-station SHU

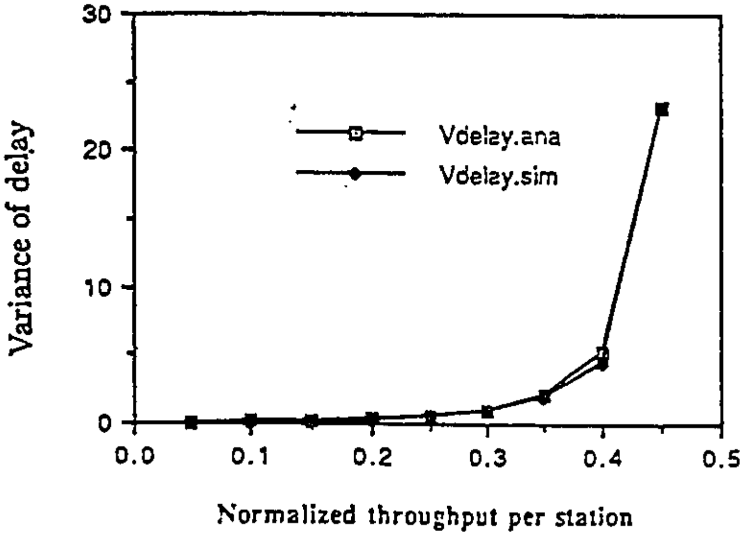

Figure 5.5: Variance of delay versus normalized throughput for the 8-station SHU

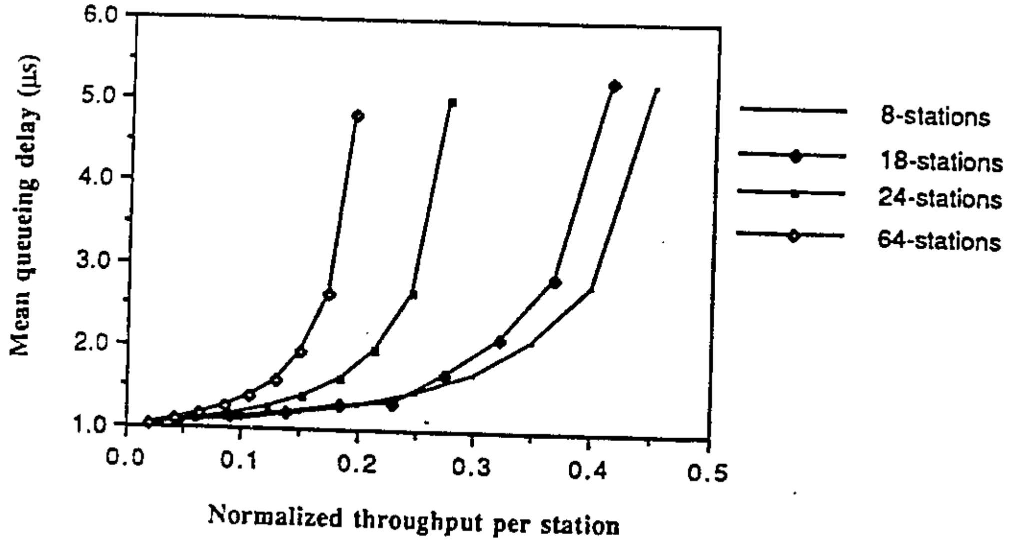

Figure 5.6: Mean queueing delay for different number of stations SHU

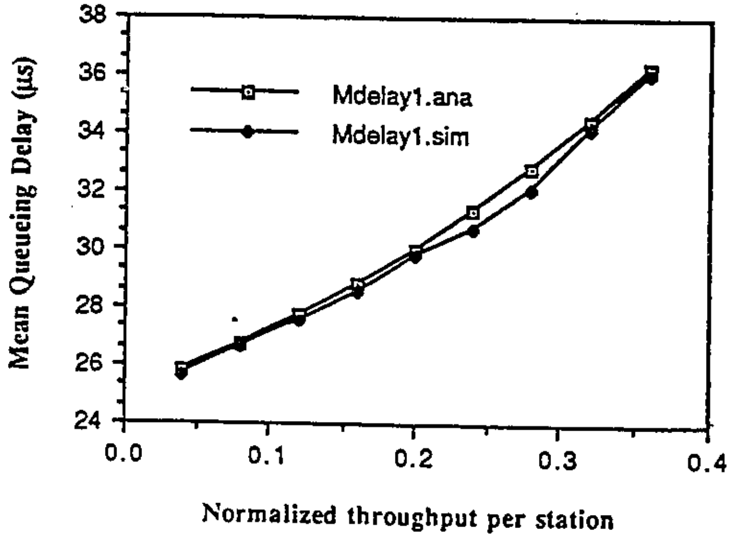

Figure 5.7: Mean queueing delay of the station in model SHU-STDM

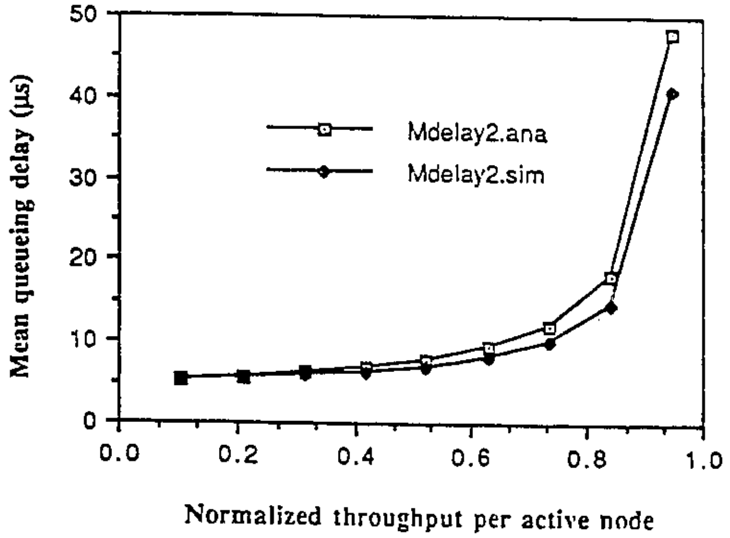

Figure 5.8: Mean queueing delay of the active node in model SHU-STDM

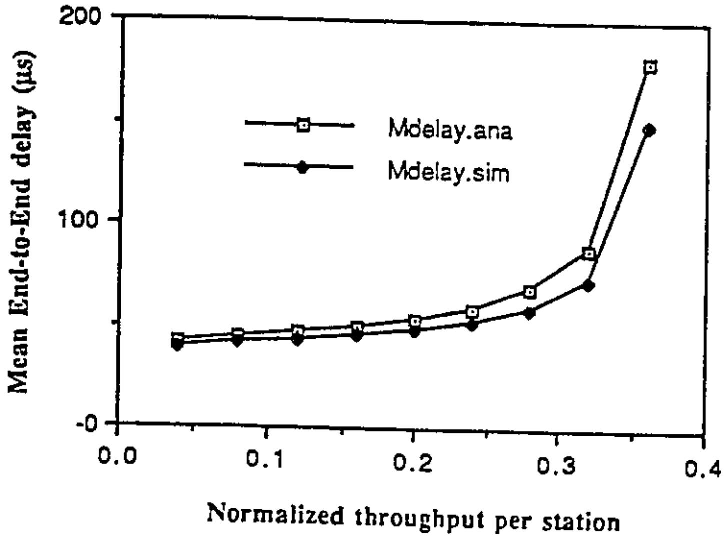

Figure 5.9: Mean End-to-End delay of packet transmission in model SHU-STDM

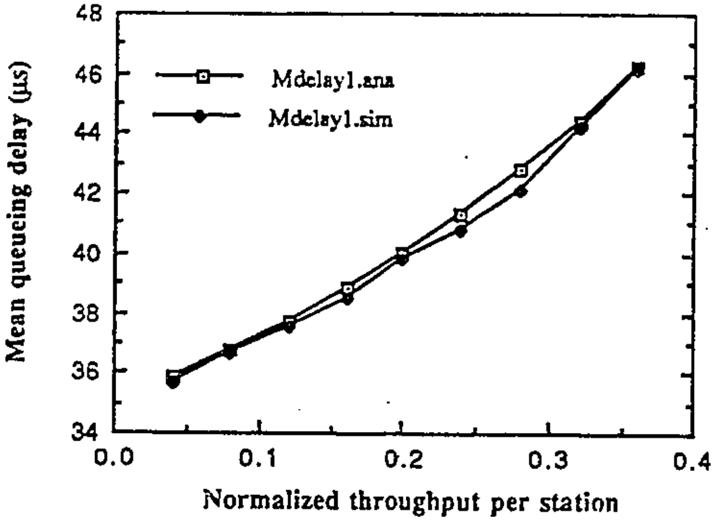

Figure 5.10: Mean queueing delay of the station in model SHU-SHU1

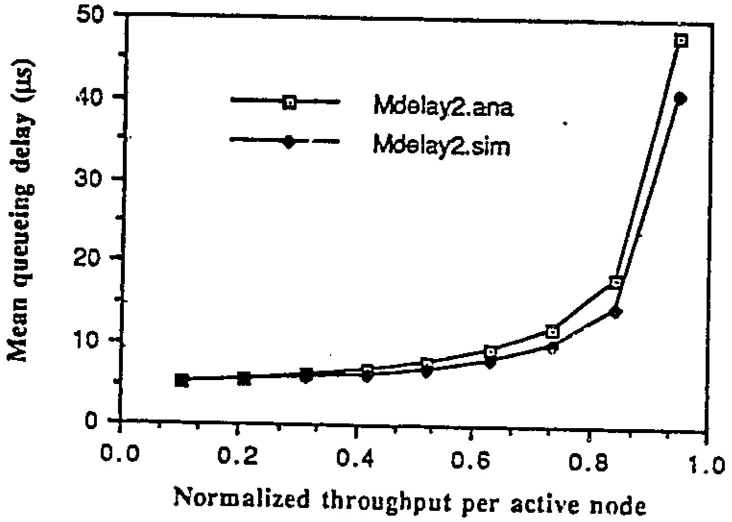

Figure 5.11: Mean queueing delay of the active node in model SHU-SHU1

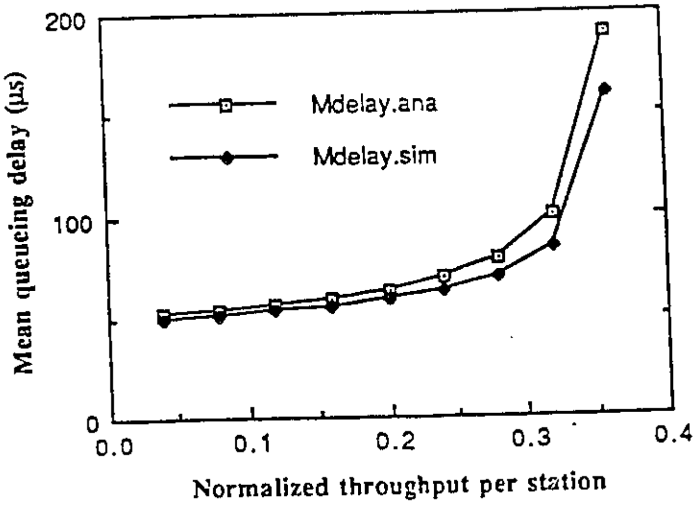

Figure 5.12: Mean End-to-End delay of packet transmission in model SHU-SHU1

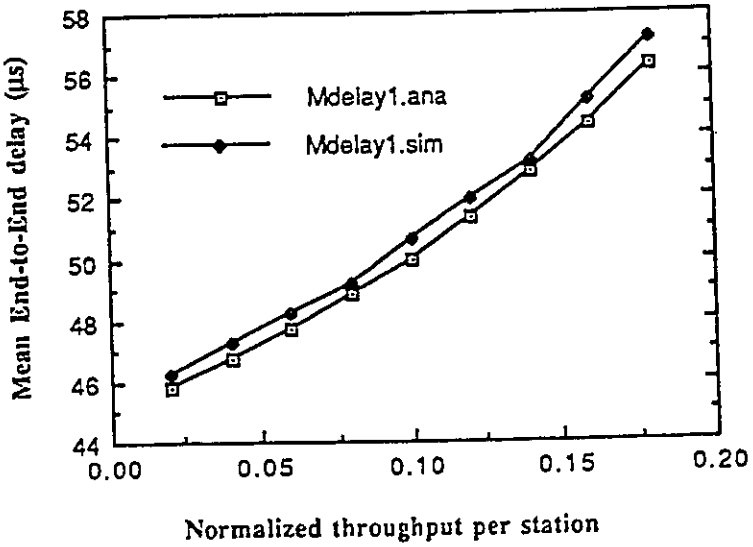

Figure 5.13: Mean queueing delay of the station in model SHU-SHU2

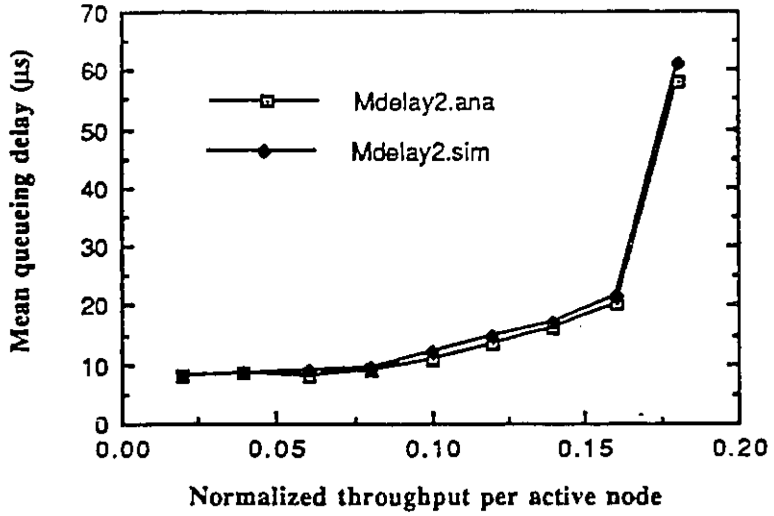

Figure 5.14: Mean queueing delay of the active node in model SHU-SHU2

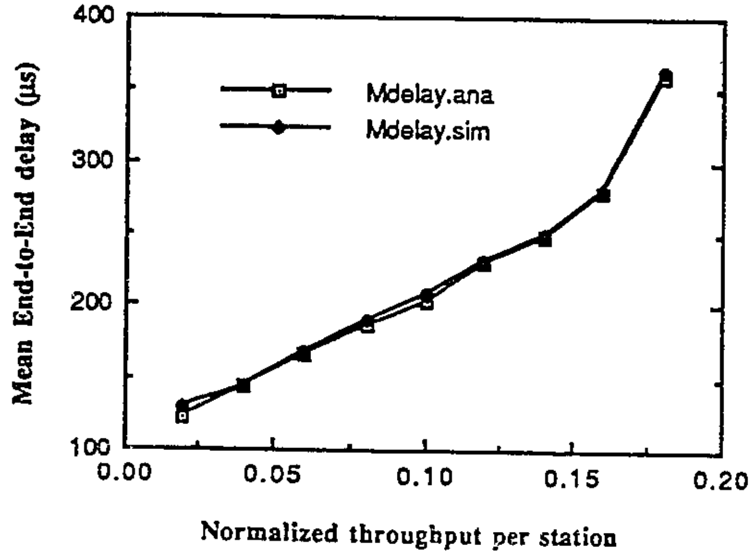

Figure 5.15: Mean End-to-End delay of packet transmission in model SHU-SHU2

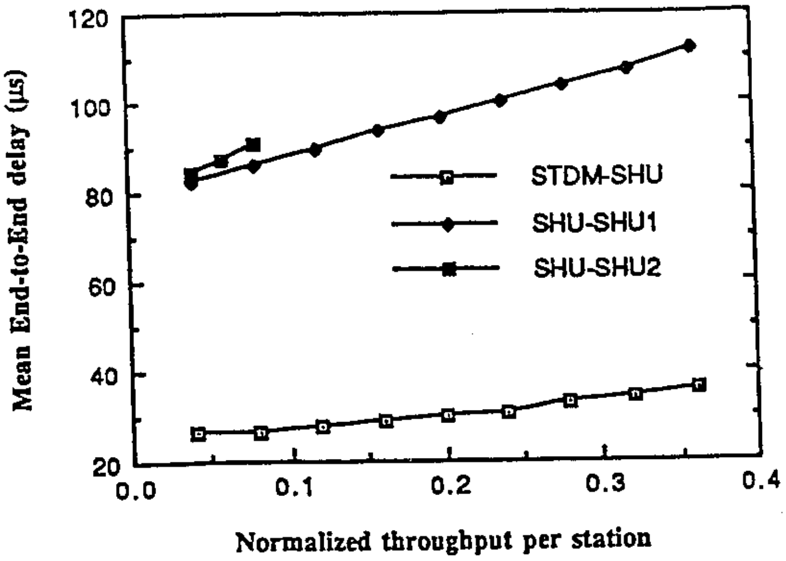

Figure 5.16: Mean End-to-End delay of local traffic for three models

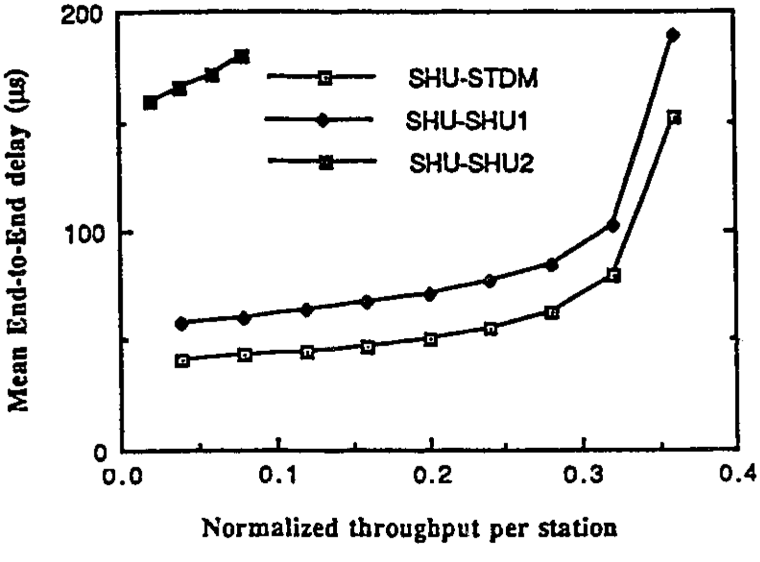

Figure 5.17: Mean End-to-End delay of outgoing traffic for three models

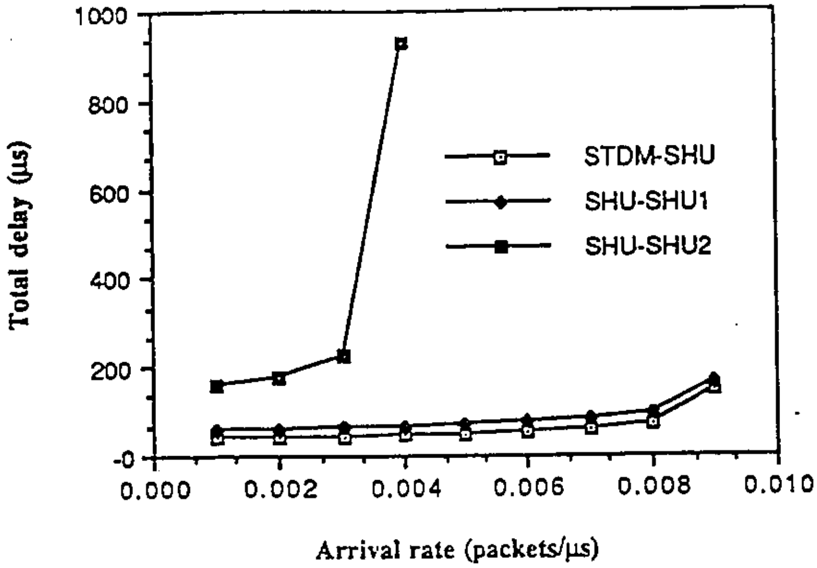

Figure 5.18: Total delay versus arrival rate for packet length L=1000 bits

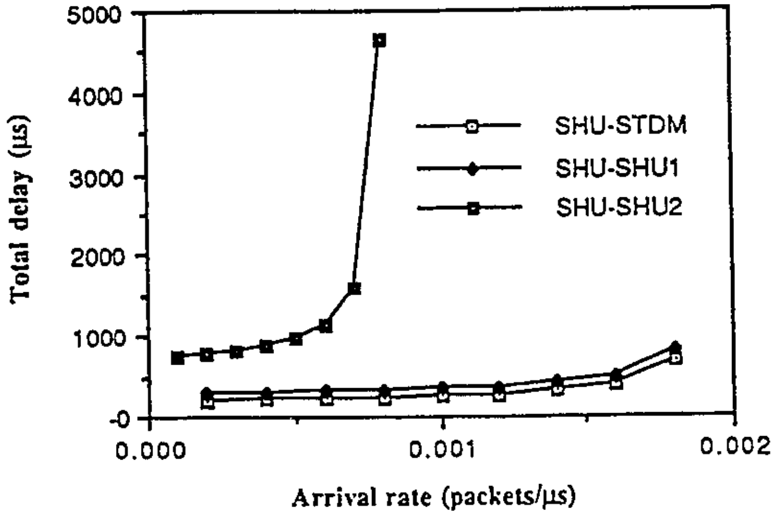

Figure 5.19: Total delay versus arrival rate for packet length L=5000 bits

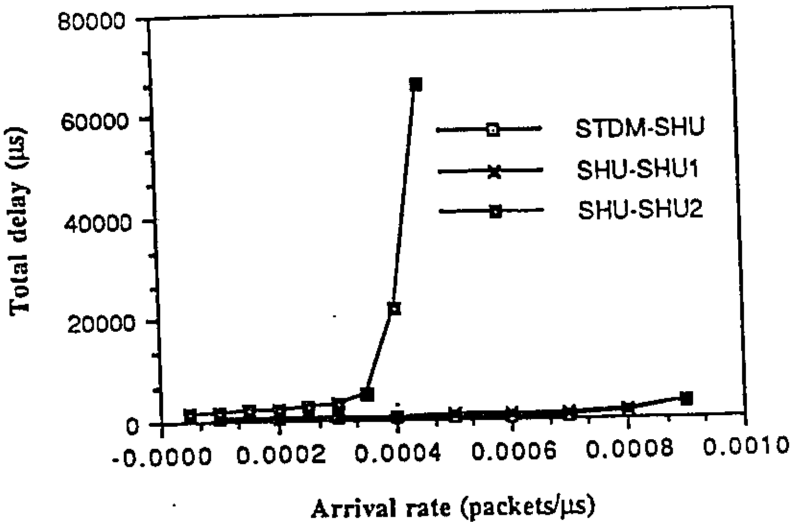

Figure 5.20: Total delay versus arrival rate for packet length L=10000 bits

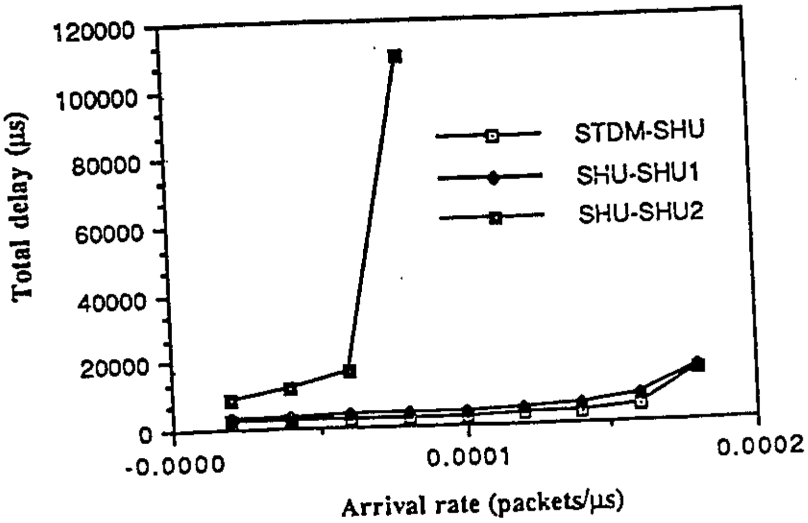

Figure 5.21: Total delay versus arrival rate for packet length L=50000 bits

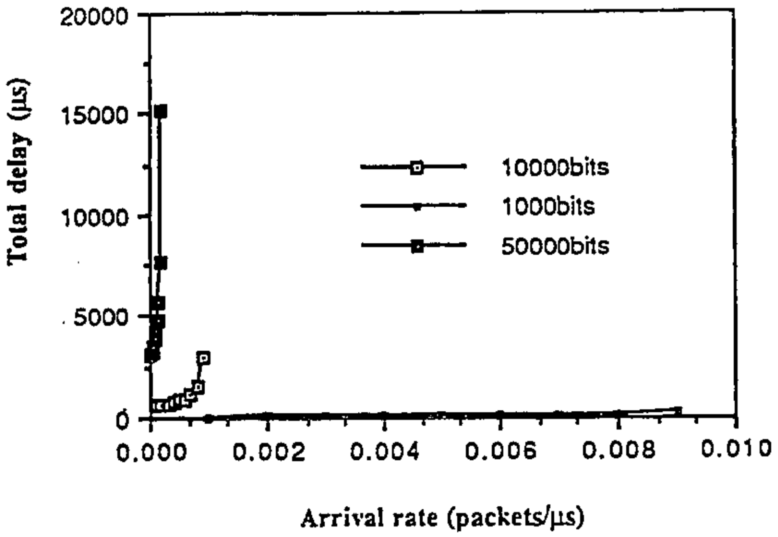

Figure 5.22: Total delay versus arrival rate for different packet length

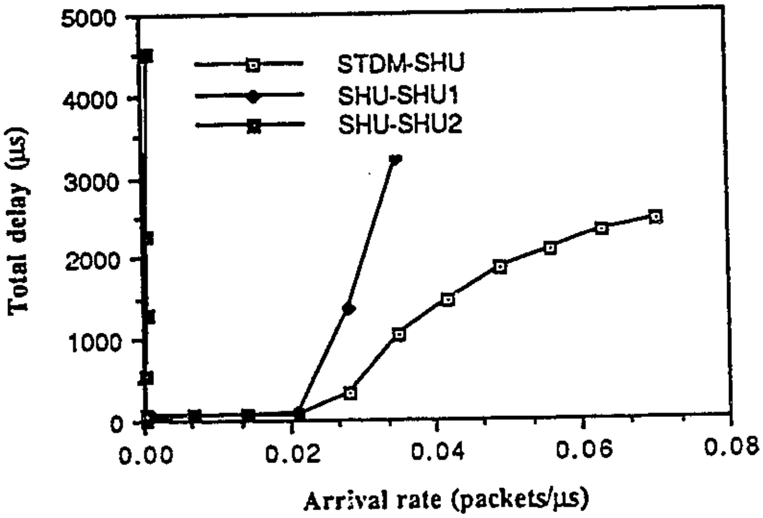

Figure 5.23: Total delay vs. arrival rate per node under no uniform traffic load

Chapter 6

Conclusions and Suggestions for Further Research

6.1 Concluding Remarks

The Metropolitan Area Network (MAN) is necessary to employ interconnection speeds of 100 Mb/s, geographical area coverage of up to 50 km, and network capabilities that would support hundreds of stations. The applications and services that emerged for the MAN architecture covered voice, video, and imaging areas. Advances in lightwave technology specifically fiber optics enabled the fabrication of relatively low-cost, low-attenuation fibers and high-quality optical transmitters, detectors, and passive components. All these demands for highly-efficient high-speed communication networks in metropolitan environment is opening new research horizons. High bandwidth, as well as a suitable topology and protocols are needed to support different types of traffic (data, voice, video, etc.). This thesis is based on the topology and protocol design for fiber optic MAN.

In this thesis, we first reviewed the existing optical network topologies such as bus, ring, star and tree. The limitation of the basic topologies is that they can not support large number of stations in a metropolitan area. To overcome the limitation of basic topologies, we introduce the fiber optic Treenet. The Treenet is a hierarchical architecture which has a tree in the upper level and buses in the lower level. The Treenet combines the advantages of both bus and tree topologies. We presented two kinds of Treenet: passive Treenet and active Treenet. The difference between the passive Treenet and the active Treenet is that there are active nodes between the tree and the buses. The active nodes are used for traffic filtering.

To overcome the large latency delay for passive Treenet proposed by Gerla and Fratta. We used a multichannel and multihop technology called Shufflenet as the basic protocol for the Treenet. We reviewed the multihop and Wavelength Division Multiplexing (WDM) techniques. WDM provides a powerful means to solve the electro-optic bottleneck and offer significant opportunities for high-speed networks. We then proposed the three design models as the protocols for Treenet. The network performance such as throughput and delay characteristics are analyzed for Shufflenet and three design models. Certain assumptions are made to simplify the analysis.

