WAVELENGTH AND TIME DIVISION MULTIPLEXING WITH LIGHTPATH TRESPASSING FOR ALL-OPTICAL STAR LOCAL AREA NETWORKS

A THESIS
SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE
OF
MASTER OF SCIENCE IN COMPUTER SCIENCE
IN THE
UNIVERSITY OF CANTERBURY
by
Justin Ray Macfarlane

University of Canterbury
1998

trěs′pass *v.i.* **1.** Make unlawful or unwarrantable intrusion.

- *Concise Oxford Dictionary*

Abstract

Many medium access control protocols have been proposed for optical wavelength division multiplexing local area networks with a star topology. These protocols range from those based on the concept of fixed-assignment of communication subchannels, such as TDMA (Time Division Multiple Access); reservation of communication subchannels, such as DAS (Dynamic Allocation Scheme); or random-access to communication subchannels, such as DT-WDMA (Dynamic Time-Wavelength Division Multiple Access). In addition various hybrid protocols have been considered, for example, protocols incorporating both fixed-assignment and reservation rules, such as HTDM (Hybrid TDM).

This thesis is on a novel hybrid protocol of fixed-assignment and random-access called "WTDMA with lightpath trespassing". This protocol combines the most desirable aspects of fixed-assignment and random-access protocols, while limiting their drawbacks. The performance of different versions of the protocol are analysed both mathematically and by stochastic simulation. The obtained results justify the introduction of the WTDMA with trespassing protocol, and indicate the situations where its use is advantageous.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Wavelength Division Multiplexing (WDM) networks are all optical networks. These third generation networks use the large bandwidth capacity of optical fibers to enable high speed data transmission of Gigabits per second. Such high transmission rates are necessary to enable conventional and new data services which require signals of large bandwidth (e.g. video, multimedia). First generation networks rely on electronic media. Second generation networks use optical fibre for transmission, but rely on electronic technology for switching, which severely limits their effective throughput.

WDM networks have been extensively studied over the past ten years. Optical fibre has become cheaper than co-axial cable, making optical networks even more attractive. At first WDM was used to upgrade the capacity of point-to-point systems, by adding independent optical channels, well separated in wavelength (Brackett, 1990). This proved inefficient, as a new transmitter-receiver pair was needed for each extra channel. The cost of each additional communicating pair was greater than the gain from speeding up the network. It was realised that WDM had networking possibilities, beyond adding more point-to-point links.

Three basic issues are distinct to WDM networks. Firstly, there is an issue of their physical topology. Secondly, the question of whether single or multi hopping is implemented on this topology. Lastly a decision must be made on what communications strategy is used to implement single or multi hopping.

Star, tree and bus topologies have all been proposed as the physical backbone for WDM networks. The simplicity of the star coupler [Figure 1.1], which combines all the transmitted wavelengths and then broadcasts them to all possible destinations, has probably been the main reason that the star physical

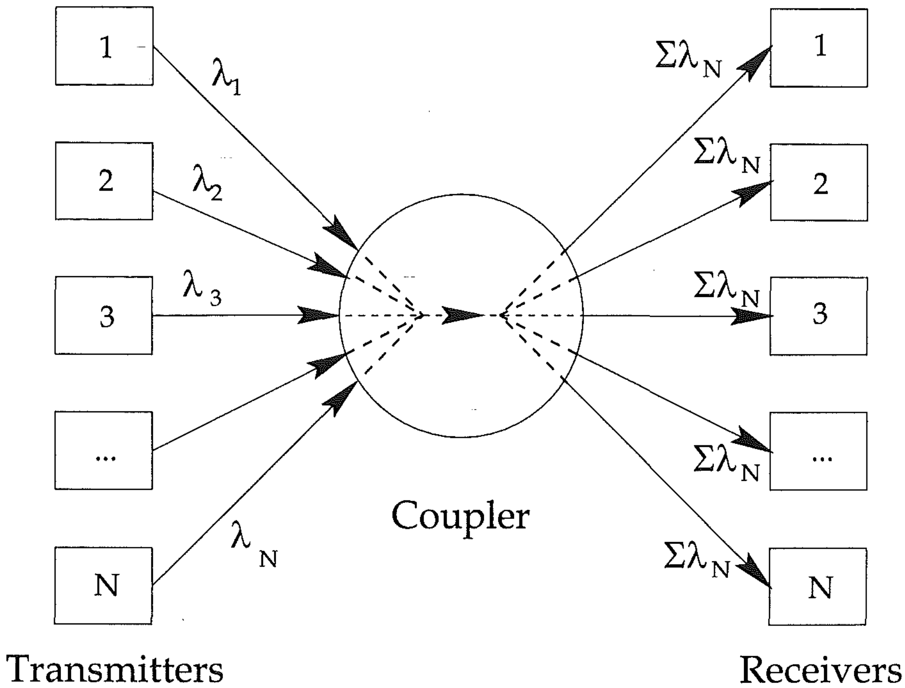

Figure 1.1: A WDM star network with a passive coupler and N nodes. λ_i = wavelength used by station i.

topology is prevalent. There are two basic kinds of couplers, active and passive. An active coupler requires power to run, and can have some form of intelligence and usually "stores and forwards" packets. A passive coupler simply combines wavelengths, then broadcasts them, without the need for additional power.

In single hop networks packets are sent directly from the source to the destination. This means a direct point–to–point link needs to be established between each communicating pair of nodes. Thus single hop networks need no routing or flow control of data streams on route to their destinations.

In multihop network packets can be passed through intermediate nodes before they reach their destinations. Multihop networks need no expensive tunable transceivers (transmitters or receivers). They trade the overhead control in setting up and maintaining direct links for the delay connected with routing a packet through more than one node.

The communications strategy is defined by the existence of control functions assigned to none, one or more of the wavelengths (or control channels) and on the number of, and type of transmitters and receivers used at each node.

The control channel allows "pre-transmission co-ordination", which can allow better utilisation of the data channels, and lowers or totally avoids collisions in channels. This is of course off-set by an overhead, which can increase packet delays, or restrict the maximum number of nodes in a network.

Most of a network's characteristics can be attributed to its topology. The network is made up of the physical and logical topology. Physical topology is defined as the way nodes are interconnected physically. The logical topology is how each node is logically connected within the physical topology.

The majority of WDM networks proposed are based on a star with a passive coupler as the physical topology, although more fully connected graphs are also considered. Logical star topologies are used in the majority of single hop networks, while multihop networks use mesh and other highly interconnected topologies.

The star topology is the simplest structure offering full broadcast capability for all the nodes. A passive star coupler has the advantage of not consuming power, which increases its operational reliability (see Section 2.4 for information on couplers). Star topology is also very flexible as the physical base, allowing many forms of logical topology to be implemented on top of it.

While star networks work well as LANs, their typical protocols are not suitable for MANs and WANs, and in such cases other architectures and protocols are more appropriate.

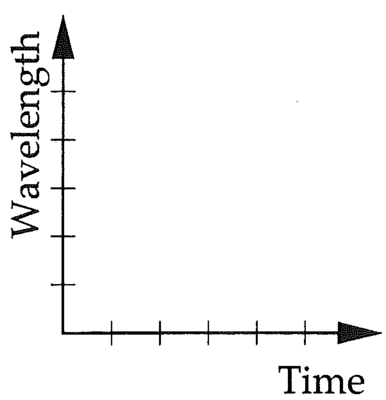

Figure 1.2: Wavelength and Time Division Multiple Access (WTDMA): both wavelengths and time are divided and shared among users of the transmission medium.

TDMA (Time Division Multiple Access) is a simple way of scheduling sub-channels (time slots) in a network: different stations can use different (pre-assigned) time slots for transmitting packets. Its generalisation in optical networks is known as synchronous WTDMA (Wavelength and Time Division Multiple Access [Figure 1.2])[1]: during one time slot, multiple packets can be transmitted, each at a different wavelength. It is an especially good technique at high traffic level where its simplicity is a major asset. However it is wasteful at low traffic level, with many time slots being unused. This is because WTDMA offers no flexibility to cope with changing traffic patterns. It can be used with any tunable transceiver setup ($TT-FR$, $FT-TR$ and $TT-TR$, see Section 2.1), and no control-channel is needed.

Recently proposed protocols for optical networks have focused on finding more flexible alternatives to WTDMA. Borella and Mukerjee (1995) use scheduling theory to assign wavelengths in a more efficient way, and take account of the latency of the transmitter or receiver to minimise delays. These authors are currently looking at ways of implementing dynamically altered scheduling, so changing traffic-flows can be accommodated.

Reservation-based protocols have also been proposed. Their major failings are that they need a control-channel; thus they are relatively complex, and an initial round-trip delay is needed in deciding rights for transmission.

Hybrid WTDMA schemes (Chipalkatti *et al.*, 1992; Chipalkatti *et al.*, 1993) have also been proposed for combining the concept of WTDMA with a reservation-based protocol. The gains in network throughput can then be balanced against higher complexity and larger round-trip delay of reservation schemes.

The p_i-persistent protocol (Mukherjee & Meditch, 1987) uses a probabilistic rule for accessing free-slots. This has been proved to work well for bus and ring networks and provides a fair, collision-free access to a network. This protocol cannot however be applied directly to optical star networks, as they work on a broadcast–and–select basis. Hence there is no way to view the access medium for free slots. In this thesis, it is shown, nevertheless, that a probabilistic rule can be applied in star networks, to decide whether to transmit data during a given time slot, at a given wavelength.

What is proposed here is a hybrid-protocol that uses a simple probabilistic approach (an original technique called "Trespassing", explained in Chapter 3).

[1] In this thesis, synchronous TDMA and synchronous WTDMA will simply be referred to as TDMA and WTDMA.

This eliminates the need for a control-channel and complex reservation techniques. Because random-access is used, the protocol can be applied when a tunable transmitter and either a tunable receiver ($TT - TR$) or a fixed receiver ($TT - FR$) is used at each station. Only $TT - FR$ is considered here, as the hardware costs of having multiple tunable transceivers could be prohibitive.

Trespassing should lower the packet delay at low traffic loads, as its random-access properties prevent packets from waiting a full TDMA cycle before being transmitted. It should also prevent the limitations on maximum throughput at high traffic level of random-access protocols, by using fixed-assignment (TDMA), that prevents multiple collisions of packets, which causes network instability.

Mathematical and simulation modelling is used to study the throughput and mean delay characteristics of synchronous WTDMA, and its extension with trespassing. These models are used in performance evaluation studies of WTDMA with and without trespassing.

1.1 Thesis Layout

This thesis deals with the classification and implementation issues of single hop and multihop networks, and the current state of technology used in optical star networks in Chapter 2. Chapter 3 describes WTDMA networks and the newly proposed concept of trespassing. Since trespassing can cause collisions of packets transmitted to the same destination, a simple collision avoidance mechanism, implementable at the coupler, is also discussed. Chapter 4 outlines the basic mathematical model used for performance evaluation of WTDMA networks, while Chapter 5 presents the generic simulation model used for the performance evaluation. Performance evaluation of various versions of the WTDMA networks is done in Chapter 6. Chapter 7 summarises results and discusses directions of possible further research.

Chapter 2

Survey of WDM networks

This chapter surveys Medium Access Control (MAC) protocols and the communications strategies for use in single hop WDM networks, implemented using a physical and logical star topology, with a passive coupler.

Single hop networks and their classification and implementation is explained in Section 2.1. Single hop networks which have different, logical topologies are also briefly discussed. The principle of multihop networks is explained in Section 2.2. The issues behind designing a WDM network protocol are outlined in Section 2.3, and the technology behind optical networks is considered in Section 2.4. The concept of hybrid protocols is covered in Section 2.5.

2.1 Single hop networks

Single hop networks have a great range of designs (Mukherjee, 1992a). This stems from the flexibility in choosing different ways of interface with a star coupler, and has led to a large number (24 are discussed here) of MAC protocols suggested for single hop WDM star networks [Table 2.1].

2.1.1 Classification

As mentioned, single hop networks are generally implemented as stars with passive couplers. They can be classified depending on how they interface with the coupler, whether the transmitters are fixed or tunable, and whether the receivers are fixed or tunable (Ramaswami, 1990). Some star networks use special control channels, which co-ordinate data transmissions.

Thus, star networks with passive couplers can be described using the following general notation (Mukherjee, 1992a):

$\mathbf{CC^c - FT^iTT^j - FR^mTR^n}$,

Here CC^c represents c control channels ($c \geq 0$), FT^i the fixed transmitters of quantity i, TT^j the tunable transmitters of quantity j, FR^m the fixed receivers of quantity m and TR^n, the tunable receivers of quantity n, where i, j, m and n can take values from 0 to N; but of course $i+j \geq 0$ and $m+n \geq 0$. Null sets are removed from this notation, e.g. $CC^0 - FT^0TT^1 - FR^2TR^0 \Rightarrow TT - FR^2$.

To simplify further discussion, we will reduce this classification by distinguishing between fixed and tunable transmitters, and, fixed and tunable receivers only. In this case, we can distinguish four basic configurations of WDM star networks. The following summarises each of them, showing their pros and cons.

- $\mathbf{CC^c - FT - FR}$ ($c \geq 0$): Characterised by a lack of tuning delays, as all nodes can communicate with each other, via broadcast packets. There is a limitation in the maximum number of channels possible, since each node requires its own unique wavelength. The need for N receivers at every node makes this network relatively expensive, especially as the number of nodes increases.

- $\mathbf{CC^c - TT - FR}$ ($c \geq 0$): A fixed address (wavelength) for each node is used, and the transmitter must tune to that wavelength to send a packet. This can lead to collision in channels, if no method of control is used. This configuration of network is restricted in size by the number of wavelength channels available, unless wavelengths are shared.

- $\mathbf{CC^c - FT - TR}$ ($c \geq 0$): The receiver must be notified what channel to tune to, either by scanning the channels, or through a control-channel. This can lead to high costs in setting up communications, given the high value of the parameter a (relative propagation delay) in optical data transmission. Because a large number of packets can be transmitted during the round trip delay, the signalling needed to setup a transmission leads to a large increase in packet delay. Destination conflict (also known as receiver collision), occurs when a packet is not received because the receiver is tuned into another wavelength, is also a problem that needs to be resolved. This form of network is again restricted in size by the maximum number of channels available, unless wavelengths are shared.

- $\mathbf{CC^c - TT - TR}$ ($c \geq 0$): This is the most flexible configuration, as the number of nodes possible is not limited by the number of wavelengths available. However it suffers the most from the limits in transceiver technology instead. Tunable transmitters and receivers are expensive, and not satisfactorily fast at this stage. Using both tunable transmitters and receivers heightens this problem. The delays in tuning both transceivers must be taken into account. If the protocol does not use fixed assignment of subchannels, then there is the extra problem of having both transceivers tuned to the same wavelength. This leads to higher signalling costs and collisions both at the receiver and in the channels.

Proposed Medium Access Control Protocols

Class	Protocol/Network	Reference
$FT - FR^{\dagger}$	LAMBDANET‡	Chlamtac & Ganz (1988a)
		Goodman (1990)
$TT - FR^{\dagger}$	FOX‡	Arthurs (1986)
	PAC	Karol & Glance (1991; 1994)
	Borella '95	Borella & Mukerjee (1995)
	MaTPi	Tridandapani *et al.* (1994)
	HRP/TSA	Sivalingam & Wang (1996)
	I-TDMA*	Bogineni *et al.* (1993)
		Bogineni & Dowd (1993)
	AMTRAC	Chlamtac & Ganz (1988b)
$FT - TR^{\dagger}$	Rainbow‡	Dono (1990), Green (1992)
		Janneillo *et al.* (1992)
	DAS	Chipalkatti *et al.* (1992; 1993)
	HTDM	Chipalkatti *et al.* (1992; 1993)
	GTDM	Kannan *et al.* (1994)
	Quadro	Chlamtac & Fumagalli (1994)
	DT-WDMA	Chen *et al.* (1990)
	CF-WDMA	Chen & Yum (1991)
	Humblet '93	Humblet *et al.* (1993)
	Starnet	Poggiolini & Kazovsky (1991)
		Kazovsky & Poggiolini (1993)
$TT - TR^{\dagger}$	TDM	Chlamtac & Ganz (1988a)
	HYPASS‡	Arthurs *et al.* (1988)
	RCA	Jia & Mukherjee (1991)
	Lookahead–reservation	Wong & Yum (1988)
	POPSMAC	Hou *et al.* (1996)
	MARKAB	Semaan (1993)
	PROTON	Levine & Akyildi (1995)

Table 2.1: WDM single hop MAC protocols, for star networks with a passive-star coupler. † – Presence or absence of control-channels is ignored. ‡ – Implemented.

2.1.2 Implementation

The WDM star networks proposed (including those implemented) are listed in Table 2.1. In this section, each of the protocols from the table is briefly discussed in more detail.

2.1.2.1 Fixed Transmitters and Fixed Receivers

With single hop networks, the full connectivity of the $FT^i - FR^m$ configuration, requires either a bank of N transmitters, or N receivers, or both, at each node (N receivers is most common). This gives every node a point to point link with every other node. This is attractive, because there are no losses of time associated with tuning delays, but it requires a large number of transceivers (at least N^2 for full connectivity), e.g. in a ten node LAMBDANET network, 10 transmitters and 100 receivers are used.

This idea is applied for example in the single hop LAMBDANET (Chlamtac & Ganz, 1988a; Goodman, 1990), which is a Bellcore demonstration network. It uses N receivers at each node and a fixed wavelength transmitter, that broadcasts onto the passive optical star coupler (i.e. $FT - FR^N$).

2.1.2.2 Tunable Transmitters and Fixed Receivers

Having a fixed receiver means that the destination has a fixed address that packets can be sent to. Because each node needs to have a unique wavelength, the size of the network is limited in maximum size to the number of wavelengths available. Additionally, without appropriate control strategies, collisions in channels can occur when two nodes both transmit on one frequency.

Examples of such networks are listed below.

- **FOX** (Fiber-Optic Crossconnect)
  – (Arthurs, 1986)

  FOX was used to investigate the potential of fast tunable lasers in a parallel environment. It was implemented as two stars, one to simulate signals travelling from the processors to memory, the other for signals going in the opposite direction. Collisions were resolved using a binary exponential backoff algorithm.

- **PAC** (Protection Against Collision)
  – (Karol & Glance, 1991; Karol & Glance, 1994)

Each node is connected to the central passive star coupler by a PAC switch, which senses the channels, looking for a carrier. If none is found, then the sensed channel is available. A transmitter can only access an available channel. If two transmitters both try to access an available channel, then both are denied access to the transmission channel.

- **Borella '95**
  – (Borella & Mukherjee, 1995)

  This protocol uses a load balancing algorithm on a traffic matrix, to efficiently allocate slots to each node. It is a fixed assignment protocol, like TDM, but for non-uniform traffic. It can also cope with limited available wavelengths. It is designed to support tens to hundreds of nodes, with only in the order of ten wavelengths. The load balancing algorithm also accounts for transceiver tuning latency. Although the proposed protocol is $TT - FR$, it is also possible to adapt it to $FT - TR$.

- **MaTPi** (Masking Tuning times through Pipelining)
  – (Tridandapani *et al.*, 1994)

  This protocol tries to take into account the tuning time of a transmitter. It assumes that the tuning time of the transmitter is in the same order of magnitude as the transmission time. The tuning time is masked by overlapping it with the transmitting time of other lasers. The goal is to have a protocol which achieves high-throughput with cheap off-the-shelf components, so multiple transceivers and control channels are used.

- **HRP/TSA** (Hybrid Reservation Pre-Allocation/Time Slot Assignment)
  – (Sivalingam & Wang, 1996)

  This protocol is an improvement on the original HRP protocol. It is a reservation based protocol, where channels are reserved in the reservation cycle, then used in the data cycle. The original protocol only allowed one channel to be reserved per node per cycle. The improved protocol allows more than one. This improves utilisation, and reduces wasted slots, especially for non-uniform traffic.

- **I-TDMA*** (Interleaved – TDMA)
  – (Bogineni *et al.*, 1993; Bogineni & Dowd, 1993)

  I-TDMA* is an extension of the previously proposed I-TDMA protocol(which was $TT - TR$) (Sivalingam *et al.*, 1992). This protocol uses a "home channel", eliminating the need for both transceivers to be tunable. This reduces the complexity of system. I-TDMA* is based on TDM and is suitable for networks where there are more nodes than available channels. The analysis done assumes the $TT - FR$ format, although it could also be implemented as $FT - TR$.

- **AMTRAC**

  – (Chlamtac & Ganz, 1988b)

  This network uses a folded-bus logical topology on a passive star coupler. Each node is assigned a wavelength, which it can share with other nodes (thus not limiting the number of nodes to the number of wavelengths). To transmit, a node selects the known channel of the destination node, and transmits when it gets access to a virtual token. This is a multi-channel and train oriented protocol, which uses cycles and mini-slots.

2.1.2.3 Fixed Transmitters and Tunable Receivers

When there is no control channel, each receiver constantly scans the channels looking for a node that wants to transmit to it. While the node transmits its request to send a message, it also listens for a response. When there is a response, it transmits its message. This guarantees that no collisions will occur. Another method of implementing $FT - TR$ is with a control channel. The control channel tells a node what wavelength to tune its receiver to, to receive a packet.

This method is open to destination conflict, because when two or more packets are sent to a node, its receiver can only tune in to one of them.

Examples of such networks are listed below:

- **Rainbow**

  – (Dono, 1990; Janneillo *et al.*, 1992; Green, 1992)

  Rainbow is a IBM project to construct a WDM MAN network. The receiver scans channels looking for a node that wanted to transmit to it. The transmitter continuously sends out setup requests to a node, and tunes its receiver to the wavelength of that setup request. When the receiver detects a setup request, it sends an acknowledgement to the transmitting node's receiver. The node can then transmit its packet. The effectiveness of the protocol is highly dependent on the number of nodes and the tuning speed of the receiver.

- **DAS** (Dynamic Allocation Scheme)
  – (Chipalkatti *et al.*, 1992; Chipalkatti *et al.*, 1993)

  DAS uses a control channel, thus it is a $CC - FT - TR$ network. The protocol dynamically reserves slots for each packet, using a random scheduling algorithm. The control channel is used to ensure each node has a queue state information about each receiver-queue in the network (The number of packets queued at each receiver, used so that it is known when a queue is empty). The Random Scheduling Algorithm (RSA) is then used on the queue state information, to dynamically allocate transmission. The system has a high signalling overhead, which severely restricts the number of nodes.

- **HTDM** (Hybrid TDM)
  – (Chipalkatti *et al.*, 1992; Chipalkatti *et al.*, 1993)

  HTDM is a hybrid of DAS and TDM (See Section 2.1.2.4 on page 16), which lowers the high signalling costs of DAS, while being more flexible than TDM, which can not handle bursty traffic.

  Like TDM, HTDM uses frames, except that there are "open" slots, in which any station can transmit (using the DAS scheme). There are N slots that operate under TDM, and M slots available for "open" access. Every $n = N/M$ slots, an "open" slots is transmitted.

- **GTDM** (Group TDM)
  – (Kannan *et al.*, 1994)

  GTDM is also a hybrid of DAS and TDM, much like HTDM (see above). Nodes in this protocol are grouped together, and nodes in a group are only allowed to transmit to nodes in another group in a slot. Inter-group communication is scheduled like TDM, while the intra-group communication is via DAS. A control-channel is needed, however the signalling on the control channel is reduced from DAS and HTDM. This lowers signalling cost, but increases packet delay.

  GTDM in its extreme cases is just TDM or DAS. When every node is a separate group, then GTDM is the same as the TDM scheme. When all the nodes belong to the same group, it is the same as DAS. GTDM has been shown to be better than HTDM, for similar load conditions. GTDM's performance can be improved by careful selection of what nodes to group together, as if nodes that rarely transmit to the same nodes are chosen, then the group does not contend for the same wavelengths.

- **DT–WDMA** (Dynamic Time - WDMA)
  – (Chen *et al.*, 1990)

  Each channel is co-ordinated via a control channel ($CC-FT-TR$). DT–WDMA uses slotted channels, with identical slotting units on control and data channels. Each slot on the control channel has mini-slots, one for each node. A node wishing to transmit must signal on its own control channel, indicating which node it wishes to send to. It then transmits the packet in the next data slot. The receiver on receiving the control information, then tunes into the wavelength of the sender to receive the packet. If more than one node is transmitting, then the one with the largest delay (also indicated in the control channel) is received. The other is lost, in a receiver collision (destination conflict). The lost packet thus must be retransmitted. Maximum throughput of this protocol is limited to 63%, due to receiver collisions, and the need to retransmit lost packets.

- **CF–WDMA** (Conflict Free - WDMA)
  – (Chen & Yum, 1991)

  This protocol uses a control channel, to allow each node to be aware of packet backlog information on the other nodes. With this information transmission can be scheduled to avoid destination conflicts. A high throughput performance is achieved through the processing of transmission, reception and processing of backlog information and the transmission and reception of data packets simultaneously in a pipeline operation. In low traffic conditions the packet delay need only be one slot larger (due to scheduling), than other protocols operating without scheduling.

- **Humblet '93**
  – (Humblet *et al.*, 1993)

  This protocol uses N control channels, one assigned for every node. This means that in all $2N$ wavelengths are needed. The control channels are $TT - FR$, while the data channels are $FT - TR$.

  This protocol supports connection-oriented traffic, as well as datagram traffic. There is a low processing cost involved, and a high throughput is acquired. The network does need synchronisation to a small fraction of the slot length.

- **Starnet**
  – (Poggiolini & Kazovsky, 1991; Kazovsky & Poggiolini, 1993)

  Although implemented on a passive star, this is logically a ring network. It is a $FTTT-FR$ network. Access control is governed by a known protocol (such as FDDI). Starnet is designed to support traffic of widely different speed and continuity characteristics. In 1993 work was in progress towards an experimental demonstration of a 3 Gbps per node network.

- **Quadro** (Queueing Arrivals for Delayed Reception/Routing Operation)
  – (Chlamtac & Fumagalli, 1994)

  Quadro applies optical packet switching, without optical processing devices. It uses switched delay lines (SDL) to implement this kind of switching. SDLs were originally proposed as a way to prevent collisions in channels in WDM networks. Quadro can be either a single hop or multihop network.

2.1.2.4 Tunable Transmitter and Tunable Receiver

This is the most flexible strategy of the four outlined. It allows the number of nodes to exceed the number of available wavelengths. It also suffers most from the limitations of the current technology of tunable transmitters and receivers.

As with $FT-TR$, co-ordination is necessary to set up a channel, and so there is extra overhead involved. This strategy is, like $FT-TT$, open to destination conflict.

The following is a list of $TT-TR$ proposals:

- **TDM** (Time Division Multiplexing)
  – (Chlamtac & Ganz, 1988a)

  TDM preassigns slots (e.g., assuming a uniform load), so each node takes its turn to transmit to every other node, according to its needs. When combined with WDM this permits high channel utilisation. There are minimum signalling costs, as each transceiver knows where to tune to. TDM does not respond dynamically to queueing delays, and has rather inefficient bandwidth use. It can not cope well with bursty traffic, as it has no means to change the assignment of slots.