To verify the analysis results, a computer simulation program has been run. The detailed network modeling and modifications are given. The analytical results were compared to simulation results and found to match very closely. This indicates that the analytic approach for Shufflenet and three design models give the close performance.

Finally, we compared the delay and throughput results for three design models. We found that SHU-STDM is the best among three for the Treenet.

6.2 Suggestions for Future Research

In our work, we use Treenet as the basic topology. From the functional standard point, the tree could be replaced by a star. The star actually provide power savings with respect to the tree and would thus permit to support more stations. Another variation is that the logical Shufflenet can be replaced by a hypercube. That is, every station can have more wavelengths to transmit and receive. This increases the extra path for a packet transmission.

In our queueing analysis, we assume infinite buffer size in each queue. In future research, finite buffer size will be considered and more queue analysis will be made. Finally the implementation of proposed network to B-ISDN may be another interesting topic.

Bibliography

[1] B. W. Abeysundara and A. E. Kamal, "High-Speed Local Area Networks and Their Performance: A Survey", *ACM Computing Surveys*, Vol. 23, No. 2, pp 221-264, June 1991.

[2] A. S. Acampora, M. J. Karol and M. G. Hluchyj, "Terabit Lightwave Networks: The Multihop Approach", *AT&T Technical Journal*, Vol. 66, Issue 6, pp 21-34, Nov./Dec. 1987.

[3] A. S. Acampora, "A Multichannel Multihop Local Lightwave Network", *IEEE GLOBECOM'87*, pp 1459-1467, 1987.

[4] A. S. Acampora, M. J. Karol and M. G. Hluchyj, "Multihop Lightwave Networks: A New Approach to Achieve Terabit Capabilities", *IEEE INFOCOM'88*, pp 46.1.1-46.1.7, 1988.

[5] A. Bondavalli, M. Conti, and E. Gregori, et al, "MAC Protocols for High-speed MANs: Performace Comparisons for a Family of Fasnet-based Protocols", *Computer Networks and ISDN Systems 18*, pp 97-113, Elsevier Science Publishers B.V. (North-Holland), 1990.

[6] W. W. Cao and O. W. W. Yang, "A Design Model for Tree-type Fiber Optics MAN", *Proc. of Canadian Conference on Electrical and Computer Engineering*, pp 45.1.1-45.1.4, 1991.

[7] M. Dècina, V. Trecordi and G. Zanolini, "Throughput and Packet Loss in Deflection Routing Multichannel-Metropolitan Area Networks", *IEEE GLOBECOM'91*, pp 34.1.1-34.1.9, 1991.

[8] D. Delisle and L. Pelamourgues, "B-ISDN and How It Works", *IEEE SPECTRUM*, pp 39-42, Aug. 1991.

[9] M. Gerla, G.-S. Wang and P. Rodrigues, "Buzz-Net: A Hybrid Token/Random Access LAN", *IEEE Journal on Selected Areas in Communications*, Vol. 5, No. 6, pp 977-988, July 1987.

[10] M. Gerla and L. Fratta, "Tree Structured Fiber Optics MANs", *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 6, pp 934-942, July 1988.

[11] M. Gerla, M. Kovačević and J. Bannister, "Multilevel Optical Networks", *ICC'91*, pp 340.4.1-340.4.5, 1992.

[12] I. M. I. Habbab, M. Kavehrad and C.-E. W. Sundberg, "Protocols for Very High-Speed Optical Local Area Networks Using a Passive Star Topology", *Journal of Light Wave Technology*, Vol. 5, No. 12, pp 1782-1794, Dec. 1987.

[13] J. Hayes, **Modelling and Analysis of Computer Communications Networks**, Plenum Press, New York, 1984.

[14] B. Hein, "An Optical Subcarrier System in a Passive Tree Shaped Network for Switched Services", *IEEE INFOCOM'89*, pp 46.5.1-46.5.5, 1989.

[15] M. G. Hluchyj and M. J. Karol, "ShuffleNet: An Application of Generalized Perfect Shuffles to Multihop Lightwave Networks", *IEEE INFOCOM'88*, pp 4B.4.1-4B.4.12, 1988.

[16] O. C. Ibe, P.-Y. Chen and P. H. Enslow, Jr., "Treenet: An Integrated Metropolitan Area Network", *IEEE INFOCOM'85*, pp 9.2.1-9.2.5, 1985.

[17] M. I. Irshid and M. Kavehrad, "A WDM Cross-Connected Star Multihop Optical Network", *ICC'91*, pp 349.1.1-349.1.5, 1992.

[18] A. E. Kamal, "A Semaphore-Based Access Protocol for Star Networks", *IEEE INFOCOM'87*, pp 11-18, 1987.

[19] I. P. Kaminow, "Photonic Multiple-Access Networks: Topologies", *AT&T Technical Journal*, pp 61-71, March/April 1989.

[20] M. J. Karol, "Optical Interconnection Using ShuffleNet Multihop Networks In Multi-Connected Ring Topologics", *SIGCOM'88*, pp 25-34, 1988.

[21] M. J. Karol and R. D. Gitlin, "High-Performance Optical Local and Metropolitan Area Networks: Enhancements of FDDI and IEEE 802.6 DQDB", *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 8, pp 1439-1448, Oct. 1990.

[22] L. Kleinock, **Queueing Theory Vol. 1: Theory**, Wiley, New York, 1975.

[23] R. W. Klessig, "Overview of Metropolitan Area Networks", *IEEE Communications Magazine*, Vol. 24, No. 1, pp 9-15, Jan. 1986.

[24] J.-F. P. Labourdette and A. S. Acampora, "Wavelength Agility in Multihop Lightwave Networks", *IEEE INFOCOM'90*, pp 1002-1029, 1990.

[25] M. E. Marhic, "Hierarchic and Combinational Star Couplers", *Optics Letters*, Vol. 9, No. 6, pp 368-370, Aug. 1984.

[26] J. W. Mark and O. W. W. Yang, "Design and Analysis of a Metropolitan Area Network: A Two-Center Tree Net", *Computer Networking Symposium*, pp 46-54, Nov. 1986.

[27] B. Materna, B. J. N. Vaughan and C. W. Britney, "Evolution from LAN and MAN Access Networks Towards the Integrated ATM Network", *IEEE INFOCOM'89*, pp 41.1.1-41.1.7, 1989.

[28] J. A. McEachern, "Gigabit Networking on the Public Transmission Network", *IEEE Communications Magazine*, pp 70-78, April 1992.

[29] P. K. McKinley and J. W. S. Liu, "Multicast Tree Construction in Bus-Based Networks", *Communications of the ACM*, Vol. 33, No. 1, pp 29-42, Jan. 1990.

[30] B. Mukherjee, "WDM-Based Local Lightwave Networks–Part II: Multihop Systems", *IEEE Network*, Vol. 6, No. 4, pp 20-31, July 1992.

[31] R. W. Newman, Z. L. Budrikis and J. L. Hullett, "The QPSX Man", *IEEE Communications Magazine*, Vol. 26, No. 4, pp 20-28, April 1988.

[32] C. S. Raghavendra, A. Avižienis and M. Ercegovac, "Fault-Tolerance in Binary Tree Architectures", *IEEE INFOCOM'83*, pp 360-364, 1983.

[33] I. Rubin and J. E. Baker, "Media Access Control for High-Speed Local Area and Metropolitan Area Communications Networks", *Proceedings of IEEE*, Vol. 78, No. 1, pp 168-205, Jan. 1990.

[34] F. W. Scholl and M. H. Coden, "Passive Optical Star Systems for Fiber Optic Local Area Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 6, pp 913-923, July 1988.

[35] T. Suda, S. Morris and T. Nguyen, "Tree LANs with Collision Avoidance: Protocol, Switch Architecture and Simulated Performance", *SIGCOM'88*, pp 155-164, 1988.

[36] A. S. Tanenbaum, **Computer Networks**, Second Edition, Prentice Hall, 1988.

[37] A. Tantawy, H. Meleis, and M. E. Zarki, et al, "Towards a High Speed Man Architecture", *IEEE INFOCOM'89*, pp 20.1.1-20.1.6, 1989.

[38] T. D. Todd, Z. Khurshid and A. M. Bignell, et al, "Photonic Multihop Bus Netwoks",*IEEE INFOCOM'91*, pp 9A.1.1-9A.1.10, 1991.

[39] C.-W. Tseng and B.-U. Chen, "D-Net, A New Scheme for High Data Rate Optical Local Area Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 1, No. 3, pp 493-499, April 1983.

[40] Z. Zhang and A. S. Acampora, "Analysis of Multihop Lightwave Networks", *IEEE INFOCOM'90*, pp 903.7.1-903.7.7, 1990.

Appendix A

Fixed Routing Algorithm for ShuffleNet

There are many possible fixed routing algorithms for ShuffleNet. We present one that is particularly simple to implement. We describe a distributed, fixed routing scheme that delivers packets from any given source to any given destination in the minimum number of hops possible for that source-destination pair. In routing packets from a source (c^s,r^s) to a destination (c^d,r^d), the routing decision at each intermediate NIU is based on one of the p-ary digits of the destination address contained in the arriving packet's header.

If we let D denote the number of columns between the source (c^s,r^s) and the destination (c^d,r^d), then

$$x = \begin{cases} (k + c^d - c^s) \bmod k & \text{if } c^d \neq c^s \\ k & \text{if } c^d = c^s \end{cases}$$

Since the column coordinate increases by one (mod k) each hop, D represents a lower bound on the number of hops required to route a packet from (c^s,r^s) to (c^d,r^d). However, it may not be possible to reach the destination in D hops. For example, in Figure 3.2 it takes three hops to route a packet from NIU 1 to NIU 7, even though NIU 7 is only one column from NIU 1 (i.e., $D = 1$).