- **RCA** (Receiver Collision Avoidance)
  – (Jia & Mukherjee, 1991)

  Adding some intelligence to receivers, makes it possible to avoid and resolve receiver collisions (destination conflict) at the data link layer. RCA uses mini-slotted control channel. The RCA protocol accounts for non-zero tuning times of transceivers. To cut down on the number of transceivers needed, only one tunable transmitter and one tunable receiver is needed per node. The maximum throughput of this system was 36%.

- **Look-ahead reservation**
  – (Wong & Yum, 1988)

  This is another protocol with a control channel ($CC - TT - TR$). Each node has a table which stores the status of all the wavelength channels, and updates it using the control channel. Access to the control channel follows the rules of slotted ALOHA. If a node wishes to transmit, it looks for an idle data channel; if there is none, then it contends on the control channel for such a channel.

- **POPSMAC** (Passive Optical Packet-Switched Medium Access Control)
  – (Hou *et al.*, 1996)

  POPSMAC is part of the Rainbow project. It is reservation based, which allows pairs of transmitters and receivers to establish connections. In (Hou *et al.*, 1996) an improved version of the original POPSMAC protocol was proposed, by avoiding collisions of control packets. The results show a better channel utilisation and lower packet delays. As part of their analysis they derived the distribution of packet delay. This protocol is effected by the propagation delay, which has consequences if it is used for larger networks.

- **PROTON**
  – (Levine & Akyildiz, 1995)

  This network uses a control channel, with the slot size the same as on the data channels. Each node contends, using a collision-free procedure for a slot on a data channel. Because it is a $CC - TT - TR$ scheme , there is no limit to network size. The effects of non-zero propagation delay and tuning times are taken into account.

- **HYPASS**

  – (Arthurs *et al.*, 1988)

  HYPASS is an extension of FOX, with tunable transmitters and receivers. This modification vastly improved throughput.

- **MARKAB** (Arabic for *carrier*)

  – (Semaan, 1993)

  This is a network using a time-frame based protocol with five control slots, and a fixed number of data slots. The data slots are reserved for free. Reserved slots are used for longer transmissions (e.g. video traffic), while the data slots are used for single packer transmission or packet reservation.

2.1.3 Summary

Comparison of $TT-FR$ and $FT-TR$ can be split into two classes. Those with pre-transmission co-ordination (control channel(s)), and those without.

The sender must continually send requests, and the receiver scan the wavelengths until a connection is made, because the destination address (wavelength) in $FT-TR$ is unknown in advance. This is inefficient. To set up a transmission, an extra set of transmissions is then required from the receiver to indicate it has seen the request. The need to scan all the wavelengths means that to scan all the wavelength, the tuning time would be 0.3 ms (with a wavelength spacing of 1 nm, a fast tunable-filter of 10 μs, with a 30 nm tuning range, has 30 available wavelengths (Section 2.4 on page 26)).

On the other hand, $FT-TR$ guarantees a collision-less connection. $TT-FR$ is much simpler, you tune the transmitter to the receiver's wavelength, and transmit. However receiver collisions can occur, which can be costly to recover from. The cost of setting up transmissions in $FT-TR$ is thus balanced with the cost of recovering from collisions in $TT-FR$.

$CC-FT-TR$ and $CC-TT-FR$ are very similar, and almost symmetric. $CC-FT-TR$ can use the "tell–and–go" policy (Humblet *et al.*, 1993), where the sender transmits on the control-channel, then immediately sends its packet. When the receiver sees the message on the control channel, it immediately tunes its receiver to the appropriate wavelength. This saves on the round-trip delay that $FT-TR$ suffers from. This method has two problems. Firstly, the receiver must tune its receiver in time to receive the packet, and secondly, if two nodes send a packet at the same time, the receiver can only tune into one of them. The second situation is called "receiver collision" or "destination conflict".

$CC - TT - FR$ is collision free, if the protocol uses the control channel appropriately, but, this leads to extra time taken in using the control channel.

Again there is a balance of the extra time needed on the control-channel, and the recovery or prevention of collisions. The applicability of using a control channel is discussed in Section 2.3, page 23.

$FT - FR$ and $TT - TR$ protocols, although the ideal solution due to their flexibility are not as practically implementable due the constraints on network cost.

2.1.4 Alternative Classification

Previously WDM single hop protocols were classified by their transceiver hardware. Another classification can look at how they deal with collisions. They can be firstly be classed according to whether or not they allow collisions to occur. They can then be sub-classed according to whether or not they use a control-channel, or specialised hardware in dealing with collisions. Table 2.2 shows protocols grouped in this way.

From Table 2.2 we can see five protocol based methods of medium access control (not including the use of of specialised hardware). Their characteristics can be summarised as follows, with a '+' indicating a positive feature, and a '−' meaning a negative feature:

- *Random Access:*
  - \+ Simple to implement, with minimal hardware, as in "FOX" and "HYPASS".
  - \+ Good performance at a low load.
  - − Performance degrades at medium to high loads, due to an excess number of retransmissions needed.
  - − DT–WDMA partially overcomes the lowered performance at higher traffic load, but the need for a control-channel restricts the network size to a small number of nodes.
  - − Collisions in channels are destructive, and no packet gets through,

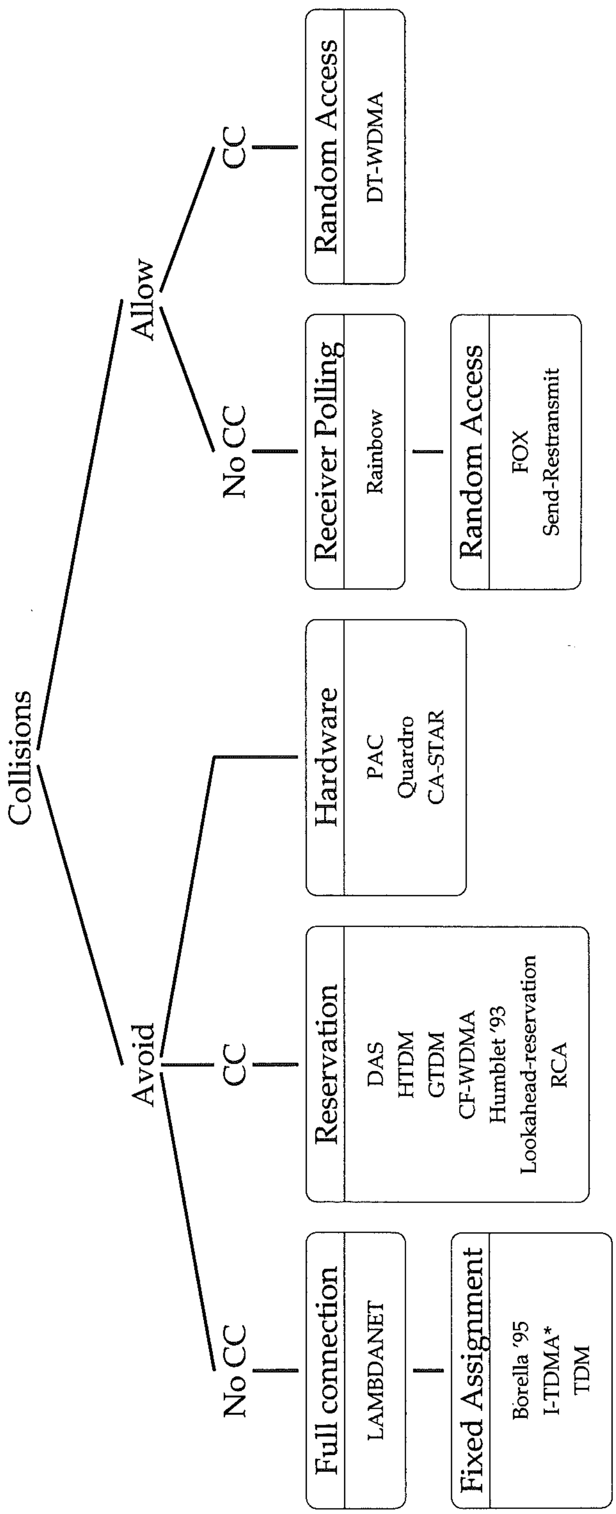

Figure 2.1: Classification of WDM single hop star networks according to collision avoidance strategy.

Classification According to Collisions Strategy		
Allow Collisions		
No Control-channel	**Control-channel**	
Random Access:	*Random Access:*	
FOX Send-Retransmit	DT-WDMA	
Scan-Setup-Transmit:		
Rainbow		
Avoid Collisions		
No Control-channel	**Control-channel**	**Hardware**
Full Connection:	*Reservation:*	PAC Quadro CA-STAR
LAMBDANET	DAS HTDM	
Fixed-Assignment:	GTDM	
Borella '95 I-TDMA* TDM	CF-WDMA Humblet '93 Lookahead-reservation RCA	

Table 2.2: Classification of WDM single hop star networks according to collision avoidance strategy.

however collisions are detectable. Destination conflict while not being destructive (one packet is successfully received), is harder to recover from unless there is a control-channel.

- *Scan-Setup-Transmit:*
  - The round-trip delay has a large affect on throughput, due to the need for setting up connections.
  - A source transmitter has know way of knowing if the destination receiver fails, meaning it may become stuck transmitting to an unavailable receiver.
- *Full Connection:*
  + Fully connected logical topology allows maximum throughput.
  - A large array of transmitters or receivers makes this type of network costly.
  - Does not scale well, as a new transceiver is needed for each additional node when a new station is added.

- *Fixed Assignment:*
  - + Good performance at a high throughput.
  - – Inflexible to changing network demands.
  - – Does not cope well with non-uniform (bursty) traffic.
  - – Does not scale well, and reconfiguration is needed if a station as added.
- *Reservation:*
  - + Adjustable to changing network demands.
  - – The algorithms needed to do the reservation are often complex.
  - – The need for a control-channel adds an extra round-trip delay, which has a large affect on the packet delay.
  - – Extra hardware is generally needed for the control-channel, which increases the the cost of such a network.

The ideal network emerging from such comparison, is one with good performance at all levels of traffic, that does not need expensive hardware (preferable one tunable and one fixed tuned component at each node), with performance insensitive to the round-trip delay and good scalability. Such a protocol may well be a hybrid of protocols (Section 2.5, page 27).

2.2 Multihop networks

Multihop networks differ from each other mainly in their logical topology. This has a major bearing in their effectiveness and properties, even if their physical topology is still a star with a passive coupler. The way the nodes are logically inter-connected gives different routing and performance characteristics.

Multihop networks (Mukherjee, 1992b) can use deflection and packet passing through intermediate nodes, to lessen the need for expensive and slow tunable transceivers. This leads to point–to–point links with fixed transceivers. Using fixed wavelengths reduces the time otherwise needed for tuning the transceivers. To enable the passing of messages through intermediate nodes, a limited amount of processing of packets is needed at each node. This must be done quickly, reading as few bits as possible, so as not to affect the network speed overly. Otherwise significant delay may be introduced.

There is a large slow down of effective (optical) transmission rate (about four orders of magnitude) if electronic data processing is done on intermediate nodes. The current limitations in the area of optical data processing also must be considered.

Some single hop networks mentioned in the previous sections can be used as multi hop networks, as well. For example, Quadro was designed to be either. Similarly, while AMTRAC is a single hop network, one variation of the protocol can be used with multihoping, if this gives a lower packet delay time.

The $FT - FR$ configuration is used predominantly in multihop WDM networks. With packet passing, they do not need the many transceivers that single hop networks need.

2.3 Main Design Issues

The design and analysis of WDM networks has many issues that must be addressed, to construct an efficient protocol.

- **Tuning configuration:** Ramaswami (1990), showed that a tunable transmitter and tunable receiver pair offers better network utilisation. This configuration is more readily adaptable if $N > M$, where N is the number of stations and M is the number of available channels (wavelengths). However the cost of two tunable transceivers per station is currently quite large, so most research is in protocols that rely on only one tunable component.

  This leads to the question of which of the two transceivers should be made tunable. Without a control-channel, $TT-FR$ and $FT-TR$ are significantly different. However with a control-channel, the configurations are essentially symmetric. Currently $FT-TR$ is the prevalent method, as tunable receiver technology is better than that of transmitters. Recently however more $TT - FR$ have been proposed.

- **The use of wavelengths:** The state of current technology means that the number of available channels of a single transmitter–receiver pair can be limited to less than 30[1]. With some protocols requiring $2N$ channels, this can limit network size to a maximum of 15 stations. The lack

[1] With a wavelength spacing of 1 nm, a fast tunable-filter with a 30 nm tuning range, has 30 available wavelengths (see Section 2.4, page 26).

of assumption on the maximum number of channels available is in fact limiting, since a proposed protocol may not be practically implementable.

- **Tuning times:** The assumption on the length of tuning times of tunable transceivers is not trivial, because it can have a marked affect on the performance of the network. Early studies over-estimated performance, when they ignored tuning times. More recent studies have taken tuning times into account, and looked for methods to minimise their effect (Azizoglu *et al.*, 1995). Recent TDM solutions have tried to schedule transmissions so as to limit the affect of tuning latency.

- **Control-channels:** These are used to avoid collisions and ensure fairness, but may require extra transceivers to be implemented (some protocols use N control channels). It can take up bandwidth from data channels, and because of the large value of a, if the control channel causes excess time setting up packet transmissions, it can be very expensive. Certain techniques such as those based on the "tell–and–go" policy (Humblet *et al.*, 1993), where the node transmits its intention to send a packet on the control channel, then immediately sends it, without waiting for acknowledgement that the destination is ready to receive, can alleviate this problem. But, "tell–and–go" allows collisions to occur.