The algorithm attempts to deliver the packet in D hops by routing successively according to the last D p-ary digits of r^d. In other words, from NIU

$$(c^s, r^s) = (c^s, r^s_{k-1} r^s_{k-2} \cdots r^s_0)$$

the packet is first routed to the NIU with address

$$((c^s + 1) \bmod k, r^s_{k-2} \cdots r^s_0 r^d_{D-1}),$$

and then to

$$((c^s + 2) \bmod k, r^s_{k-3} \cdots r^s_0 r^d_{D-1} r^d_{D-2}),$$

and so on, until it reaches the NIU with address

$$(c^d, r^s_{k-1-D} \cdots r^s_0 r^d_{D-1} \cdots r^d_0)$$

This corresponds to cyclically rotating in the destination's row coordinate one digit at a time. The NIU reached after D hops is the destination (c^d, r^d) if and only if

$$r^s_{k-1-D} = r^d_{k-1}, r^s_{k-2-D} = r^d_{k-2}, \cdots, \text{ and } r^s_0 = r^d_D.$$

Otherwise, the packet is not in the destination's row r^d when it reaches column c^d, and the packet takes k more hops (one hop in each of the k columns of the cylindrical connectivity graph) to reach its destination (c^d, r^d). So, a packet reaches its destination in the minimum number of hops, which may be either D or $D + k$.

The following routing decisions are made when an arbitrary NIU $(\hat{c}, \hat{r})$ receives a packet with destination address (c^d, r^d). If $(c^d, r^d) = (\hat{c}, \hat{r})$, the packet is accepted by the local user port. Otherwise, the packet is routed to $((\hat{c} + 1) \bmod k, \hat{r}_{k-2}, \cdots, \hat{r}_0 r^d_{X-1})$, where X denotes the number of columns between the current location $(\hat{c}, \hat{r})$ and the destination (c^d, r^d). That is

$$X = \begin{cases} (k + c^d - \hat{c}) \bmod k & \text{if } c^d \neq \hat{c} \\ k & \text{if } c^d = \hat{c} \end{cases}$$

Appendix B

Simulation Program

```
                                                                  ***********************************************************
&**********************************************************
&                                          Program name: Shufflenet
&****************************************************************************************************************************
                                                                                                         
&****************************************************************************************************************************
& This program is to get the delay performance for Shufflenet.
& The fixed routing scheme introduced for Shufflenet in appendex A is used.
& This is a general program suitable for any number of stations N=K*P**K.
&****************************************************************************************************************************

/CONTROL/
         OPTION=NSOURCE,NRESULT;

/DECLARE/
REAL                                        & PACKET ARRIVAL TIME IN A SOURCE
          T_ARRIVE,                         & PACKET SERVIVE TIME IN A STATION
          T_SERVICE,                        & ARRIVAL RATE TO A STATION
          AR_RATE,                          & ACTUAL ARRIVAL RATE TO A STATION
          AR1_RATE,                         & UTINIZATION
          RHO,                              & ACTUAL UTILIZATION
          RHO1,                             & MEAN DELAY OF PACKET TRANSMISSION FROM ANALYSIS
          M_DEL                             & MEAN DELAY OF PACKET TRANSMISSION FROM SIMULATION
          M_DEL',                           & VARIANCE OF DELAY FROM ANALYSIS
          V_DEL,                            & VARIANCE OF DELAY FROM SIMULATION
          V_DEL',                           & MEAN WAITING TIME FROM ANALYSIS
          M_WA,
          PROP,                             & AVARAPE NUNBER OF HOPS OF SHUFFLENET
          M_HOP;



 INTEER
          I,J,                              & NUMBER OF STATIONS IN THE SHUFFLENET
          N_STATION=24,                     & COLUMNS IN THE SHUFFLENET
          K=3,                              & CONNECTIVITY OF EACH STATION IN THE SHUFFLENET
          P=2;


 CUSTOMER INTEPER
          D1,                               & DESTINATION ADDRESS OF A PACKET
          DES_ADD,                          & THE COLUMN THAT THE DESTINATION STATION BELONGS TO
          COLUMN_DES,                       & THE ROW THAT THE DESTINATION STATION BELONGS TO
          DBN_DES(K);


 QUEUE INTEPER
          D2,X,RESD,
          COLUMN, DBN(K),                   & ADDRESS OF A SOURCE STATION
          STA_ADD,                          & THE COLUMN THAT THE SOURCE STATION BELONGS TO
          COLUMN_STA,                       & THE ROW THAT THE SOURCE STATION BELONGS TO
          DBN_STA(K);                       & IDENTIFICATION OF A QUEUE
          IDEN


  QUEUE                                     & SOURCE TO GENERATE PACKETS
          SOURCE(N),                        & SERVICE QUEUE
          Q(N);
```

```
&*****************************************************************************************
&             THIS PROGRAM IS TO REALIZE THE SOURCE TO GENERATE PACKETS
&*****************************************************************************************

/STATION/NAME=SOURCE(1 STEP 1 UNTIL K*P**K);
TYPE=SOURCE;
SERVICE=BEGIN
          EXP(T_ARRIVE);
          DES_ADD:=RINT(1,K*P**K);                & THE DETINATION ADDRESS IS GENERATED
          COLUMN_DES:=(DES_ADD+P**K-1)/(P**K);    & DETERMINE THE COLUMN OF THE DETINATION
          D1:=DES_ADD-1-((DES_ADD-1)/(P**K))*(P**K);
          FOR I:=1 STEP 1 UNTIL K DO
          BEGIN
            DBN_DES(I):=D1-(D1/P)*P;              & DETERMINE THE ROW OF THE DESTINATION
            D1:=D1/P;
          END;
          IF DES_ADD=IDEN
          THEN TRANSIT(OUT)
          ELSE TRANSIT(Q(IDEN));
        END;

&*****************************************************************************************
&         THIS PROGRAM IS TO REALINZE THE STATION IN THE SHUFFLENET. THE
&         ROUTINE IS DETERMINED ACCORDINT TO THE ALGORITHEM
******************************************************************************************

/STATION/NAME=Q(1 STEP 1 UNTIL K*P**K);
     SERVICE=BEGIN
               CST(T_SERVICE);
               COLUMN:=(COLUMN_STA+1)-((COLUMN_STA+1)/(K+1))*K;    & DETERMINE THE COLUMN
               IF COLUMN_DES=COLUMN_STA                            & OF THIS STATION
               THEN X:=K
               ELSE X:=(K+COLUMN_DES-COLUMN_STA)-
                       ((K+COLUMN_DES-COLUMN_STA)/K)*K;
               DBN(1):=DBN_DES(X);
               FOR I:=1 STEP 1 UNTIL K-1
               DO DBN(I+1):=DBN_STA(I);
               RESD:=0;
               FOR I:=1 STEP 1 UNTIL K
               DO RESD:=DBN(I)*(P**(I-1))+RESD;
               STA_ADD:=(COLUMN-1)*(P**K)+RESD+1;
               IF DES_ADD=STA_ADD              & IF THE PACKET ARRIVES TO THE DESTINATION
               THEN TRANSIT(OUT)               & RECEIVE THE PACKET
               ELSE TRANSIT(Q(STA_ADD));       & OTHERWISE TRANSMIT TO THE NEXT HOP
             END;

&*****************************************************************************************

/CONTROL/TMAX=150000;
         MARPINAL=Q(6);
         ACCURACY=ALL QUEUE;
         ESTIM=SPECTRAL;

&*****************************************************************************************
&             THE FOLLOWING PROGROM IS FOR THE EXECUTION AND RESULTS
&*****************************************************************************************

/EXEC/ BEGIN
       FOR J:=1 STEP 1 UNTIL K*(P**K)
       DO BEGIN
            SOURCE(J).IDEN:=J;                    & GIVE A IDENTIFICATION TO A SOURCE
            Q(J).IDEN:=J;                         & GIVE A IDENTIFICATION TO A QUEUE
            Q(J).COLUMN_STA:=(J+P**K-1)/(P**K);   & DETERMINE THE COLUMN OF A QUEUE
            Q(J).D2:=J-((J-1)/(P**K))*(P**K)-1;   & DETERMINE THE ROW OF A QUEUE
            FOR I:=1 STEP 1 UNTIL K DO
            BEGIN
```

```
            Q(J).DEN_STA(I):=Q(J).D2-(Q(J).D2/P)*P;
            Q(J).D2:=Q(J).D2/P;
          END;
        END;
PRINT("*************************************************************");
PRINT("*  RHO          *  M_DEL      *  M_DEL'     *    V_DEL     *  V_DEL'       ");
PRINT("*************************************************************");
        FOR J:= 1 STEP 1 UNTIL 9 DO
        BEGIN                                               & ARRIVAL TIME OF A PACKET
          T_ARRIVE:= 1.75/J;                                & SERVICE TIME OF A PACKET
          T_SERVICE:=1.;
          SIMUL;
          PROP:=(K*P**K-1)/(K*P**K);
          M_HOP:=(((K*P**K)*(P-1)*(3*K-1)-2*K*(P**K-1)))/ & MEAN HOP OF A PACKET TRANSMITTE
                      (2*(P-1)*(K*P**K-1));                   & FROM A SOURCE TO A DETINATION
          AR_RATE:=PROP/T_ARRIVE;                  & MEAN PACKET ARRIVAL RATE TO A STATION
          AR1_RATE:=M_HOP*AR_RATE;                 & ACTUAL MEAN PACKET ARRIVAL RATE TO A STATION
          RHO:=AR_RATE*T_SERVICE;                  & UTILIZATION OF A PACKET
          RHO1:=AR1_RATE*T_SERVICE;                & ACTUAL UTILIZATION OF A PACKET
          M_WA:=RHO1*T_SERVICE/(2.*(1 -RHO1));        & MEAN WAITING TIME OF A PACKET
          M_DEL:=M_WA +T_SERVICE;               & MEAN TOTAL DELAY OF A PACKET FROM A ANALYSIS
          M_DEL':=MRESPONSE(Q(6));              & MEAN TOTAL DELAY OF A PACKET FROM A SIMULATION
          V_DEL:=(M_WA)**2 +(AR1*(T_SERVICE**3))/(3.*(1 -RHO1));
          V_DEL':=VRESPONSE(Q(6));                          & VARIANCE OF DELAY FROM SIMULATION
          PRINT("*",RHO,"*",M_DEL,"*",M_DEL',"*",V_DEL,"*",V_DEL',"*");
      END;
PRINT("*************************************************************");
END;

/END/
```

```
&*********************************************************************************
&                          PROPRAM NAME: SHU.STDM
&*********************************************************************************
&*********************************************************************************
&  This program is to achieve the delay and throughput performance for SHU.STDM Model.
&*********************************************************************************


/CONTROL/
          OPTION=NSOURCE,NRESULT;

/DECLARE/
INTEGER
                 I,J,L,
                 K=2,               & COLUMNS IN THE SHUFFLENET IN THE UPPER LEVEL
                 P=2,               & CONNECTIVITY OF EACH STATION IN THE SHUFFLENET
                 N_STATION=8,       & NUMBER OF STATIONS IN THE LOWER LEVEL
                 N_ACT_BUS          & NUMBER OF ACTIVE BUS IN THE UPPER LEVEL



REAL
                 T_ARRIVE(N_STATION,N_ACT_BUS),  & PACKET ARRIVAL TIME IN A SOURCE
                 T_SLOT,                    & PACKET TRANSMISSION TIME IN STDM
                 T_FRAME,                   & FRAME DURATION IN STDM
                 T_SERVICE,                 & SERVICE TIME FOR A PACKET IN THE UPPER LEVEL
                 AR_RATE,                   & ARRIVAL RATE TO A STATION
                 RHO1,                      & UTINIZATION OF A STATION
                 M_DEL11,                   & MEAN QUEUEING DELAY IN A STATION
                 M_DEL12,                   & MEAN QUEUEING DELAY IN THE ACTIVE BUS
                 INTRA_D,                   & MEAN TOTAL DELAY OF THE LOCAL TRAFFIC FROM ANALYSIS
                 INTRA_D',                  & MEAN TOTAL DELAY OF THE LOCAL TRAFFIC FROM SIMULATION
                 INTER_D,                   & MEAN TOTAL DELAY OF THE OUTGOING TRAFFIC FROM ANALYSI
                 INTER_D';                  & MEAN TOTAL DELAY OF THE OUTGOING TRAFFIC FROM SIMULAT


 CUSTOMER INTEPER
                 D1,                        & THE DESTINATION BUS ADDRESS OF A PACKET
                 FOR_ADD,                   & THE DESTINATION STATION ADDRESS OF A PACKET
                 LOC_ADD,                   & THE COLUMN THAT THE DESTINATION STATION BELONGS TO
                 COLUMN_DES,                & THE ROW THAT THE DESTINATION STATION BELONGS TO
                 DBN_DES(K);


 QUEUE INTGER
                 STAT,                         & THE LOCAL ADDRESS OF A QUEUE
                 BUS,                          & THE OUTGOING ADDRESS OF A QUEUE
                 COLUMN_STA,                   & THE COLUMN THAT THE SOURCE STATION BELONGS TO
                 DBN_STA(K),                   & THE ROW THAT THE SOURCE STATION BELONGS TO
                 D2,X,COLUMN,
                 RESD,STAADD,DBN(K);

 QUEUE
                 S(N_STATION,N_ACT_BUS),           & SOURCE QUEUE
                 Q(N_STATION,N_ACT_BUS),           & STATION QUEUE
                 MONITOR(N_ACT_BUS),               & ACTIVE NODE QUEUE
                 CLOCK(N_ACT_BUS),                 & THE CLOCK USED FOR SYNCHONIZATION IN STDM
                 CT,AT;

 FLAG
                 FG(N_STATION,N_ACT_BUS);          & FLAG
```

```
&*****************************************************************************************
&              THIS PROGRAM IS TO REALIZE THE SOURCE TO GENERATE PACKETS
&*****************************************************************************************

/STATION/NAME=S(1 STEP 1 UNTIL N_STATION, 1 STEP 1 UNTIL K*P**K);
TYPE=SOURCE;
SERVICE=BEGIN                                    & THE DISTRIBUTION OF PACKET INTERARRIVAL TIME
          EXP(T_ARRIVE(STAT,BUS));                    & THE FOREIGN ADDRESS OF A PACKET
          FOR_ADD:=RINT(1,K*P**K);                    & THE LOCAL ADDRESS OF A PACKET
          LOC_ADD:=RINT(1,N_STATION);                     & DETERMINE THE COLUMN
          COLUMN_STATION:=(FOR_ADD+P**K-1)/(P**K);
          D1:=FOR_ADD-1-((FOR_ADD-1)/(P**K))*(P**K);
          FOR I:=1 STEP 1 UNTIL K DO
          BEGIN                                           & DETERMINE THE ROW
            DBN_STATION(I):=D1-(D1/P)*P;
            D1:=D1/P;
          END;
          IF FOR_ADD=BUS
          THEN BEGIN
                 IF LOC_ADD=STAT
                 THEN TRANSIT(OUT)
                 ELSE BEGIN
                        P(CT);
                        TRANSIT(Q(STAT,BUS));
                      END;
               END
           ELSE BEGIN
                  P(AT);
                  TRANSIT(Q(STAT,BUS));
                END;
         END;

&*****************************************************************************************
&       THIS PROGRAM IS TO REALINZE THE STDM IN A BUS. THE FLAG IS USED AS A PERMISSION FOR
&       A STATION TO TRANSMIT PACKETS. WHENEVER THE FLAG IN A STATION IS 1, THE STATION CAN
&       TRANSMIT PACKETS
&*****************************************************************************************


/STATION/NAME=Q(1 STEP 1 UNTIL N_STATION, 1 STEP 1 UNTIL K*P**K);
SERVICE=BEGIN
          WAIT (FG(STAT,BUS));     & WAITING FOR THE FLAG
          RESET (FG(STAT,BUS));    & RESET THE FLAG TO 1
          CST(T_SLOT);             & THE SLOT NAME IS CONSTANT
          IF FOR_ADD=BUS           & IF THE DESTINATION OF THIS PACKET IS FOR THIS BUS
          THEN BEGIN
                 V(CT);            & TRACE THIS PACKET
                 TRANSIT(OUT);     & RECEIVE THIS PACKET
               END
          ELSE TRANSIT(MONITOR(BUS));  & OTHERWISE TRANSMIT THIS PACKET TO THE ACTIVE NODE
        END;
&*****************************************************************************************

/STATION/NAME=CT;                  & RESOURCE QUEUE TO TRACE THE PACKET
TYPE=RESOURCE,INFINIT
&*****************************************************************************************

/STATION/NAME=AT;                  & RESOURCE QUEUE TO TRACE THE PACKET
TYPE=RESOURCE,INFINITE;
&*****************************************************************************************
```

```
&       THIS IS A CLOCK TO MODEL THE GENERATION OF A FRAME IN STDM.
&*****************************************************************************************

/STATION/NAME=CLOCK(1 STEP 1 UNTIL K*P**K);
         INIT=1;
         SERVICE=BEGIN
                  FOR I:=1 STEP 1 UNTIL N_STATION
                  DO BEGIN