- **Multiple transmitters and receivers:** Many proposed protocols require multiple transmitters and receivers at each node. Although this can give better performance, the cost involved can make it practically unrealistic. On the other-hand, the cost of devices is falling, so in the future having multiple transceivers at each node may become practical. Recent literature has placed emphasis on having one transceiver pair per node, with only on tunable element.

When designing a WDM network, there are some existing limitations. Some of these are typical to networks in general, while others are caused by WDM's optical and high-speed nature.

- **Tuning times:** The state of current technology severely limits the performance of WDM networks. If tuning times are to be fast, then a limited number of wavelengths is available. In a $TT - TR$ network, the large bottleneck in the tuning times would becomes readily apparent.

- **Tuning range:** If a high tuning speed is wanted, then there is a limit of only about 15 channels per transceiver. This limits the $TT - FR$ and $FT - TR$ networks to only 15 nodes, which is quite small for practical use.

- **Relative propagation delay:** (the ratio of propagation delay and packet transmission time, a) The large value of a means that, unlike Ethernet and other relatively slow networks, there can be many packets in transit at one time. This means that the optical fibre acts as a form of storage medium. To use the bandwidth effectively, this storage effect has to be taken into account. The effect is especially noticeable with complex signalling, as the time spent waiting for a reply wastes a lot of bandwidth. Thus if a protocol acts with a round-trip delay, then its performance can be impaired.

- **Scalability:** WDM networks have been designed for the LAN, MAN and WAN environments. However WDM LAN networks, with a high signalling demand are not readily scalable to WANs, due to high costs occurring when relative propagation delay is large.

- **Destination conflict:** is a problem unique to WDM networks. If more than one node transmits on separate wavelengths simultaneously, then a tunable receiver can intercept only one transmission. The other transmissions will be lost. The protocol must recover any packets lost due to such collisions. Destination conflict affects the throughput of the network, although not all studies published have accounted for it in their analysis and results. Chen *et al.* (1990) stated that due to destination conflict, only 60% throughput can be achieved. Also recovering from conflicts can take up to 15 round-trip delays, due to re-transmissions.

- **Packet collisions in channels:** Many WDM networks must cope with possible collisions of transmitted packets in communication channels. Collisions must be dealt with quickly, or the packet delay will be large. Many packets are likely to be transmitted during the time taken to re-transmit a collided packet. This form of collision is more readily detectable than destination conflict, but any collision is destructive, as none of the colliding packets will be received by the receiver.

- **Establishing efficient connections**: It is hard to efficiently set up a dynamic link between nodes, as the signalling required lowers the throughput, and the processing time is relatively slow. If the connections are setup in advance (TDM) then they do not necessarily use the bandwidth efficiently.

2.4 Technology

One of the major limitations effecting WDM networks is the technology. WDM networks require communications hardware that is on the edge of what is currently available. To be practical, single hop WDM networks require transmitters or receivers with fast tuning times and large tunable bandwidths. This allows for minimal delay due to tuning times, and access to many wavelengths. However the use of the best technology is offset by the high cost of such hardware.

2.4.1 Fibre optics

Fibre has a usable bandwidth of about 50 THz, between the wavelengths 1.2 μm and 1.6 μm (Brackett, 1990). This is made up of two windows of low attenuation. One is in the area of 1.5 μm, which has an attenuation of 0.2 dB/km. Within a wavelength range of about 200 nm wide (giving a bandwidth of 25 THz) this attenuation is stable at below 0.4 dB/km. The second window is around the 1.3 μm range, and also has a bandwidth of around 25 THz (Green, 1993).

Dense WDM networks are those that have a wavelength spacing in the order of one nm (Brackett, 1990). Conventional WDM networks used separate transmitter/receiver pairs at each wavelength allowing a much wider spacing. The use of dense WDM networks allows tunable transceivers to access many channels.

2.4.2 Tunable Filters

The speed and range of tunable transceivers is a major bottleneck of current optical networks. Transmitters and receivers have tended to have a fast tuning time or a large bandwidth, not both. Tunable Wavelength Filters are one of the most important building blocks of WDM networks (Sneh *et al.*, 1995). The filter can be the tuning element in receivers and transmitters. The ideal filter has a fast tuning speed, wide tuning range, low power consumption and a low cost.

Nematic liquid crystal (LC) Fabry-Perot filters have a wide tuning range, low power consumption and low cost, but their tuning time is in the range of milliseconds. Fiber Fabry-Perot (FFP) filters are faster (tuning in hundreds of microseconds), but still too slow for a practical WDM network. Acoustooptic tunable filters (AOTF) are fast (tuning speed of less than 10 μs), but have a broad passband resolution (1.3 nm at the 1.5 μm wavelength, i.e. the channel spacing has to be quite wide), and its power consumption is relatively high. A chiral smectic liquid crystal (CSLC) Fabry-Perot filter (Sneh *et al.*, 1995), has a tuning speed of less than 10 μs and a tuning range of around 30 nm, in the wavelength of 1.5 μm. These are very promising performance characteristics for WDM networks.

Lasers and filters are the subject of intensive research (Coquin *et al.*, 1988; Cheung *et al.*, 1989; Shimosaka, 1989; Kobrinski *et al.*, 1990b; Kobrinski *et al.*, 1990a; Illek *et al.*, 1990; Toba *et al.*, 1990; Brackett, 1990; OFC, 1992) , and faster transmitters and receivers of broader range are expected to be available soon.

2.4.3 Star-couplers

Star-couplers are well understood and practically implementable as the base of WDM networks, at not too much expense. The simplest passive star-couplers are constructed with stages of 2×2 couplers to form a $N \times N$ star, where N is a power of 2 (Brackett, 1990). The number of stages is $log_2 N$, while the number of 2×2 couplers required is $\frac{N}{2} log_2 N$ [Figure 2.2].

2.5 Hybrid Protocols and integrated services

Hybrid protocols combine different aspects of two or more other protocols trying to make profit of their best features. For example, HTDM is a combination of the DAS and TDM (Section 2.1.2.3). It merges the low signalling costs of TDM, with good performance of DAS handling bursty traffic, resulting in a hybrid which suffers less from the weaknesses of the two. TDM and DAS are ideal to combine, since TDM has low signalling but can not handle bursty traffic, while DAS can handle bursty traffic but has high signalling costs.

Finding new hybrid protocols is potentially rewarding, if we can get the best

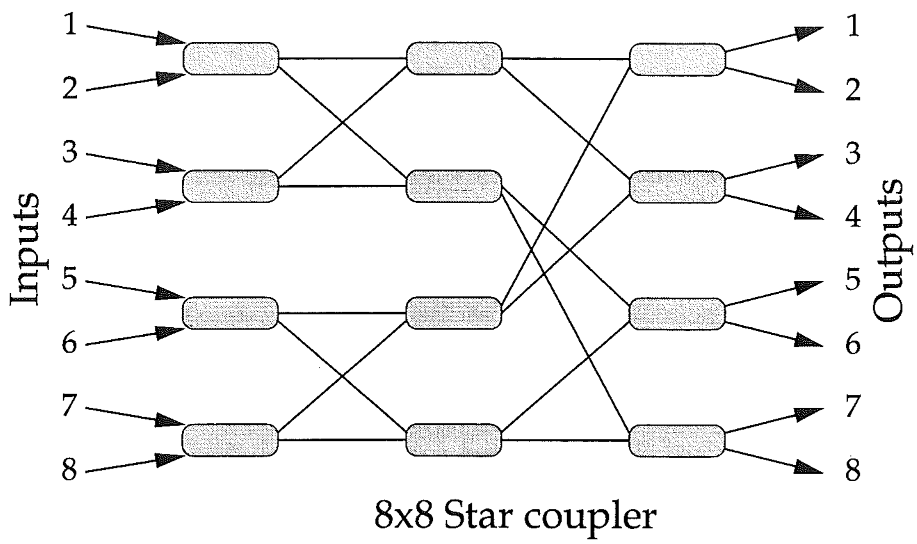

Figure 2.2: A N=8 star network, made up of three stages and twelve 2×2 couplers to form a 8×8 star coupler.

of the component protocols. Integrated services could be one area, to benefit from such an approach. Combining a good protocol proposed for asynchronous data with a good protocol for synchronous or isochronous data (maybe even taking protocols not from the WDM domain) could provide a simple and effective way of implementing integrated services, rather than having to design a new protocol directly.

Chapter 3

WTDMA with trespassing

In this chapter we propose a new class of protocols for WDM star networks, incorporating the new concept of trespassing. The basic WTDMA protocol is explained in section 3.1, then trespassing is described in section 3.2. The new protocol, WTDMA with trespassing is then described section 3.3 and the issue of Collision Avoidance is discussed in
section 3.4.

3.1 WTDMA

In WTDMA networks, during each time slot each station can transmit uncontested on different wavelengths. There are two types of such networks, one uses random allocation of wavelengths, the other allocates wavelengths deterministically. These are called WTDMA/R and WTDMA/D respectively.

In the case of WTDMA/R, the wavelength allocated to a station for transmission during a given slot is decided randomly. Given all stations use the same pseudo-random number generators and the same seed, and have perfect synchronisation, then it is possible for all stations to transmit packets without collision. Specific patterns of traffic generated by stations can be satisfied (on average) by appropriate adjustment of the probability distribution of numbers used for wavelength selection.

In WTDMA/D, both time and wavelengths are assigned to stations in a deterministic way, according to an access table (e.g. Figure 3.1). At the beginning of each time slot, a station is allocated a wavelength to transmit on. Thus, in a network with N stations, after $N-1$ slots the wavelength allocated is repeated, which matches the duration of a TDMA frame. In the simplest model, with

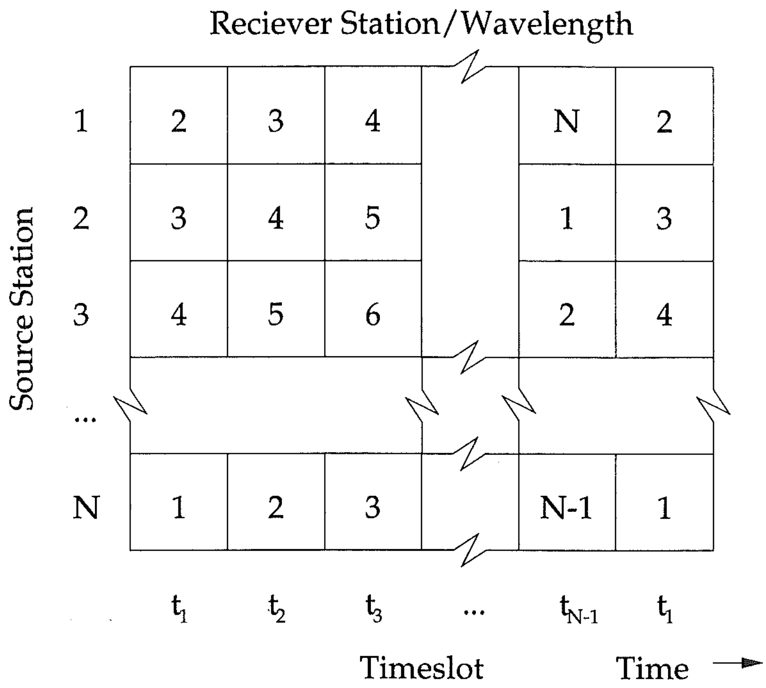

Figure 3.1: Time-wavelength access table of the WTDMA/D protocol ($TT-FR$). The entries in the table indicate the allocated destination station, or equivalently, the wavelength allocated for transmission.

the number of available wavelengths and the number of stations being equal, wavelength λ_i is the receiving wavelength of station S_i, i.e. if station S_j wants to transmit to station S_i, then station S_j has to transmit on wavelength λ_i.

3.2 Trespassing

Trespassing is an extension of the WTDMA protocol, permitting stations to transmit on other-than-assigned wavelengths, when a given station has no packet for transmission on its allocated wavelength.

WTDMA with fixed wavelength and time scheduling is able to provide good throughput performance, especially at high network loads, but the mean delay performance could be unsatisfactory. Random-access techniques provide good mean delay characteristics at lower loads, but at higher loads both throughput and mean packet delay suffer as the network can become unstable. A hybrid of WTDMA and random-access, should be able to provide the best of both worlds. This observation has led to the concept of trespassing, which introduces such a hybridisation.

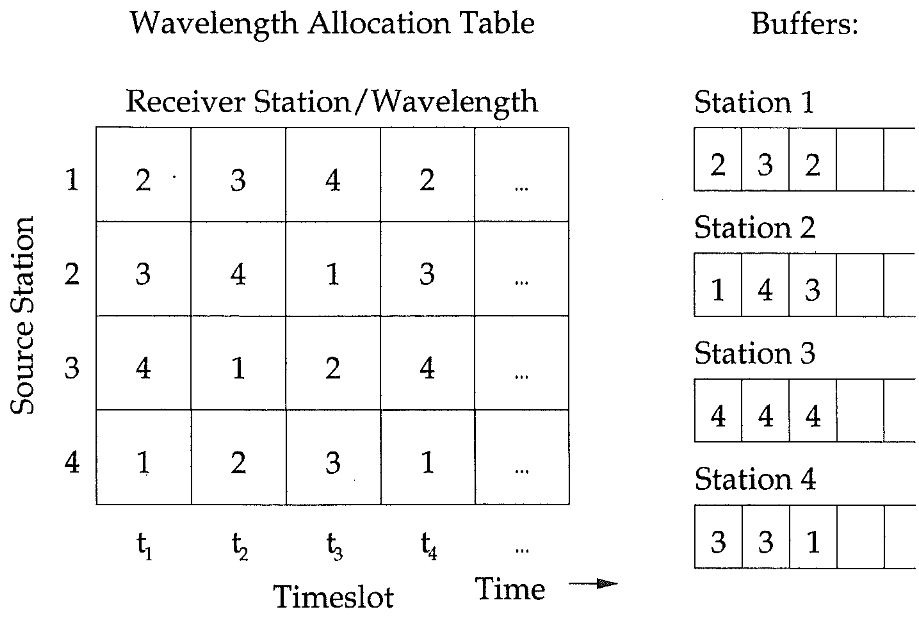

Figure 3.2: Example network, $N = 4$. Entries in the buffers symbolise the destination stations of packets, and the entries in the table are according to figure 3.1

If a station has a packet to transmit to station S_i, then under WTDMA it has to wait for the time-slot in which it is allocated the wavelength λ_i. If S_i has nothing to transmit on its allocated wavelength during a given time slot, then during that time slot that optical channel is wasted. To avoid wastage, if a station is allocated wavelength λ_i (to station S_i), but has a packet ready for transmission to another station, S_j (on wavelength λ_j), then it can "trespass" on wavelength λ_j and send a packet to S_j. Since the wavelength λ_j is allocated for another station during that time slot, such action constitutes trespassing.