                      SET (FG(I,BUS));
                      IF Q(I,BUS).NB=0
                      THEN RESET (FG(I,BUS));
                      CST(T_SLOT);
                      END;
                  END;
           TRANSIT=CLOCK(BUS);

&*****************************************************************************************
&       THIS IS TO REALIZE THE ACTIVE NODE ON THE TOP OF THE BUS. IF THE PACKET IS
&       A LOCAL TRAFFIC,THE ACTIVE NODE TRANSMITS THE PACKET TO THE BUS UNDER IT.
&       IF THE PACKET IS A OUTGOING TRAFFIC, THE ACTIVE NODE WILL TRANSMIT THE PACKET
&       TO ANOTHER ACTIVE NODE USING THE SHUFFLENET
&*****************************************************************************************


/STATION/NAME=MONITOR(1 STEP 1 UNTIL K*P**K);
SERVICE=BEGIN
          CST(T_SERVICE);          & THE SERVICE TIME IN THE ACTIVE NODE IS CONSTANT
          IF FOR_ADD=BUS           & IF THE PACKET IS LOCAL TRAFFIC
          THEN BEGIN
                 V(AT);            & TRACE THIS PACKET
                 TRANSIT(OUT);     & TRANSMIT THIS PACKET TO THE BUS
               END
          ELSE BEGIN               & OTHERWISE CALCULATE THE ROUTE USING SHUFFLENET
                 COLUMN:=(COLUMN_STA+1)-((COLUMN_STA+1)/(K+1))*K;
                 IF COLUMN_STATION=COLUMN_STA
                 THEN X:=K+1
                 ELSE X:=ABS(COLUMN_STATION-COLUMN_STA)+1;
                 DBN(1):=DBN_STATION(X-1);
                 FOR I:=1 STEP 1 UNTIL K-1
                 DO DBN(I+1):=DBN_STA(I);
                 RESD:=0;
                 FOR I:=1 STEP 1 UNTIL K
                 DO RESD:=DBN(I)*(P**(I-1))+RESD;
                 STAADD:=(COLUMN-1)*(P**K)+RESD+1;
                 TRANSIT(MONITOR(STAADD));  & TRANSMIT IT TO ANOTHER ACTIVE NODE
               END;
         END;
&*****************************************************************************************

/CONTROL/TMAX=10000000;
         MARPINAL=Q(4,3);
         ACCURACY=ALL QUEUE;

&*****************************************************************************************
&             THE FOLLOWING PROGROM IS FOR THE EXECUTION AND RESULTS
&*****************************************************************************************

/EXEC/
BEGIN
  FOR L:=1 STEP 1 UNTIL K*P**K DO
  BEGIN
    FOR I:=1 STEP 1 UNTIL N_STATION DO
    BEGIN                          & GIVE THE STATION IDENTIFICATION TO A SOURCE
      S(I,L).STAT:=I;
```

```
    S(I,L).BUS:=L;                 & GIVE THE BUS IDENTIFICATION TO A SOURCE
    Q(I,L).STAT:=I;                & GIVE THE STATION IDENTIFICATION TO A QUEUE
    Q(I,L).BUS:=L;                 & GIVE THE BUS IDENTIFICATION TO A QUEUE
  END;
  MONITOR(L).BUS:=L;               & GIVE A IDENTIFICATION TO A ACTIVE NODE
  CLOCK(L).BUS:=L;                 & GIVE A IDENTIFICATION TO A CLOCK
  MONITOR(L).COLUMN_STA:=(L+P**K-1)/(P**K);
  MONITOR(L).D2:=L-((L-1)/(P**K))*(P**K)-1;
  FOR J:=1 STEP 1 UNTIL K DO
  BEGIN
    MONITOR(L).DBN_STA(J):=MONITOR(L).D2-(MONITOR(L).D2/P)*P;
    MONITOR(L).D2:=MONITOR(L).D2/P;
  END;
END;
PRINT("*******************************************************************");
PRINT("*   RHO1     *   M_DEL11  *   M_DEL12  *  INTER_D   *  INTER_D'   *");
PRINT("*******************************************************************");
FOR I:=1 STEP 1 UNTIL N_STATION DO
  BEGIN
    FOR J:=2 STEP 1 UNTIL N_ACT_BUS
    DO T_ARRIVE(I,J):=4000.;.
  END;
FOR I:=2 STEP 1 UNTIL N_STATION DO
  BEGIN
    FOR J:=1 STEP 1 UNTIL N_ACT_BUS
    DO T_ARRIVE(I,J):=4000.;
  END;
FOR J:=1 STEP 1 UNTIL 9 DO              & CHANPE INTERARRIVE TIME;
  BEGIN
    T_ARRIVE(1,1):=40000./(2**J);       & MEAN INTERARRIVAL TIME
    T_SLOT:=5.;                         & MEAN SERVICE TIME FOR THE STATION
    T_FRAME:=5.*8.;                     & ACTUAL SERVICE TIME FOR THE STATION
    T_STOT:=5.;                         & MEAN SERVICE TIME FOR THE MONITOR
    SIMUL;
    AR_RATE:=1./T_ARRIVE(1,1);          & ARRIVE RATE TO A STATION
    RHO1:=AR_RATE*T_FRAME;              & UTILIZATION OF A STATION
    M_DEL11:=MRESPONSE(Q(1,1));         & MEAN QUEUEING DELAY IN A STATION
    M_DEL12:=MRESPONSE(MONITOR(1));     & MEAN QUEUEING DELAY IN THE ACTIVE BUS
    INTRA_D':=MRESPONSE(CT);            & MEAN TOTAL DELAY OF THE LOCAL TRAFFIC
    INTER_D':=MRESPONSE(AT);            & MEAN TOTAL DELAY OF THE OUTGOING TRAFFIC

    PRINT("*",RHO1,"*",INTRA_D,"*",INTRA_D',"*",INTER_D,"*",INTER_D',"*");
  END;
 PRINT("*******************************************************************");
END;

/END/
```

```
                                                                                  ***************************
&****************************************************************************************************
&                                PROGRAM NAME: Shu.Shul  ****************************************
&****************************************************************************************************
                                                                                                  *
   &*********************************************************************************************
   &    This program is to get the delay performance of Shu.shul.                                *
   &    The physical configution of the upper leyer is a tree and the                            *
   &    lower leyer is a bus. The logic configuration of both upper                              *
   &    and lower leyer are SHUFFLENET. There are 8 nodes in the upper                           *
   &    leyer and 8 nodes in each bus.  **************************************************************
   &*********************************************************************************************
/CONTROL/
        OPTION=NSOURCE,NRESULT;

/DECLARE/
REAL
        T_ARRIVE,                   & PACKET ARRIVAL TIME IN A SOURCE
        T_SERVIVE1,                 & PACKET SERVIVE TIME IN A STATION
        T_SERVIVE1',
        T_SERVIVE2,                 & PACKET SERVIVE IN A ACTIVE NODE
        AR1_RATE,                   & ARRIVAL RATE TO A STATION
        RHO1,                       & UTINIZATION IN A STATION
        RHO2,                       & UTINIZATION IN A ACTIVE NODE
        M_NB1,                      & MEAN NUMBER OF PACKETS IN A STATION FROM ANALYSIS
        M_NB1',                     & MEAN NUMBER OF PACKETS IN A STATION FROM SIMULATION
        M_DEL1,                     & MEAN QUEUEING DELAY IN A STATION FROM ANALYSIS
        M_DEL1',                    & MEAN QUEUEING DELAY IN A STATION FROM SIMULATION
        AR2_RATE,                   & ARRIVAL RATE TO A ACTIVE NODE
        M_NB2,                      & MEAN NUMBER OF PACKETS IN A ACTIVE NODE FROM ANALYSIS
        M_NB2',                     & MEAN NUMBER OF PACKETS IN A ACTIVE NODE FROM SIMULATION
        M_DEL2,                     & MEAN QUEUEING DELAY IN A ACTIVE NODE FROM ANALYSIS
        M_DEL2',                    & MEAN QUEUEING DELAY IN A ACTIVE NODE FROM SIMULATION
        AR_RATE,                    & TOTAL ARRIVAL RATE
        M_DEL,                      & MEAN TOTAL DELAY OF A PACKET TRANSMISSION FROM ANALYSIS
        M_DEL';                     & MEAN TOTAL DELAY OF A PACKET TRANSMISSION FROM SIMULATION


 INTEGER
        I=8,J,K;


 CUSTOMER INTEGER
        FOR_ADD,                    & THE DESTINATION BUS ADDRESS OF A PACKET
        LOC_ADD;                    & THE DESTINATION STATION ADDRESS OF A PACKET


 QUEUE INTEGER
        IDEN;                       & IDENTIFICATION OF A QUEUE


 QUEUE
        QA(I),QB(I),QC(I),QD(I),QE(I),QF(I),QG(I),QH(I),CT,
        SA(I),SB(I),SC(I),SD(I),SE(I),SF(I),SG(I),SH(I),MONITOR(I);
                                                                   ***********************************
 &****************************************************************************************************
 & THIS PROGRAM IS TO REALIZE THE FIRST SOURCE STATION IN ONE BUS ************************************
 &****************************************************************************************************

 /STATION/NAME=SA(1 STEP 1 UNTIL 8);
 TYPE=SOURCE;
 SERVICE=BEGIN
         EXP( T_ARRIVE);
         FOR_ADD:=RINT(1,8);            & THE FOREIGN ADDRESS OF A PACKET
```

```
         LOC_ADD:=RINT(1,8);              & THE LOCAL ADDRESS OF A PACKET
         IF FOR_ADD=IDEN
         THEN BEGIN
                IF LOC_ADD=1     & IF THE DESTINATION OF THIS PACKET IS FOR THIS BUS
                THEN TRANSIT(OUT)     & RECEIVE THIS PACKET
                ELSE BEGIN
                        P(CT);            & START TRACE THIS PACKET
                        TRANSIT(QA(IDEN));
                      END;
              END
         ELSE BEGIN
                P(CT);
                TRANSIT(QA(IDEN));
              END;
       END;
&**********************************************************************
&  THIS PROGRAM IS TO REALIZE THE FIRST QUEUE STATION IN ONE BUS
&**********************************************************************

/STATION/NAME=QA(1 STEP 1 UNTIL 8);
SERVICE=BEGIN
          CST( T_SERVIVE1);
          IF FOR_ADD=IDEN
          THEN BEGIN
                  IF ((LOC_ADD=5) OR (LOC_ADD=6))
                  THEN BEGIN
                         V(CT);         & END TRACE THIS PACKET
                         TRANSIT(OUT);
                       END
                  ELSE BEGIN
                       IF ((LOC_ADD=2) OR (LOC_ADD=8))
                       THEN TRANSIT(QE(IDEN))
                       ELSE TRANSIT(QF(IDEN));
                       END;
                END
           ELSE TRANSIT(MONITOR(IDEN));
          END;
 &*******************************************************************************************
 &  THIS PROGRAM IS TO REALIZE THE SECOND SOURCE STATION IN ONE BUS
 &*******************************************************************************************

 /STATION/NAME=SB(1 STEP 1 UNTIL 8);
 TYPE=SOURCE;
 SERVICE=BEGIN
         EXP( T_  ARRIVE);
         FOR_ADD:=RINT(1,8);
         LOC_ADD:=RINT(1,8);
         IF FOR_ADD=IDEN
         THEN BEGIN
               IF LOC_ADD=2
               THEN TRANSIT(OUT)
               ELSE BEGIN
                     P(CT);
                     TRANSIT(QB(IDEN));
                     END;
               END
          ELSE BEGIN
               P(CT);
               TRANSIT(QB(IDEN));
               END;
          END;

   &**********************************************************************
```

```
&  THIS PROGRAM IS TO REALIZE THE FIRST QUEUE STATION IN ONE BUS
&*********************************************************************....

/STATION/NAME=QB(1 STEP 1 UNTIL 8);
SERVICE=BEGIN
        CST( T_SERVIVE1);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF ((LOC_ADD=7) OR (LOC_ADD=8))
             THEN BEGIN
                  V(CT);
                  TRANSIT(OUT);
                  END
             ELSE BEGIN
                  IF ((LOC_ADD=1) OR (LOC_ADD=6))
                  THEN TRANSIT(QG(IDEN))
                  ELSE TRANSIT(QH(IDEN));
                  END;
              END
          ELSE TRANSIT(MONITOR(IDEN));
          END;
&*****************************************************************************************

/STATION/NAME=SC(1 STEP 1 UNTIL 8);
TYPE=SOURCE;
SERVICE=BEGIN
        EXP( T_  ARRIVE);
        FOR_ADD:=RINT(1,8);
        LOC_ADD:=RINT(1,8);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF LOC_ADD=3
             THEN TRANSIT(OUT)
             ELSE BEGIN
                  P(CT);
                  TRANSIT(QC(IDEN));
                  END;
             END
        ELSE BEGIN
             P(CT);
             TRANSIT(QC(IDEN));
             END;
        END;

/STATION/NAME=QC(1 STEP 1 UNTIL 8);
SERVICE=BEGIN
        CST( T_SERVIVE1);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF ((LOC_ADD=5) OR (LOC_ADD=6))
             THEN BEGIN
                  V(CT);
                  TRANSIT(OUT);
                  END
             ELSE BEGIN
                  IF ((LOC_ADD=4) OR (LOC_ADD=7))
                  THEN TRANSIT(QF(IDEN))
                  ELSE TRANSIT(QE(IDEN));
                  END;
              END
          ELSE TRANSIT(MONITOR(IDEN));
          END;
&*****************************************************************************************

/STATION/NAME=SD(1 STEP 1 UNTIL 8);
```

```
TYPE=SOURCE;
SERVICE=BEGIN
        EXP( T_  ARRIVE);
        FOR_ADD:=RINT(1,8);
        LOC_ADD:=RINT(1,8);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF LOC_ADD=4
             THEN TRANSIT(OUT)
             ELSE BEGIN
                  P(CT);
                  TRANSIT(QD(IDEN));
                  END;
             END
        ELSE BEGIN
             P(CT);
             TRANSIT(QD(IDEN));
             END;
        END;


/STATION/NAME=QD(1 STEP 1 UNTIL 8);
SERVICE=BEGIN
        CST( T_SERVIVE1);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF ((LOC_ADD=7) OR (LOC_ADD=8))
             THEN BEGIN
                  V(CT);
                  TRANSIT(OUT);
                  END
             ELSE BEGIN
                  IF ((LOC_ADD=3) OR (LOC_ADD=5))
                  THEN TRANSIT(QH(IDEN))
                  ELSE TRANSIT(QG(IDEN));
                  END;
              END
         ELSE TRANSIT(MONITOR(IDEN));
         END;
&*******************************************************************************************

/STATION/NAME=SE(1 STEP 1 UNTIL 8);
TYPE=SOURCE;
SERVICE=BEGIN
        EXP( T_  ARRIVE);
        FOR_ADD:=RINT(1,8);
        LOC_ADD:=RINT(1,8);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF LOC_ADD=5
             THEN TRANSIT(OUT)
             ELSE BEGIN
                  P(CT);
                  TRANSIT(QE(IDEN));
                  END;
             END
        ELSE BEGIN
             P(CT);
             TRANSIT(QE(IDEN));
             END;
        END;

/STATION/NAME=QE(1 STEP 1 UNTIL 8);
SERVICE=BEGIN
        CST( T_SERVIVE1);
        IF FOR_ADD=IDEN
```

```
        THEN BEGIN
             IF ((LOC_ADD=1) OR (LOC_ADD=2))
             THEN BEGIN
                  V(CT);
                  TRANSIT(OUT);
                  END
             ELSE BEGIN
                  IF ((LOC_ADD=4) OR (LOC_ADD=6))
                  THEN TRANSIT(QA(IDEN))