For example, let us consider the situation shown in Figure 3.2. Assume that only the packet stored at the head of the queue in a given buffer can be transmitted, the following activities could be observed:

- During the first time slot (t_1): Under WTDMA, only stations S_1 and S_3 can transmit, as they have packets to transmit on their allocated wavelengths. Stations S_2 and S_4 are blocked until the third time slot (t_3). If trespassing is allowed, then all four stations can transmit successfully, providing stations S_2 and S_4 trespass on each other's allocated wavelength.
- During the second time slot (t_2): Under WTDMA, stations S_1 and S_2 can transmit on λ_3 and λ_4, respectively. If trespassing is permitted, stations

S_3 and S_4 can also transmit. However in this case their packets would collide with those from station S_1 and S_2, meaning none of the packets would be successfully delivered to their destinations.

- During the third time slot (t_3): Under WTDMA none of the stations can transmit on their allocated wavelengths. On the other hand, applying trespassing all four stations can transmit their packets successfully.

3.2.1 Trespassing Rule

This example shows that trespassing can potentially lead to a large number of collisions. It can also be unfair, if a station using its allocated wavelength has its transmission disturbed by another station which trespasses. This can lead to performance degradation caused by collisions of legitimate and trespassing packets.

To limit this effect, a probabilistic trespassing rule can be applied, with a station deciding to trespass with a probability (Z). Each time a packet is selected to trespass, then a random number, $Z_0, where 0 \leq Z_0 \leq 1$, is chosen. This is compared with the value of Z, and if $Z_0 \leq Z$ then the packet is allowed to trespass.

The "trespass probability" must be appropriately chosen to limit the number of collisions, but still allow transmissions on unused wavelengths. Trespassing should increase throughput of a network, as the result of using wavelengths that would otherwise remain unused. It should also lower a network's mean packet delay, as packets can be transmitted before they normally would under WTDMA.

3.3 WTDMA with trespassing

Protocols with random access to communication channels have been known since slotted Aloha (SA), one of the simplest random-access protocols. Their major limitation is that collisions between transmitted packets can occur, and at high traffic levels such protocols can become unstable (Hammond & O'Reilly, 1986). A hybrid protocol that combines the good performance of random access protocols at low traffic level and good high traffic level performance of WTDMA thus seems ideal. It should be simple and should not need control-channel(s), if collisions are appropriately dealt with.

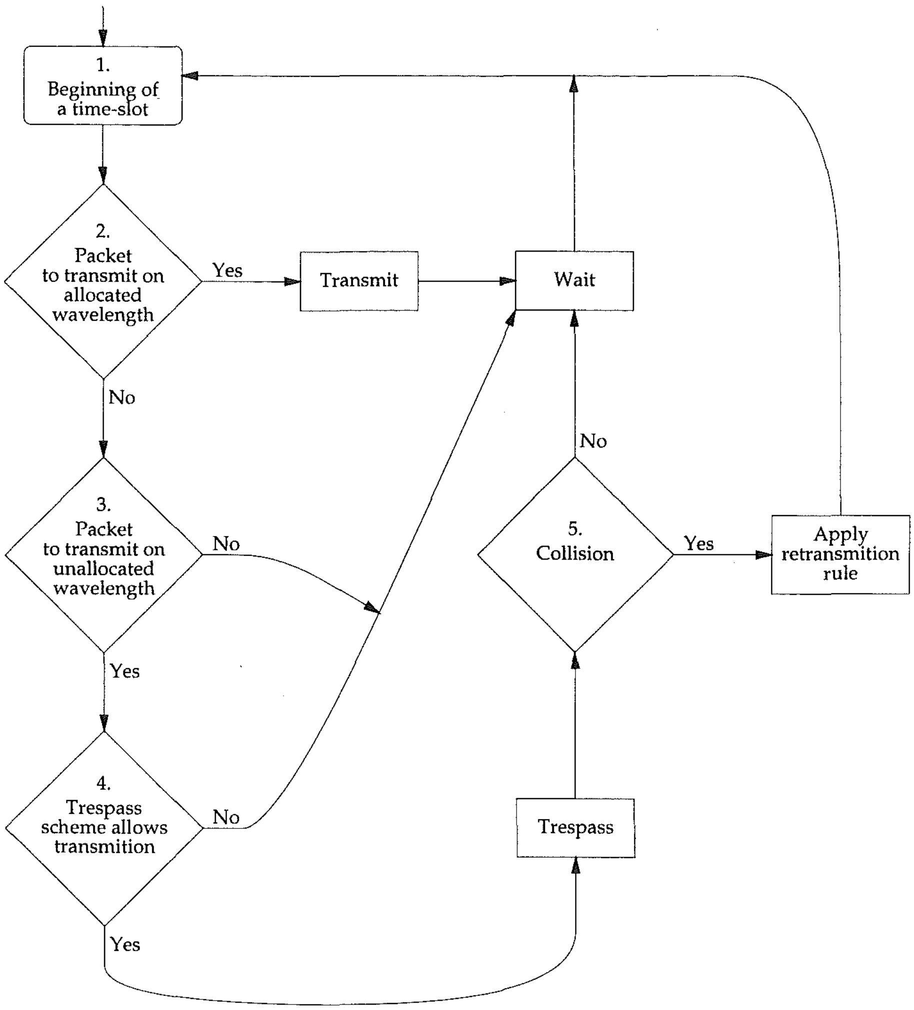

Figure 3.3: Flowchart of WTDMA with trespassing.

The WTDMA protocol with trespassing, applied by stations with appropriate trespassing probabilities, is such a hybrid of fixed WTDMA and random access to wavelengths. The proposed protocol can be implemented either in the $TT - FR$ or $TT - TR$ type architecture. It will not work under $FT - TR$ as then random access is impossible. The aim is to have a low complexity protocol, with no control channel, and a minimum number of transmitters and receivers.

The basic operations at a station working under the WTDMA with trespassing protocol are described [Figure 3.3] as follows:

1. At station S_k, at the beginning of the next time-slot: the TDM wavelength allocation table [Figure 3.1] is checked for the wavelength (λ_i) it is allocated.

2. Check buffer for a packet destined for S_i, to be sent on the allocated wavelength λ_i. If there is a packet ready for transmission to S_i, transmit the packet on wavelength λ_i, and go back to step 1. If there is no packet destined for S_i, continue to step 3.

3. Check buffer input queue(s) at random[1] for a packet destined for any other station (S_j, $j \neq i$). If there is no packet in the buffer, wait until beginning of the next time-slot (go back to step 1). Otherwise continue to step 4.

4. Select a random number Z_0 and compare the probability of trespassing (Z). If $Z_o > Z$ then wait until the beginning of the next time-slot (go back to step 1). Otherwise, transmit the packet to S_j on wavelength λ_j (trespass).

3.4 Collision Avoidance (CA)

Given that the cost of retransmitting collided packets can be large, and possesses various problems, a simple method of avoiding collisions would be extremely helpful. The nature of the passive star-coupler used for WDM networks makes resolving collisions possible.

As has been demonstrated in Yau (1996), it is possible to resolve conflicts while packets are en route to their destinations. This is done by using a "central arbiter", which is placed at the entrance to the star coupler. It can detect conflicts and resolve them, before packets collide at the coupler.

In the case of this study, where Collision Avoidance (CA) is used, it is assumed that if packets are going to collide, then:

- If the allocated station has transmitted a packet, then the packet is allowed to progress through the coupler to its destination, while all other packets are reflected back to their sources. This means that the allocated station has guaranteed service.

- If the allocated station has not transmitted a packet, and more than one station has trespassed on a particular wavelength, then one of the packets is picked to progress through the coupler, while the others are reflected.

[1] Other checking schemes may have better performance, however randomness is assumed initially to simplify analysis.

- If a packet is reflected back to its source, then it is placed back into the front of the queue in the buffer.

The hardware architecture of a buffered central arbiter has been proposed in Yau (1996). Here, instead of being buffered, packets that would have collided are reflected. As reflected packets are not on the same wavelength as their source's receiver, wavelength conversion is needed. The central arbiter needs a built-in wavelength allocation table, so that it is aware of allocated wavelengths for each slot. It also needs to decide on which packet to forward and which to reflect. A one slot delay for each packet may be necessary to accomplish this. If either optical logic or a control channel is available to the arbiter, then it would be possible to keep packets exclusively in the optical domain.

In general, if collision avoidance is implemented, then the value of trespassing should be deterministic ($Z = 1$), as the more (successfully) transmitted packets, the shorter delays and the larger throughput could be achieved. If the central arbiter is limited in its capacity to recover from collisions, then a lower trespass probability should possibly be used.

Buffering packets at the central arbiter instead of reflecting them is also a possibility. If all colliding packets could be successfully be buffered, then this would eliminate the need for wavelength converters. It would also eliminate a round trip delay, and prevent packets arriving at the destination out-of-order. If buffering was used at the central arbiter to store packets for forwarding, then, when the buffer starts to reach capacity, the trespass probability could be dynamically reduced, to allow the buffer to empty. If the buffers becomes full, then trespassing could be stopped, until the buffers become less congested.

3.4.1 Avoidance of out-of-order transmission

For networks with a non-zero propagation delay, it is possible that if a packet is transmitted and then reflected, it arrives back at its source after another packet has been successfully sent to the same destination. This leads to packets arriving at the destination out-of-order. Recovery from packets transmitted out-of-order either requires buffering (undesirable at high transmission rates), or retransmissions. To prevent this, it is necessary to avoid the situation in which packets can be transmitted out-of-order.

With trespassing, this means that if the distance to the coupler is equivalent to d slots, then a station cannot trespass on λ_i, if λ_i is its allocated wavelength within the next $2d$ slots (a round trip). Also after trespassing, it should not trespass again on the same wavelength for $2d$ slots, after which the trespassing packet will have been successfully transmitted, or reflected.

3.5 Conclusions

WTDMA with trespassing could be implemented as an extension to WTDMA on any star network that uses tunable transmitters. If collision avoidance is to be used, then a central arbiter with either a wavelength conversion capability or optical buffering would also be needed. Combining fixed scheduling of wavelengths and time slots with random access to wavelengths gives a hybrid protocol that could provide both the high load performance of WTDMA, and short packet delays of random-access techniques.

Trespassing can be customised to give the best performance for a given load, by altering the trespass probability.

The performance of various versions of WTDMA with trespassing will be analysed in the following chapters. The mathematical model used for analysis and simulation is outlined in Chapter 4, while simulation is covered in Chapter 5. The results of the performance evaluation of version versions of WTDMA with trespassing that are analysed and simulated are shown in the Chapter 6.

Chapter 4

Mathematical Model

4.1 Model

In this section four simple models for analysing WTDMA with trespassing are described. These models are used to obtain numerically results for throughput for WTDMA/R and WTDMA/D, and their equivalents with trespassing.

4.1.1 Assumptions

The analysis considers two different models of stations, assuming that separate queues are formed for buffering packets addressed to different destinations, and that: (i) all buffers are of unit size, or (ii) all buffers are infinite in size; for each of these buffer sizes, two different traffic patterns are assumed: (a) a uniform traffic pattern, with destinations of packets being uniformly distributed, and (b) a non-uniform traffic pattern, where one station has a higher proportion of packets destined to it, while the other $N-1$ stations equally share the remainder of the traffic; giving a total of four models.

4.1.1.1 Model 1

Assumptions associated with the first model are as follows:

A1: There are N stations in the network, and they have N wavelengths available for transmission, so each station has its own "home wavelength".

The number of wavelengths practically available is limited, restricted by the state of technology. Because the protocols proposed are based on WTDMA, it is possible for wavelengths to be shared (i.e. wavelengths can be allocated to more than one station), meaning this assumption is not too restrictive.