                  ELSE TRANSIT(QB(IDEN));
                  END;
              END
         ELSE TRANSIT(MONITOR(IDEN));
         END;
&*****************************************************************************************

/STATION/NAME=SF(1 STEP 1 UNTIL 8);
TYPE=SOURCE;
SERVICE=BEGIN
        EXP( T_   ARRIVE);
        FOR_ADD:=RINT(1,8);
        LOC_ADD:=RINT(1,8);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF LOC_ADD=6
             THEN TRANSIT(OUT)
             ELSE BEGIN
                  P(CT);
                  TRANSIT(QF(IDEN));
                  END;
             END
        ELSE BEGIN
             P(CT);
             TRANSIT(QF(IDEN));
             END;
        END;


/STATION/NAME=QF(1 STEP 1 UNTIL 8);
SERVICE=BEGIN
        CST( T_SERVIVE1);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF ((LOC_ADD=3) OR (LOC_ADD=4))
             THEN BEGIN
                  V(CT);
                  TRANSIT(OUT);
                  END
             ELSE BEGIN
                  IF ((LOC_ADD=2) OR (LOC_ADD=5))
                  THEN TRANSIT(QC(IDEN))
                  ELSE TRANSIT(QD(IDEN));
                  END;
              END
         ELSE TRANSIT(MONITOR(IDEN));
         END;
&*****************************************************************************************

/STATION/NAME=SG(1 STEP 1 UNTIL 8);
TYPE=SOURCE;
SERVICE=BEGIN
        EXP( T_   ARRIVE);
        FOR_ADD:=RINT(1,8);
        LOC_ADD:=RINT(1,8);
        IF FOR_ADD=IDEN
        THEN BEGIN
```

```
             IF LOC_ADD=7
             THEN TRANSIT(OUT)
             ELSE BEGIN
                  P(CT);
                  TRANSIT(QG(IDEN));
                  END;
             END
        ELSE BEGIN
             P(CT);
             TRANSIT(QG(IDEN));
             END;
        END;


/STATION/NAME=QG(1 STEP 1 UNTIL 8);
SERVICE=BEGIN
        CST( T_SERVIVE1);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF ((LOC_ADD=1) OR (LOC_ADD=2))
             THEN BEGIN
                  V(CT);
                  TRANSIT(OUT);
                  END
             ELSE BEGIN
                  IF ((LOC_ADD=3) OR (LOC_ADD=8))
                  THEN TRANSIT(QB(IDEN))
                  ELSE TRANSIT(QA(IDEN));
                  END;
               END
          ELSE TRANSIT(MONITOR(IDEN));
          END;
&**********************************************************************************************

/STATION/NAME=SH(1 STEP 1 UNTIL 8);
TYPE=SOURCE;
SERVICE=BEGIN
        EXP( T_  ARRIVE);
        FOR_ADD:=RINT(1,8);
        LOC_ADD:=RINT(1,8);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF LOC_ADD=8
             THEN TRANSIT(OUT)
             ELSE BEGIN
                  P(CT);
                  TRANSIT(QH(IDEN));
                  END;
             END
        ELSE BEGIN
             P(CT);
             TRANSIT(QH(IDEN));
             END;
        END;


/STATION/NAME=QH(1 STEP 1 UNTIL 8);
SERVICE=BEGIN
        CST( T_SERVIVE1);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF ((LOC_ADD=3) OR (LOC_ADD=4))
             THEN BEGIN
                  V(CT);
                  TRANSIT(OUT);
                  END
```

```
              ELSE BEGIN
                   IF ((LOC_ADD=1) OR (LOC_ADD=7))
                   THEN TRANSIT(QD(IDEN))
                   ELSE TRANSIT(QC(IDEN));
                   END;
               END
           ELSE TRANSIT(MONITOR(IDEN));
           END;
&************************************************************************************************
   &**********************************************************************
   &  The monitors are the intermediate nodes for interconnecting buses.*
   &  There is a monitor on the top of each bus. When the packet is    *
   &  going to outside of the bus, it is going through the monitors to  *
   &  reach the destination. The logic connection of the monitors is    *
   &  the 8 node shufflenet.
   &**********************************************************************

/STATION/NAME=MONITOR(1);
 SERVICE=BEGIN
         CST( T_SERVIVE2);
         IF FOR_ADD=1             & IF THE PACKET IS LOCAL TRAFFIC
         THEN BEGIN
              V(CT);              & END OF TRACE THE PACKET
              TRANSIT(OUT);       & TRANSMIT THIS PACKET TO THE BUS
              END
         ELSE BEGIN
              IF ((FOR_ADD=2) OR (FOR_ADD=5) OR (FOR_ADD=8))
              THEN TRANSIT(MONITOR(5))
              ELSE TRANSIT(MONITOR(6));
              END;
         END;
 &***********************************************************************************************

 /STATION/NAME=MONITOR(2);
  SERVICE=BEGIN
          CST( T_SERVIVE2);
          IF FOR_ADD=2
          THEN BEGIN
               V(CT);
               TRANSIT(OUT);
               END
          ELSE BEGIN
               IF ((FOR_ADD=1) OR (FOR_ADD=6) OR (FOR_ADD=7))
               THEN TRANSIT(MONITOR(7))
               ELSE TRANSIT(MONITOR(8));
               END;
          END;
  &**********************************************************************************************

  /STATION/NAME=MONITOR(3);
   SERVICE=BEGIN
           CST( T_SERVIVE2);
           IF FOR_ADD=3
           THEN BEGIN
                V(CT);
                TRANSIT(OUT);
                END
           ELSE BEGIN
                IF ((FOR_ADD=4) OR (FOR_ADD=6) OR (FOR_ADD=7))
                THEN TRANSIT(MONITOR(6))
                ELSE TRANSIT(MONITOR(5));
                END;
           END;

   &*********************************************************************************************
```

```
/STATION/NAME=MONITOR(4);
 SERVICE=BEGIN
         CST( T_SERVIVE2);
         IF FOR_ADD=4
         THEN BEGIN
              V(CT);
              TRANSIT(OUT);
              END
         ELSE BEGIN
              IF ((FOR_ADD=3) OR (FOR_ADD=5) OR (FOR_ADD=8))
              THEN TRANSIT(MONITOR(8))
              ELSE TRANSIT(MONITOR(7));
              END;
         END;
&*******************************************************************************************

/STATION/NAME=MONITOR(5);
 SERVICE=BEGIN
         CST( T_SERVIVE2);
         IF FOR_ADD=5
         THEN BEGIN
              V(CT);
              TRANSIT(OUT);
              END
         ELSE BEGIN
              IF ((FOR_ADD=6) OR (FOR_ADD=1) OR (FOR_ADD=4))
              THEN TRANSIT(MONITOR(1))
              ELSE TRANSIT(MONITOR(2));
              END;
         END;
&*******************************************************************************************

/STATION/NAME=MONITOR(6);
 SERVICE=BEGIN
         CST( T_SERVIVE2);
         IF FOR_ADD=6
         THEN BEGIN
              V(CT);
              TRANSIT(OUT);
              END
         ELSE BEGIN
              IF ((FOR_ADD=5) OR (FOR_ADD=2) OR (FOR_ADD=3))
              THEN TRANSIT(MONITOR(3))
              ELSE TRANSIT(MONITOR(4));
              END;
         END;
&*******************************************************************************************

/STATION/NAME=MONITOR(7);
 SERVICE=BEGIN
         CST( T_SERVIVE2);
         IF FOR_ADD=7
         THEN BEGIN
              V(CT);
              TRANSIT(OUT);
              END
         ELSE BEGIN
              IF ((FOR_ADD=2) OR (FOR_ADD=3) OR (FOR_ADD=8))
              THEN TRANSIT(MONITOR(2))
              ELSE TRANSIT(MONITOR(1));
              END;
         END;
&*******************************************************************************************

/STATION/NAME=MONITOR(8);
 SERVICE=BEGIN
```

```
       CST( T_SERVIVE2);
       IF FOR_ADD=8
       THEN BEGIN
            V(CT);
            TRANSIT(OUT);
            END
       ELSE BEGIN
            IF ((FOR_ADD=4) OR (FOR_ADD=1) OR (FOR_ADD=7))
            THEN TRANSIT(MONITOR(4))
            ELSE TRANSIT(MONITOR(3));
            END;
       END;
&*****************************************************************************************

/STATION/NAME=CT;
TYPE=RESOURCE,INFINITE;
&*****************************************************************************************

/CONTROL/TMAX=3000000.;
         ACCURACY=ALL QUEUE;
         ESTIM=SPECTRAL;

&**********************************************************************************************
&           THE FOLLOWING PROGROM IS FOR THE EXECUTION AND RESULTS
&**********************************************************************************************

/EXEC/
     BEGIN
     FOR K:=1 STEP 1 UNTIL 8
     DO BEGIN
        SA(K).IDEN:=K;
        SB(K).IDEN:=K;
        SC(K).IDEN:=K;
        SD(K).IDEN:=K;
        SE(K).IDEN:=K;
        SF(K).IDEN:=K;
        SG(K).IDEN:=K;
        SH(K).IDEN:=K;
        QA(K).IDEN:=K;
        QB(K).IDEN:=K;
        QC(K).IDEN:=K;
        QD(K).IDEN:=K;
        QE(K).IDEN:=K;
        QF(K).IDEN:=K;
        QG(K).IDEN:=K;
        QH(K).IDEN:=K;
        MONITOR(K).IDEN:=K;
        END;
  PRINT("****************************************************************************");
  PRINT("*      AR1_RATE     *    RHO1     *    RHO2     *  M_DEL     *  M_DEL'    *");
  PRINT("****************************************************************************");
      BEGIN                                     & CHANGE INTER  ARRIVE TIME;
        FOR J:=1 STEP 1 UNTIL 9
       DO
        BEGIN                                   & MEAN INTER  ARRIVAL TIME
         T_  ARRIVE:=500./(0.1*J);              & ARRIVAL RATE
          AR:=1./ T_  ARRIVE;
          AR1_RATE:=1.09/ T_  ARRIVE;           & ACTUAL   ARRIVAL RATE FOR THE MONITOR
          AR2_RATE:=21./ T_  ARRIVE;            &  SERVICE TIME FOR THE STATION
         T_SERVIVE1:=200.;                      & MEAN SERVICE TIME FOR THE MONITOR
         T_SERVIVE2:=25.;
        SIMUL;                                  & UTILIZATION OF A STATION
     RHO1:=  AR1_RATE* T_SERVIVE1;
     M_NB1:=(RHO1/(1-RHO1))-(RHO1*RHO1/(2*(1-RHO1)));
```

```
    M_NB1':=MCUSTNB(QA(3));
    M_DEL1:=M_NB1/  AR1_RATE;
    M_DEL1':=MRESPONSE(QA(3));
    RHO2:=  AR2_RATE* T_SERVIVE2;
    M_NB2:=(RHO2/(1-RHO2))-(RHO2*RHO2/(2*(1-RHO2)));
    M_NB2':=MCUSTNB(MONITOR(3));
    M_DEL2:=M_NB2/  AR2_RATE;
    M_DEL2':=MRESPONSE(MONITOR(3));         & MEAN TOTAL DELAY FROM ANALYSIS
    M_DEL:=M_DEL1+3*M_DEL2;                 & MEAN TOTAL FROM SIMULATION
    M_DEL':=MRESPONSE(CT);
PRINT("*",,"*",RHO1,"*",RHO2,"*",M_DEL,"*",M_DEL',"*");
        END;
     END;
PRINT("*****************************************************************");
END;

/END/
```

```
&***********************************************************************************************
&                   PROGRAM NAME: SHU.SHU2
&***********************************************************************************************

  &*************************************************************************************
  &    This program is to get the delay performance of Shu.shu2.                        *
  &    The physical configution of the upper leyer is a tree and the                    *
  &    lower leyer is a bus. The logic configuration of both upper                      *
  &    and lower leyer are SHUFFLENET. There are 8 nodes in the upper                   *
  &    leyer and 8 nodes in each bus.                                                   *
  &*************************************************************************************

/CONTROL/
       OPTION=NSOURCE,NRESULT;

/DECLARE/
INTEGER             I=8,K,J;

REAL                D1,D2,D3,D4,D5,D6,D7,D8,D_M8,
                    T_ARRIVAL,                    & PACKET ARRIVAL TIME IN A SOURCE
                    T_SERVICE1,                   & PACKET SERVIVE TIME IN A STATION
                    T_SERVICE2,                   & PACKET SERVIVE IN A ACTIVE NODE
                    AR_RATE,                      & ARRIVAL RATE TO A STATION
                    RHO1,                         & UTINIZATION IN A STATION
                    RHO2,                         & UTINIZATION IN A ACTIVE NODE
                    M_DEL';                       & MEAN TOTAL DELAY FROM SIMULATION


 CUSTOMER INTEGER                                 & THE DESTINATION BUS ADDRESS OF A PACKET
                    FOR_ADD,                      & THE DESTINATION STATION ADDRESS OF A PACKET
                    LOC_ADD,
                    CHA_ADD;


 QUEUE INTEGER                                    & IDENTIFICATION OF A QUEUE
                    IDEN;


 QUEUE
                    QA(I),QB(I),QC(I),QD(I),QE(I),QF(I),QG(I),QH(I),CT,
                    SA(I),SB(I),SC(I),SD(I),SE(I),SF(I),SG(I),SH(I),MONITOR(I);

  &**********************************************************************************************
  &  THIS PROGRAM IS TO REALIZE THE FIRST SOURCE STATION IN ONE BUS
  &**********************************************************************************************

  /STATION/NAME=SA(1 STEP 1 UNTIL 8);
  TYPE=SOURCE;
  SERVICE=BEGIN
          EXP(T_ARRIVAL);
          FOR_ADD:=RINT(1,8);
          LOC_ADD:=RINT(1,8);
          IF FOR_ADD=IDEN
          THEN BEGIN
               IF LOC_ADD=1
               THEN TRANSIT(OUT)
               ELSE BEGIN
                    P(CT);
                    CHA_ADD:=LOC_ADD;
                    TRANSIT(QA(IDEN));
                    END;
```

```
          END
     ELSE BEGIN
          P(CT);
          CHA_ADD:=8;
          TRANSIT(QA(IDEN));
          END;
     END;

&********************************************************************
&  THIS PROGRAM IS TO REALIZE THE FIRST QUEUE STATION IN ONE BUS
&********************************************************************

/STATION/NAME=QA(1 STEP 1 UNTIL 8);
SERVICE=BEGIN
        CST(T_SERVICE1);
        IF ((CHA_ADD=5) OR (CHA_ADD=6))
        THEN BEGIN
             V(CT);
             TRANSIT(OUT);
             END
        ELSE BEGIN
             IF ((CHA_ADD=2) OR (CHA_ADD=8))
             THEN TRANSIT(QE(IDEN))
             ELSE TRANSIT(QF(IDEN));
             END;
        END;

&*********************************************************************************************
&  THIS PROGRAM IS TO REALIZE THE SECOND SOURCE STATION IN ONE BUS
&*********************************************************************************************

/STATION/NAME=SB(1 STEP 1 UNTIL 8);
TYPE=SOURCE;
SERVICE=BEGIN
        EXP(T_ARRIVAL);
        FOR_ADD:=RINT(1,8);