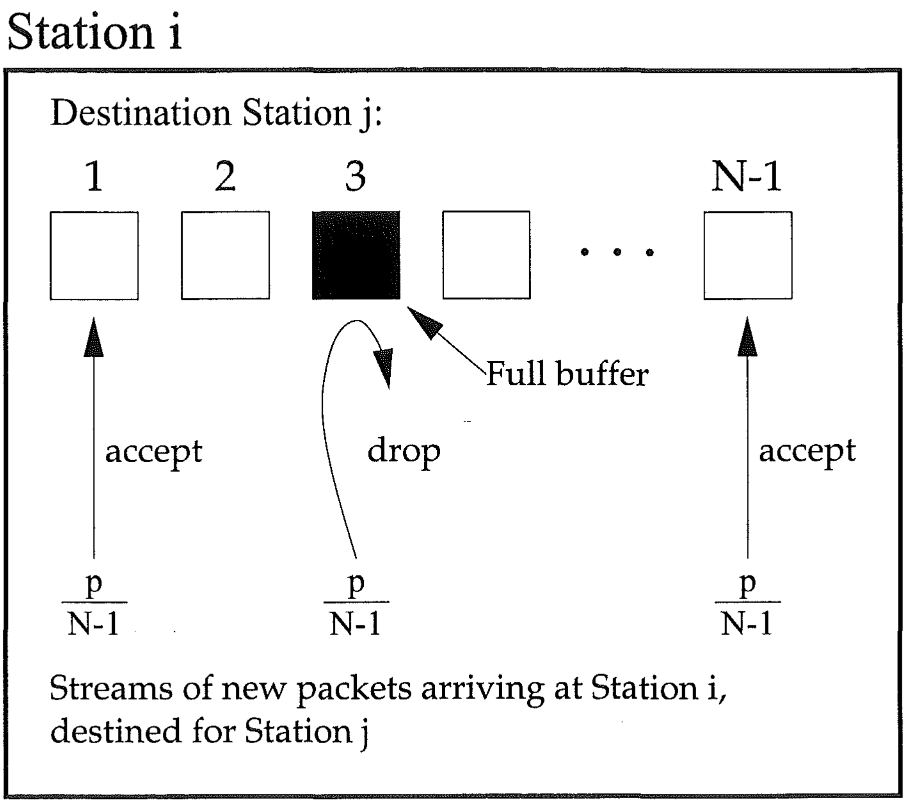

Figure 4.1: Buffer structure at station i (uniform traffic pattern).

A2: Each station has $N-1$ buffers (one for each destination) of unit size to store outgoing packets [Figure 4.1].

A practical implementation of this may in fact involve one buffer, with a capacity of $N-1$, with a pointer for each destination.

A3: Arrivals at each station are modelled by independent and identical Bernoulli processes. In the case of a uniform traffic pattern, the probability that a single packet arrives at a station in a time slot and is destined to another station equals $p/(N-1)$; [Figure 4.1]. Thus, the total arrival rate of packets at a station equals p packets per slot.

The use of independent Bernoulli processes is justified, as it accurately models optical networks, where the transmission rate is faster than the rate at which sources are able to generate data.

A4: A packet can leave the buffer at the beginning of a time slot [Figure 4.3). It is assumed that arrivals of new packets occur immediately after the

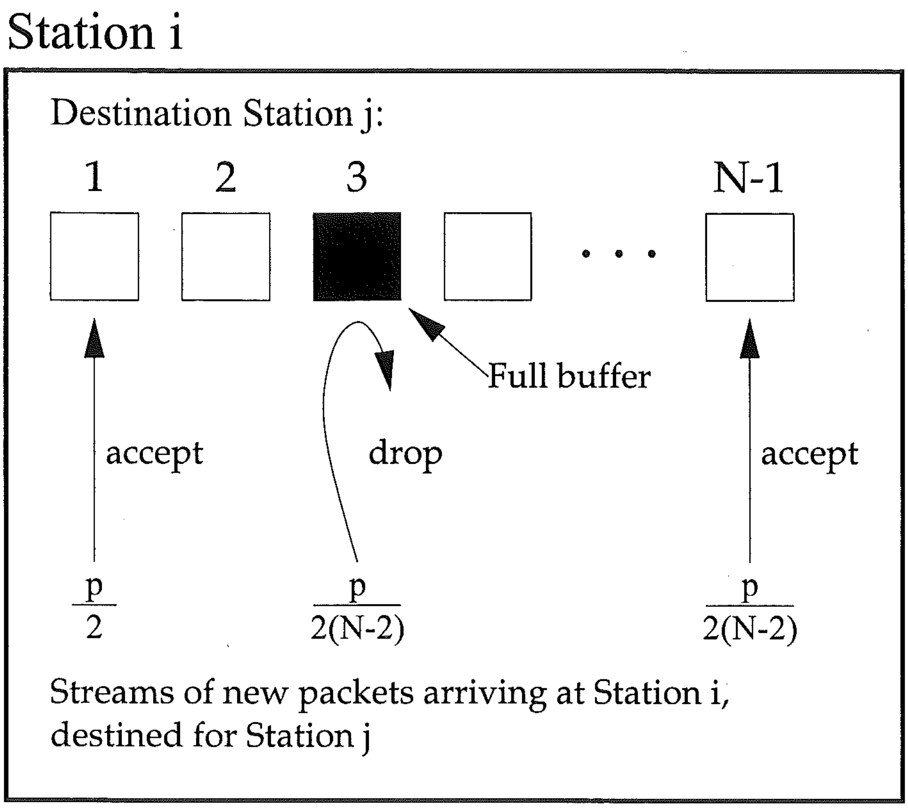

Figure 4.2: Buffer structure at station i (non-uniform traffic pattern).

departure of a packet for transmission (into an intermediate buffer), and a new packet needs the rest of the slot to be fully buffered.

Such an assumption means that a packet that is selected for transmission is placed in an intermediate buffer from which it is transmitted during one slot. This allows arrivals to take place immediately after the beginning of a time slot. As the buffering would most likely be done optically, it is possible that while a packet is leaving the buffer to be transmitted from one end, an arrival will be entering the buffer at the other end.

A5: The offered load (ρ) is uniformly spread over all stations, and equals $\rho = 1.p + 0.(1-p) = p$ for each station, per slot time.

A6: Collisions rates are calculated, but no specific mechanism for recovery is considered unless otherwise mentioned.

Collision avoidance/resolution mechanisms are discussed in Section 3.4.

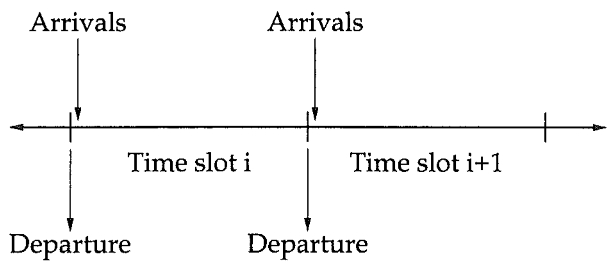

Figure 4.3: Time instances when packets can arrive at and depart from a buffer at a station.

A7: Tuning times and propagation delays are not taken into account, unless otherwise mentioned.

4.1.1.2 Model 2

In the second model, we will assume that the buffering capacity of each station is infinite. This means that assumption A2 will be replaced by:

A2′: Each station has $N-1$ buffers (one for each destination) of infinite size to store outgoing packets.

4.1.1.3 Model 3

In the third model, where the traffic pattern is non-uniform, and buffers are unit size, assumption A3 will be replaced by:

A3′: Arrivals at each station are independent and identical Bernoulli processes, with a non-uniform traffic pattern [Figure 4.2]. One station, representing a server has 50 percent of the load destined for it. The other half of the load is evenly distributed among the remaining stations. Thus, the total load equals $p = 0.5p + (N-2).(0.5p/(N-2))$ packets per slot.

4.1.1.4 Model 4

In the fourth model, where the traffic pattern is non-uniform, and buffers are of infinite size, assumption A3 is replaced by A3′, and A2 is replaced by A2′.

The buffers will be modelled by Markov chains. The results of their analysis can then be used to calculate the traffic generated from a station. From assumption A2, each station has a separate buffer for each destination, and so $N-1$ independent Markov chains are used for modelling each station's buffers. As the stations and buffers are independent, only one buffer needs to be analysed to obtain the global characteristics.

4.1.2 Notation

The following notation will be used in this thesis:

A – the binary random variable equals zero if a station has no packet ready for transmission at the allocated wavelength, and equals one if there is a packet to transmit at the allocated wavelength.

B – the random variable equal to the number of stations trespassing onto a specific unallocated wavelength (λ_i).

Z = the trespass probability.

The following notation will be used in analysis of the states of the stations:

x_i = the number of full buffers at a station at the beginning of time slot i, before transmission.

$\bar{X}_i$ = the mean number of occupied buffers at a station at the beginning of the ith time slot.

t_i = the number of packets transmitted in the ith time slot at a station.

a_i = the number of new packets that arrive during the ith time slot at a station.

d_i = the number of new packets dropped, because the buffer associated with their destination is full, during the ith time slot at a station.

4.2 Conclusions

The adopted assumptions restrict the analysis to networks in which the number of wavelengths available is equal to the number of stations. Although this is quite a restrictive factor given today's technology, where little more than 100 wavelengths are available, it can be expected that further achievements in optical technology should make the proposed network architecture applicable for even larger networks (see section 2.4).

Incoming traffic is modelled by Bernoulli processes. While it is sufficient for analysis in this thesis, it could be improved by assuming more realistic models of teletraffic, such as modulated Bernoulli, modulated Poisson and/or self-similar processes.

Finite tuning times of transmitters are not explicitly taken into account. The latency of a tunable transmitter is not negligible, and relatively large if compared to the transmission time. WTDMA can be optimised to allow the best performance, minimising the effect of tuning times. If trespassing is used, then it must be possible for a trespassing station to tune to a wavelength as fast as the allocated station, to be able to successfully transmit in a slot. It could be assumed that a station is limited in the number of wavelengths it can trespass on, by the tunable transmitter's latency or tuning range.

Propagation delay can have a significant effect on the performance of optical networks, especially as the transmission rates are high, meaning that many packets can be in transit on the communications medium at one time. Thus reservation techniques and collision recovery that require a round trip delay can prove expensive.

The buffers modelled by Markov chains are assumed to be independent. More accurate modelling would involve dependent Markov chains.

Chapter 5

Simulation

5.1 Simulation Model

Quantitative stochastic simulation was performed to confirm the results of mathematical analysis, and to extend them. The same assumptions as the mathematical model were used in simulation (Section 4.1.1).

The network's performance was simulated following the flowchart in figure 5.1, by applying it cyclically through time slots. As simulation time progressed from one time slot to the next, each station tried to transmit, and then checked for arrivals in turn. Once each station had done this, then the simulation model proceeded on to the next time slot.

The transmission process [Figure 5.2] involved checking the buffer associated with the allocated wavelength. If it was full, then the packet was transmitted[1] and removed from the buffer. If it was empty, then an attempt at trespassing on another wavelength could be made.

Trespassing [Figure 5.3] involved searching for an occupied buffer. This was done in the reverse order to the wavelength allocation table. If a packet was found, then it was transmitted with the trespassing probability (Z). If the network operated under pure WTDMA, then $Z = 0$.

The arrival process [Figure 5.4] was executed for each of the $N - 1$ destinations in turn. Firstly it was checked whether a packet arrived. If a packet did arrive, then the buffer it arrived into was checked, and if it was full, then the packet was dropped. If it was empty, then the packet was placed into the buffer.

Measurements were taken at the end of each slot to ascertain:

[1] Transfered to the intermediate buffer of the transmitter.

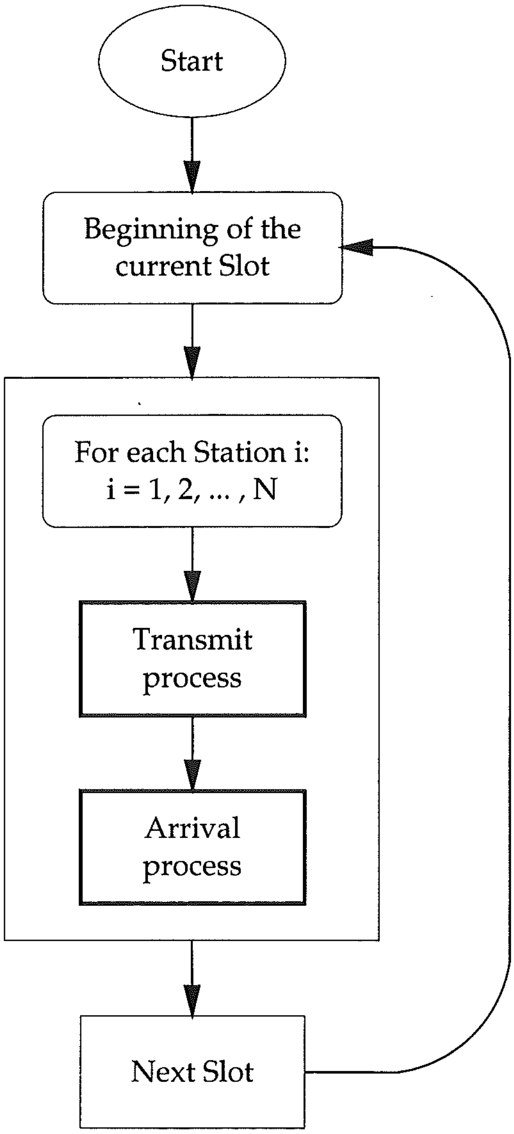

Figure 5.1: Flowchart: Basic processes simulated at each station.

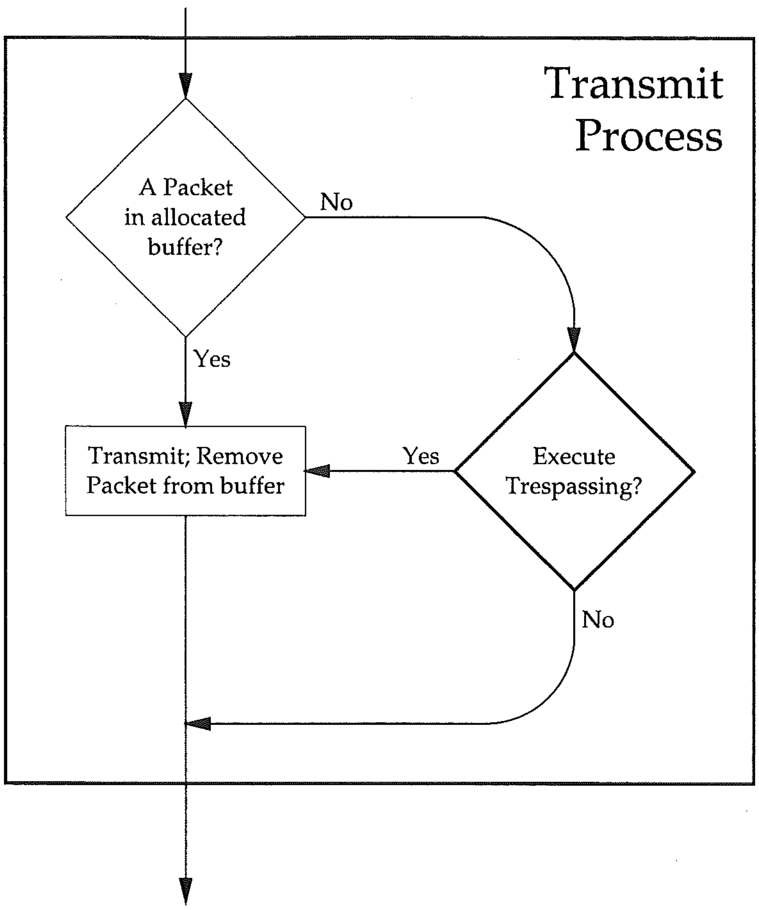

Figure 5.2: Flowchart: Transmit process, a subset of Figure 5.1.

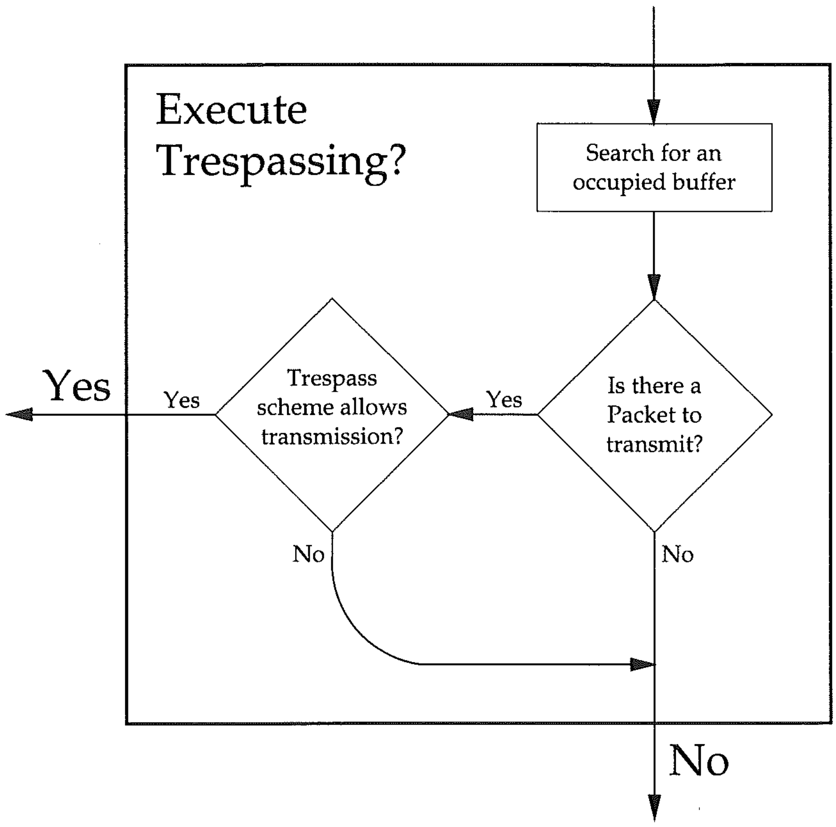

Figure 5.3: Flowchart: Execute Trespassing process, a subset of Figure 5.2.

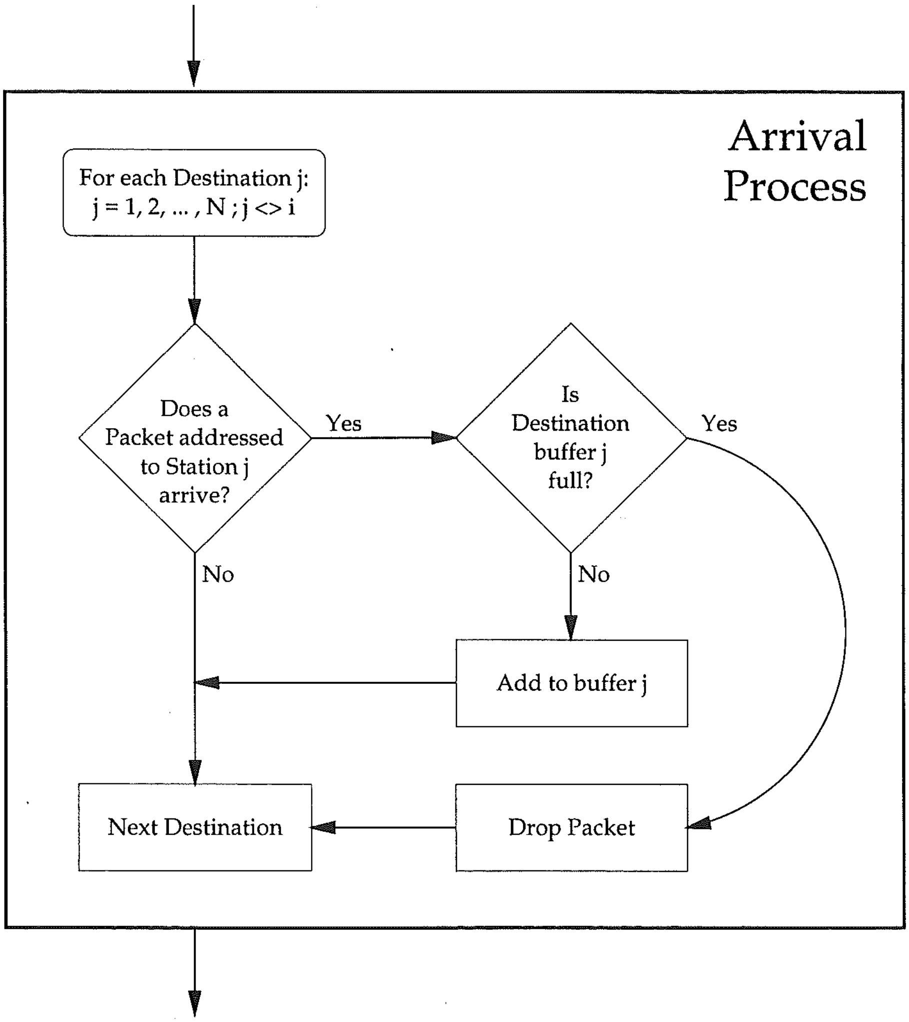

Figure 5.4: Flowchart: Arrival process, a subset of Figure 5.1.

- The number of packets transmitted on each wavelength. If one packet was transmitted, this meant a successful transmission. If more than one packet was transmitted on the same wavelength, this amounted to a collision. This measurement of transmitted packets was taken at the end of each time slot, after transmission and arrivals of packets in the time slot, and just before the beginning of the next time slot.

- The number of packets waiting in each queue. This was done at the same time as the number of packets transmitted was calculated.

- The number of time slots between when a packet arrived in the buffer and it was successfully transmitted. This gives the delay (in time slots) the packet had to wait. At the end of each time slot, each packet that was successfully transmitted has its arrival time slot subtracted from the current time slot, to give the packet delay.

From these measurements we can obtain the throughput per station per time slot, the mean queue lengths, and the mean delay of packets.

5.1.1 Choosing a Trespassing Packet

Whereas in the mathematical analysis, it is assumed that a packet is chosen at random out of a full buffer for trespassing, if the buffer for the allocated wavelength is empty, this is not the best practical method of choosing a packet to trespass with.

In simulation, we assumed that first the buffers containing packets to the farthest destination (in relation to the destination reachable on the allocated wavelength) are checked, and then buffers to closer destinations, finishing with the closest one, are checked. This in effect takes the first full buffer that would have the longest wait until it was served as the allocated wavelength.

This method of choosing a trespassing packet should minimise delays, as the packets that would have the longest wait are used as trespassing packets first (Simulation results showed that differences in choosing a trespassing packet has a small, but insignificant effect on mean packet delay).

5.1.2 Collisions

In simulation, the effect of colliding packets was dealt with in several different ways, depending of the model being used. When there was no collision avoidance, all colliding packets were dropped, and considered lost, with no attempt made to recover them. In real-life, this would not be a practical implementation.

With collision avoidance, all packets that had collided were sent back to their sources, and placed back at the front of the buffer, ready for immediate re-transmission. Practically speaking, this would require wavelength converters at the central-arbiter, to convert the packets from the wavelength received by their destination station, to that received by their source. In the case where propagation delay was also considered, packets that had collided had to wait the number of slots it would take them to propagate back to their source, before being placed back into the buffer.

Packets that are reflected at the coupler could possibly collide with packets addressed to their source. To simplify simulation, it was assumed that all packets were successfully reflected.

5.2 Execution of Simulation

The simulations were performed under Akaroa2 (Ewing *et al.*, 1996), an improved version of the Akaroa package (Yau & Pawlikowski, 1993). Akaroa is a simulation controller for running quantitative stochastic simulations in parallel. Simulation under the control of Akaroa is executed in multiple replications in parallel (MRIP) scenario, i.e., a simulation program is run simultaneously on multiple processors (Pawlikowski & Yau, 1992; Pawlikowski *et al.*, 1994; Ewing *et al.*, 1995; McNickle *et al.*, 1996).

Akaroa launches the simulation program on each processor, detects the end of the initial transient period (Stacey, 1993), and then analyses the steady-state performance by collecting observations from each simulation engine. The observations are collected together by a global analyser, to calculate a global estimate. When all estimates reach the appropriate precision, the simulation is halted, and the final estimates displayed, together with their precision.

In this investigation observations were made to collect estimates for throughput, mean number of collisions, mean queue lengths and mean delay of packets for the various protocols considered. All final simulation results presented in this thesis were obtained with the relative precision of five percent (or better),

Model	Total Number Observations	Transient Period	Percentage Transient
WTDMA/R, unit buffer, $p = 0.7$	4,536	252	5.6%
WTDMA/R, unit buffer, $p = 0.5$	111,000	8,093	7.3%
WTDMA/R, unit buffer, $Z = 1$, $p = 0.8$	4,500	250	5.6%
WTDMA/R, unit buffer, $Z = 1$, $p = 0.1$	311,748	9,688	3.1%
WTDMA/D, unit buffer, $p = 0.9$	4,215	218	5.2%
WTDMA/D, unit buffer, $p = 0.5$	42,534	5,018	11.8%
WTDMA/R, infinite buffer, $Z = 1$, $p = 1$	2,367	263	11.1%
WTDMA/R, infinite buffer, $Z = 1$, $p = 0.1$	413,634	9,707	2.3%
WTDMA/D, server model, infinite buffer, $Z = 1$, $p = 0.001$	1,356	226	16.7%
WTDMA/D, server model, infinite buffer, $Z = 1$, $p = 0.015$	19,686	1,868	9.5%

Table 5.1: The number of observations collected and transient period of various simulation experiments executed when studying performance of WTDMA networks.

at a 95 percent confidence level.

The simulations were run on anywhere between eight and 40 processors, and took anywhere from a few seconds, to a few hours to reach the required precision. The number of observations made before a simulation run ended, varied from only 1,350, to 413,634 observations per given estimate. The initial transient period ranged from between a few hundred observations, to almost 10,000, and anywhere from 2.3 percent of the total simulation run, to 16.7 percent of the total simulation run [Table 5.1].

5.3 Comments

The selection of the same assumptions for analytical and simulation models allowed the comparison of results obtained by mathematical analysis and simulation, to back up the findings of each method of analysis.

Simulation also allowed the analysis of models that were too complex for analytical studies, such as the analysis of a collision avoidance mechanism, inclusion of non-zero propagation delays and prevention of out-of-order transmission of packets, and the traffic pattern based on a server model. Akaroa proved to be an ideal tool for running the simulations for the following reasons:

- It enabled the running of simulation in parallel, which shortened the time taken for a simulation run.

- It allowed the simulation run to be stopped when the final results obtain the required precision.
- It made the complex statistical simulation output analysis automatically.

Chapter 6

Performance Modelling and Evaluation

In this chapter we present performance analysis of for various protocols based on WTDMA/R, and WTDMA/D. These include cases where the propagation delay and protection from out-of-order transmissions are considered. Finally a non-uniform traffic case, where a large proportion of the load is destined for one station is analysed. Specifically, the following sections the following cases were evaluated for their throughput per station per slot, assuming both unit and infinite buffers:

- pure WTDMA/R.
- WTDMA/R with trespassing for: $Z = 1$, $Z = (1 - p)/N$ and $Z = 1$ with Collision Avoidance (CA).
- pure WTDMA/D.
- WTDMA/D with trespassing for: $Z = 1$, $Z = (1 - p)/N$ and $Z = 1$ with Collision Avoidance (CA).
- WTDMA/D with deterministic trespassing ($Z = 1$) and Collision Avoidance (WTDMA/D, $Z = 1$, CA), with propagation delay and no out-of-order transmission (unit buffer only).
- WTDMA/D and WTDMA/D with trespassing, $Z = 1$ and CA, under a non-uniform server model of traffic.

The following cases were also analysed for their mean delay performance, with both unit and infinite buffers:

- Pure WTDMA/R and WTDMA/R, $Z = 1$ and CA.
- Pure WTDMA/D and WTDMA/D, $Z = 1$ and CA.
- WTDMA/D with trespassing, $Z = 1$ and CA, with propagation delay and no out-of-order transmission (unit buffer only).
- WTDMA/D with trespassing, $Z = 1$ and CA, under a non-uniform server model of traffic.