        LOC_ADD:=RINT(1,8);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF LOC_ADD=2
             THEN TRANSIT(OUT)
             ELSE BEGIN
                  P(CT);
                  CHA_ADD:=LOC_ADD;
                  TRANSIT(QB(IDEN));
                  END;
             END
        ELSE BEGIN
             P(CT);
             CHA_ADD:=8;
             TRANSIT(QB(IDEN));
             END;
        END;

&********************************************************************
&  THIS PROGRAM IS TO REALIZE THE FIRST QUEUE STATION IN ONE BUS
&********************************************************************

/STATION/NAME=QB(1 STEP 1 UNTIL 8);
SERVICE=BEGIN
        CST(T_SERVICE1);
        IF CHA_ADD=8
        THEN BEGIN
```

```
          IF FOR_ADD=IDEN
          THEN BEGIN
                V(CT);
                TRANSIT(OUT);
                END
          ELSE BEGIN
                CHA_ADD:=LOC_ADD;
                TRANSIT(MONITOR(IDEN));
                END;
          END
      ELSE BEGIN
          IF CHA_ADD=7
          THEN BEGIN
                V(CT);
                TRANSIT(OUT);
                END;
          IF ((CHA_ADD=1) OR (CHA_ADD=6))
          THEN TRANSIT(QG(IDEN))
          ELSE TRANSIT(QH(IDEN));
          END;
      END;
&**************************************************************************************************
/STATION/NAME=SC(1 STEP 1 UNTIL 8);
TYPE=SOURCE;
SERVICE=BEGIN
        EXP(T_ARRIVAL);
        FOR_ADD:=RINT(1,8);
        LOC_ADD:=RINT(1,8);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF LOC_ADD=3
             THEN TRANSIT(OUT)
             ELSE BEGIN
                  P(CT);
                  CHA_ADD:=LOC_ADD;
                  TRANSIT(QC(IDEN));
                  END;
             END
         ELSE BEGIN
             P(CT);
             CHA_ADD:=8;
             TRANSIT(QC(IDEN));
             END;
         END;

 /STATION/NAME=QC(1 STEP 1 UNTIL 8);
 SERVICE=BEGIN
         CST(T_SERVICE1);
         IF ((CHA_ADD=5) OR (CHA_ADD=6))
         THEN BEGIN
              V(CT);
              TRANSIT(OUT);
              END
         ELSE BEGIN
              IF ((CHA_ADD=4) OR (CHA_ADD=7))
              THEN TRANSIT(QF(IDEN))
              ELSE TRANSIT(QE(IDEN));
              END;
         END;

  &**************************************************************************************************

  /STATION/NAME=SD(1 STEP 1 UNTIL 8);
```

```
TYPE=SOURCE;
SERVICE=BEGIN
        EXP(T_ARRIVAL);
        FOR_ADD:=RINT(1,8);
        LOC_ADD:=RINT(1,8);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF LOC_ADD=4
             THEN TRANSIT(OUT)
             ELSE BEGIN
                  P(CT);
                  CHA_ADD:=LOC_ADD;
                  TRANSIT(QD(IDEN));
                  END;
             END
        ELSE BEGIN
             P(CT);
             CHA_ADD:=8;
             TRANSIT(QD(IDEN));
             END;
        END;

/STATION/NAME=QD(1 STEP 1 UNTIL 8);
SERVICE=BEGIN
        CST(T_SERVICE1);
        IF CHA_ADD=8
        THEN BEGIN
             IF FOR_ADD=IDEN
             THEN BEGIN
                  V(CT);
                  TRANSIT(OUT);
                  END
             ELSE BEGIN
                  CHA_ADD:=LOC_ADD;
                  TRANSIT(MONITOR(IDEN));
                   END;
           END
         ELSE BEGIN
              IF CHA_ADD=7
              THEN BEGIN
                   V(CT);
                   TRANSIT(OUT);
                   END;
              IF ((CHA_ADD=3) OR(CHA_ADD=5))
              THEN TRANSIT(QH(IDEN))
              ELSE TRANSIT(QG(IDEN));
              END;
          END;
&************************************************************************************************

/STATION/NAME=SE(1 STEP 1 UNTIL 8);
TYPE=SOURCE;
SERVICE=BEGIN
        EXP(T_ARRIVAL);
        FOR_ADD:=RINT(1,8);
        LOC_ADD:=RINT(1,8);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF LOC_ADD=5
             THEN TRANSIT(OUT)
             ELSE BEGIN
                  P(CT);
                  CHA_ADD:=LOC_ADD;
                  TRANSIT(QE(IDEN));
```

```
                  END;
             END
        ELSE BEGIN
             P(CT);
             CHA_ADD:=8;
             TRANSIT(QE(IDEN));
             END;
        END;


/STATION/NAME=QE(1 STEP 1 UNTIL 8);
SERVICE=BEGIN
        CST(T_SERVICE1);
        IF ((CHA_ADD=1) OR (CHA_ADD=2))
        THEN BEGIN
             V(CT);
             TRANSIT(OUT);
             END
        ELSE BEGIN
             IF ((CHA_ADD=4) OR (CHA_ADD=6))
             THEN TRANSIT(QA(IDEN))
             ELSE TRANSIT(QB(IDEN));
             END;
        END;
&********************************************************************************************
/STATION/NAME=SF(1 STEP 1 UNTIL 8);
TYPE=SOURCE;
SERVICE=BEGIN
        EXP(T_ARRIVAL);
        FOR_ADD:=RINT(1,8);
        LOC_ADD:=RINT(1,8);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF LOC_ADD=6
             THEN TRANSIT(OUT)
             ELSE BEGIN
                  P(CT);
                  CHA_ADD:=LOC_ADD;
                  TRANSIT(QF(IDEN));
                  END;
             END
        ELSE BEGIN
             P(CT);
             CHA_ADD:=8;
             TRANSIT(QF(IDEN));
             END;
        END;


/STATION/NAME=QF(1 STEP 1 UNTIL 8);
SERVICE=BEGIN
        CST(T_SERVICE1);
        IF ((CHA_ADD=3) OR (CHA_ADD=4))
        THEN BEGIN
             V(CT);
             TRANSIT(OUT);
             END
        ELSE BEGIN
             IF ((CHA_ADD=2) OR (CHA_ADD=5))
             THEN TRANSIT(QC(IDEN))
             ELSE TRANSIT(QD(IDEN));
             END;
        END;
&********************************************************************************************
```

```
/STATION/NAME=SG(1 STEP 1 UNTIL 8);
TYPE=SOURCE;
SERVICE=BEGIN
        EXP(T_ARRIVAL);
        FOR_ADD:=RINT(1,8);
        LOC_ADD:=RINT(1,8);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF LOC_ADD=7
             THEN TRANSIT(OUT)
             ELSE BEGIN
                  P(CT);
                  CHA_ADD:=LOC_ADD;
                  TRANSIT(QG(IDEN));
                  END;
             END
        ELSE BEGIN
             P(CT);
             CHA_ADD:=8;
             TRANSIT(QG(IDEN));
             END;
        END;

/STATION/NAME=QG(1 STEP 1 UNTIL 8);
SERVICE=BEGIN
        CST(T_SERVICE1);
        IF ((CHA_ADD=1) OR (CHA_ADD=2))
        THEN BEGIN
             V(CT);
             TRANSIT(OUT);
             END
        ELSE BEGIN
             IF ((CHA_ADD=3) OR (CHA_ADD=8))
             THEN TRANSIT(QB(IDEN))
             ELSE TRANSIT(QA(IDEN));
             END;
        END;

&********************************************************************************************
/STATION/NAME=SH(1 STEP 1 UNTIL 8);
TYPE=SOURCE;
SERVICE=BEGIN
        EXP(T_ARRIVAL);
        FOR_ADD:=RINT(1,8);
        LOC_ADD:=RINT(1,8);
        IF FOR_ADD=IDEN
        THEN BEGIN
             IF LOC_ADD=8
             THEN TRANSIT(OUT)
             ELSE BEGIN
                  P(CT);
                  CHA_ADD:=LOC_ADD;
                  TRANSIT(QH(IDEN));
                  END;
             END
        ELSE BEGIN
             P(CT);
             CHA_ADD:=LOC_ADD;
             TRANSIT(MONITOR(IDEN));
             END;
        END;

/STATION/NAME=QH(1 STEP 1 UNTIL 8);
SERVICE=BEGIN
```

```
        CST(T_SERVICE1);
        IF ((CHA_ADD=3) OR (CHA_ADD=4))
        THEN BEGIN
             V(CT);
             TRANSIT(OUT);
             END
        ELSE BEGIN
             IF ((CHA_ADD=1) OR (CHA_ADD=7))
             THEN TRANSIT(QD(IDEN))
             ELSE TRANSIT(QC(IDEN));
             END;
        END;

&*********************************************************************************************
   &*********************************************************************
   &  The monitors are the intermediate nodes for interconnecting buses.*
   &  There is a monitor on the top of each bus. When the packet is     *
   &  going to outside of the bus, it is going through the monitors to  *
   &  reach the destination. The logic connection of the monitors is    *
   &  the 8 node shufflenet.                                             *
   &*********************************************************************


/STATION/NAME=MONITOR(1);
 SERVICE=BEGIN
        CST(T_SERVICE2);
        IF FOR_ADD=1
        THEN IF CHA_ADD=8
             THEN BEGIN
                  V(CT);
                  TRANSIT(OUT);
                  END
             ELSE TRANSIT(QH(1))
        ELSE BEGIN
             IF ((FOR_ADD=2) OR (FOR_ADD=5) OR (FOR_ADD=8))
             THEN TRANSIT(MONITOR(5))
             ELSE TRANSIT(MONITOR(6));
             END;
        END;

&*********************************************************************************************

/STATION/NAME=MONITOR(2);
 SERVICE=BEGIN
        CST(T_SERVICE2);
        IF FOR_ADD=2
        THEN IF CHA_ADD=8
             THEN BEGIN
                  V(CT);
                  TRANSIT(OUT);
                  END
            ELSE BEGIN
                  CHA_ADD:=LOC_ADD;
                  TRANSIT(QH(2));
                  END
        ELSE BEGIN
             IF ((FOR_ADD=1) OR (FOR_ADD=6) OR (FOR_ADD=7))
             THEN TRANSIT(MONITOR(7))
             ELSE TRANSIT(MONITOR(8));
             END;
        END;

&*********************************************************************************************

/STATION/NAME=MONITOR(3);
```

```
 SERVICE=BEGIN
        CST(T_SERVICE2);
        IF FOR_ADD=3
        THEN IF CHA_ADD=8
             THEN BEGIN
                  V(CT);
                  TRANSIT(OUT);
                  END
             ELSE TRANSIT(QH(3))
        ELSE BEGIN
             IF ((FOR_ADD=4) OR (FOR_ADD=6) OR (FOR_ADD=7))
             THEN TRANSIT(MONITOR(6))
             ELSE TRANSIT(MONITOR(5));
             END;
        END;
&*********************************************************************************************

/STATION/NAME=MONITOR(4);
 SERVICE=BEGIN
        CST(T_SERVICE2);
        IF FOR_ADD=4
        THEN IF CHA_ADD=8
             THEN BEGIN
                  V(CT);
                  TRANSIT(OUT);
                  END
             ELSE TRANSIT(QH(4))
        ELSE BEGIN
             IF ((FOR_ADD=3) OR (FOR_ADD=5) OR (FOR_ADD=8))
             THEN TRANSIT(MONITOR(8))
             ELSE TRANSIT(MONITOR(7));
             END;
        END;
&*********************************************************************************************

/STATION/NAME=MONITOR(5);
 SERVICE=BEGIN
        CST(T_SERVICE2);
        IF FOR_ADD=5
        THEN IF CHA_ADD=8
             THEN BEGIN
                  V(CT);
                  TRANSIT(OUT);
                  END
             ELSE TRANSIT(QH(5))
        ELSE BEGIN
             IF ((FOR_ADD=1) OR (FOR_ADD=4) OR (FOR_ADD=6))
             THEN TRANSIT(MONITOR(1))
             ELSE TRANSIT(MONITOR(2));
             END;
        END;
&*********************************************************************************************

/STATION/NAME=MONITOR(6);
 SERVICE=BEGIN
        CST(T_SERVICE2);
        IF FOR_ADD=6
        THEN IF CHA_ADD=8
             THEN BEGIN
                  V(CT);
                  TRANSIT(OUT);
                  END
             ELSE TRANSIT(QH(6))
```

```
          ELSE BEGIN
               IF ((FOR_ADD=2) OR (FOR_ADD=5) OR (FOR_ADD=3))
               THEN TRANSIT(MONITOR(3))
               ELSE TRANSIT(MONITOR(4));
               END;
          END;

&*******************************************************************************************

/STATION/NAME=MONITOR(7);
 SERVICE=BEGIN
          CST(T_SERVICE2);
          IF FOR_ADD=7
          THEN IF CHA_ADD=8
               THEN BEGIN
                    V(CT);
                    TRANSIT(OUT);
                    END
               ELSE TRANSIT(QH(7))
          ELSE BEGIN
               IF ((FOR_ADD=2) OR (FOR_ADD=3) OR (FOR_ADD=8))
               THEN TRANSIT(MONITOR(2))
               ELSE TRANSIT(MONITOR(1));
               END;
          END;

&*******************************************************************************************

/STATION/NAME=MONITOR(8);
 SERVICE=BEGIN
          CST(T_SERVICE2);
          IF FOR_ADD=8
          THEN IF CHA_ADD=8
               THEN BEGIN
                    V(CT);
                    TRANSIT(OUT);
                    END
               ELSE TRANSIT(QH(8))
          ELSE BEGIN
               IF ((FOR_ADD=1) OR (FOR_ADD=4) OR (FOR_ADD=7))
               THEN TRANSIT(MONITOR(4))
               ELSE TRANSIT(MONITOR(3));
               END;
          END;

&*******************************************************************************************

/STATION/NAME=CT;
TYPE=RESOURCE,INFINITE;

&*******************************************************************************************
&              THE FOLLOWING PROGROM IS FOR THE EXECUTION AND RESULTS
&*******************************************************************************************

/CONTROL/TMAX=8000000;

/EXEC/
     BEGIN
     FOR K:=1 STEP 1 UNTIL 8
     DO BEGIN
        SA(K).IDEN:=K;
        SB(K).IDEN:=K;
        SC(K).IDEN:=K;
        SD(K).IDEN:=K;
        SE(K).IDEN:=K;
        SF(K).IDEN:=K;
```

```
        SG(K).IDEN:=K;
        SH(K).IDEN:=K;
        QA(K).IDEN:=K;
        QB(K).IDEN:=K;
        QC(K).IDEN:=K;
        QD(K).IDEN:=K;
        QE(K).IDEN:=K;
        QF(K).IDEN:=K;
        QG(K).IDEN:=K;
        QH(K).IDEN:=K;
        MONITOR(K).IDEN:=K;
        END;
PRINT("*****************************************************");
PRINT("*    AR      *   RHO1     *   RHO2     *  M_DEL'    *");
PRINT("*****************************************************");
     FOR J:=1 STEP 1 UNTIL 9       & CHANGE INTERARRIVE TIME;
     DO BEGIN
        T_ARRIVAL:=200/(0.1*J);             & MEAN INTERARRIVAL TIME
        AR:=1./T_ARRIVAL;
        T_SERVICE1:=40;                     & MEAN SERVICE TIME
        T_SERVICE2:=5;
        SIMUL;
        RHO1:=MTHRUPUT(QH(4));              & UTILIZATION OF A STATION
        RHO2:=MTHRUPUT(MONITOR(4));         & UTILIZATIN OF A ACTIVE NODE
        M_DEL':=MRESPONSE(CT);              & MEAN TOTAL DELAY
        PRINT("*",AR,"*",RHO1,"*",RHO2,"*",M_DEL',"*");
        END;
PRINT("*****************************************************");
     END;

/END/
```