Pure WTDMA/x means that no trespassing or collision avoidance mechanism is applied. In all cases the network is assumed to have $N = 100$ stations.

6.1 WTDMA/R, Unit buffer

In this section WTDMA/R, with unit destination buffers is considered. Then its performance is compared with WTDMA/R with trespassing, $Z = 1$ and$(1 - p)/N$, as well as with WTDMA/R with trespassing, $Z = 1$ and collision avoidance.

6.1.1 Mathematical Analysis

Two alternative methods of analysis are used to attain the mean number of occupied buffers at a station at the beginning of a time slot. The first is based on analysis of the mean values in steady state, the second is based on a Markov chain analysis. These were then used to obtain the throughput.

6.1.1.1 Method of mean value analysis

Assuming that stations operate under the fixed wavelength and time slot assignment rule of the WTDMA protocol, we can get the formula for the global state of the queues at a given station at the beginning of the ith time slot as:

$$x_i = x_{i-1} - t_{i-1} + a_{i-1} - d_{i-1} \qquad (6.1)$$

Where x_i is the total number of packets in $N - 1$ buffers at a given station at the beginning of the ith time slot. A packet is transmitted from that station during a given time slot whenever the buffer associated with the allocated wavelength is occupied. Thus, in steady state, the average number of transmitted packets is equal to the average number of packets in that buffer, i.e.:

$$\bar{t}_{i-1} = \frac{\bar{x}_{i-1}}{N - 1} \qquad (6.2)$$

Following the assumption A3, the arrival rate is simply given by p:

$$\bar{a}_{i-1} = p \tag{6.3}$$

The probability that a packet is dropped is given by the probability that an arriving packet sees its buffer is full. In this case this means that the mean number of dropped packets equals the mean number of full buffers ($\bar{x}_{i-1}$) minus the average number of transmissions from the station during a time slot (equal to: $\bar{x}_{i-1}/(N-1)$), divided by the total number of buffers ($N-1$), and multiplied by the arrival rate p:

$$\bar{d}_{i-1} = \frac{p}{N-1}\left(\bar{x}_{i-1} - \frac{\bar{x}_{i-1}}{N-1}\right) \tag{6.4}$$

Taking into account equation (6.1) we get the following:

$$\bar{X}_i = \bar{X}_{i-1} - \frac{\bar{X}_{i-1}}{N-1} + p - \frac{p}{N-1}\left(\bar{X}_{i-1} - \frac{\bar{X}_{i-1}}{N-1}\right) \tag{6.5}$$

Since, it is assumed that the network is in a steady state (for $t \rightarrow \infty, \bar{X}_i = \bar{X}_{i-1} = \bar{X}$):

$$\frac{\bar{X}}{N-1} = p - \frac{p}{N-1}\left(\bar{X} - \frac{\bar{X}}{N-1}\right) \tag{6.6}$$

or simply:

$$\bar{X} = \frac{p.(N-1)}{1 + p.\dfrac{N-2}{N-1}} \tag{6.7}$$

Note that:

$$\text{for } p \rightarrow 0, \qquad \bar{X} \rightarrow 0, \tag{6.8}$$

$$\text{while for } p \rightarrow 1, \quad \bar{X} \rightarrow \frac{(N-1)^2}{2N-3} \simeq \frac{N}{2} \text{ if } N \gg 1 \tag{6.9}$$

This means that even at high traffic, only up to half the station's buffers will be full, on average.

6.1.1.2 Method of Markov chains

If we use two state Markov chains to model each destination buffer in a station (Figure 6.1), then we get a total of $N-1$ identical Markov chains (Figure 6.2).

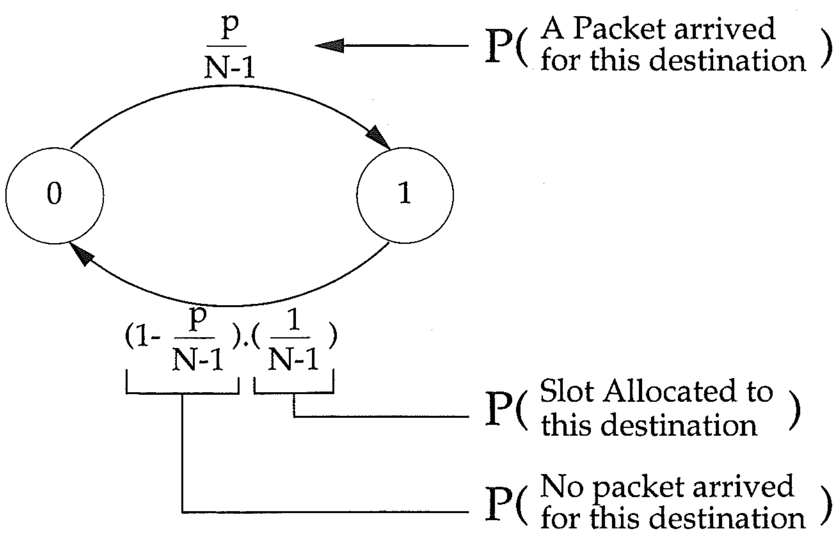

Figure 6.1: A two state Markov chain for the state of a destination buffer at a Station, for WTDMA/R with unit buffers.

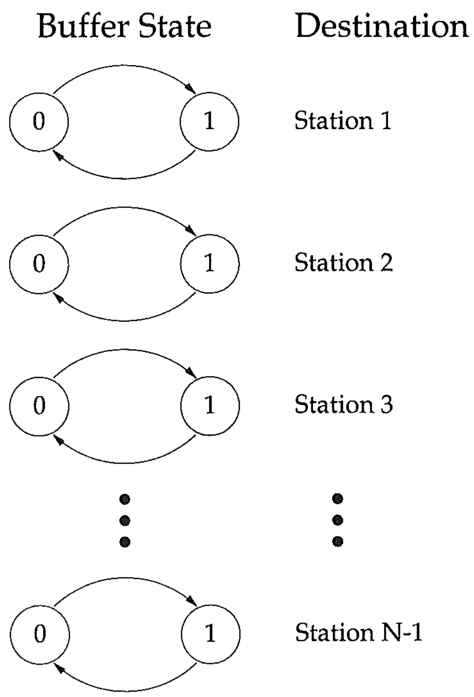

Figure 6.2: $N-1$ independent two state Markov chains as a model of buffers at a Station.

To get the probability that a buffer is empty (P_0), and the probability that it is full (P_1), not that:

$$P_0 + P_1 = 1 \tag{6.10}$$

In steady state the arrival and departure rates from each state are equal. Thus, we can then get the following probability balance equation:

$$\frac{p}{N-1}.P_0 = \left(1 - \frac{p}{N-1}\right).\frac{1}{N-1}.P_1 \tag{6.11}$$

From (6.10) and (6.11) we get:

$$P_1 = \frac{p}{1 + p.\frac{N-2}{N-1}} \tag{6.12}$$

From this, the mean number of packets awaiting transmission to all $N-1$ destinations is $\bar{X} = (N-1).P_1$, which agrees with that obtained by the mean value analysis method (6.7).

6.1.1.3 WTDMA/R with trespassing

The Markov chain model developed for WTDMA/R can now be extended for the WTDMA/R protocol with trespassing.

Firstly we need to alter the Markov chains to reflect the the changing probability of transmitting. Now the probability of transmitting a packet is altered to the probability to transmit to the allocated station plus the probability that a packet is not transmitted to the allocated station, but there is at least one packet awaiting transmission to another destination, multiplied by the trespass probability [Figure 6.3]:

$$\mu = \frac{1}{N-1} + \left(1 - \frac{1}{N-1}\right).\left(1 - P_0^{N-2}\right).Z \tag{6.13}$$

Substituting the new departure rate into (6.11), gives us the following non-linear equation:

$$\begin{aligned}
&\left(\left(1 - \frac{p}{N-1}\right).(N-2).Z - \frac{p}{N-1} + 1\right).P_0^{N-1} + \\
&+ \left(\left(\frac{p}{N-1} - 1\right).(N-2).Z - p + \frac{p}{N-1} - 1\right).P_0^{N-2} + \\
&+ \left(\frac{p}{N-1} - 1\right).(N-2).Z.P_0 + \\
&+ \left(1 - \frac{p}{N-1}\right).(N-2).Z = 0
\end{aligned} \tag{6.14}$$

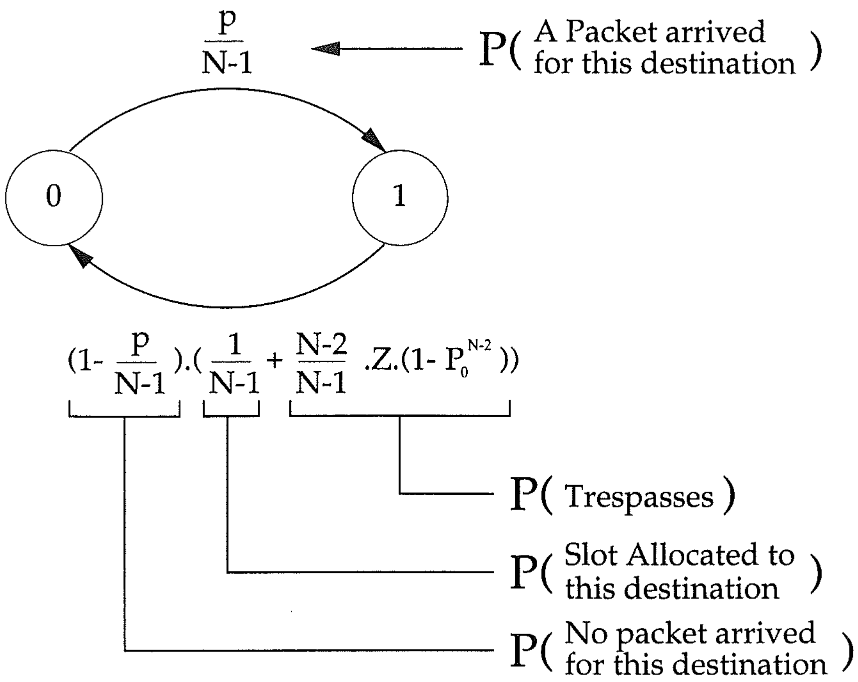

Figure 6.3: A two state Markov chain for the state of a destination buffer at a Station, using trespassing.

From this we can get an equation from the total traffic from a station to all $N-1$ destinations (in packets per time slot):

$$\text{Traffic} = (N-1).\left(\frac{1}{N-1} + \frac{N-2}{N-1}.\left(1 - P_0^{N-2}\right).Z\right).P_1 \quad (6.15)$$

Since, $(N-2)/(N-1).(1-P_0^{N-2}).Z \geq 0$, this shows that with trespassing the traffic generated will always be greater or equal to that of WTDMA. However this is not the throughput as the traffic includes collisions when more than one station transmits on a wavelength. We must therefore calculate the total successful traffic, or simply throughput. Note that the probability that a station transmits on its allocated wavelength in a time slot equals:

$$\begin{aligned} P(A=1) &= \frac{\bar{X}}{N-1} \\ &= P_1 \end{aligned} \quad (6.16)$$

Probability that a station does not transmit on its allocated wavelength in a time slot:

$$\begin{aligned} P(A=0) &= 1 - \frac{\bar{X}}{N-1} \\ &= P_0 \end{aligned} \quad (6.17)$$

Note that the probability a station trespasses is the probability that the station does not want to transmit on its allocated slot, multiplied by the probability that it has a packet to transmit on another wavelength, multiplied by the proportion of full buffers, multiplied by the trespass probability:

$$\pi = P_0.\left(1 - P_0^{N-2}\right).P_1.Z \tag{6.18}$$

Probability that only one station wants to trespass on an unallocated wavelength can be derived from the binomial distribution:

$$P(B=1) = (N-2).\pi.(1-\pi)^{N-3} \tag{6.19}$$

Probability that no stations want to trespass on an unallocated wavelength can be also derived from the binomial distribution:

$$P(B=0) = (1-\pi)^{N-2} \tag{6.20}$$

Probability that no-one transmits on a wavelength equals:

$$\begin{aligned} P(\text{None}) &= P(A=0 \cap B=0) \\ &= \left(1 - \frac{\bar{X}}{N-1}\right)(1-\pi)^{N-2} \end{aligned} \tag{6.21}$$

Probability that no-one trespasses and the allocated station transmits:

$$\begin{aligned} P(\text{Allocated}) &= P(A=1 \cap B=0) \\ &= \frac{\bar{X}}{N-1}(1-\pi)^{N-2} \end{aligned} \tag{6.22}$$

Probability that only one station trespasses in a time slot, and the allocated station does not transmit on its wavelength:

$$\begin{aligned} P(\text{Trespass}) &= P(A=0 \cap B=1) \\ &= \left(1 - \frac{\bar{X}}{N-1}\right)(N-2).\pi.(1-\pi)^{N-3} \end{aligned} \tag{6.23}$$

Probability of a successful transmission gives throughput generated by each station.

$$\begin{aligned} P(\text{Success}) &= P(\text{Allocated}) + P(\text{Trespass}) \\ &= (1-\pi)^{N-3}\left(\frac{\bar{X}}{N-1}(1-\pi) + \left(1 - \frac{\bar{X}}{N-1}\right)(N-2).\pi\right) \end{aligned} \tag{6.24}$$

Finally, probability of a packet colliding in a given slot can be obtained as:

$$P(A+B>1) = 1 - P(\text{None}) - P(\text{Allocated}) - P(\text{Trespass}) \tag{6.25}$$

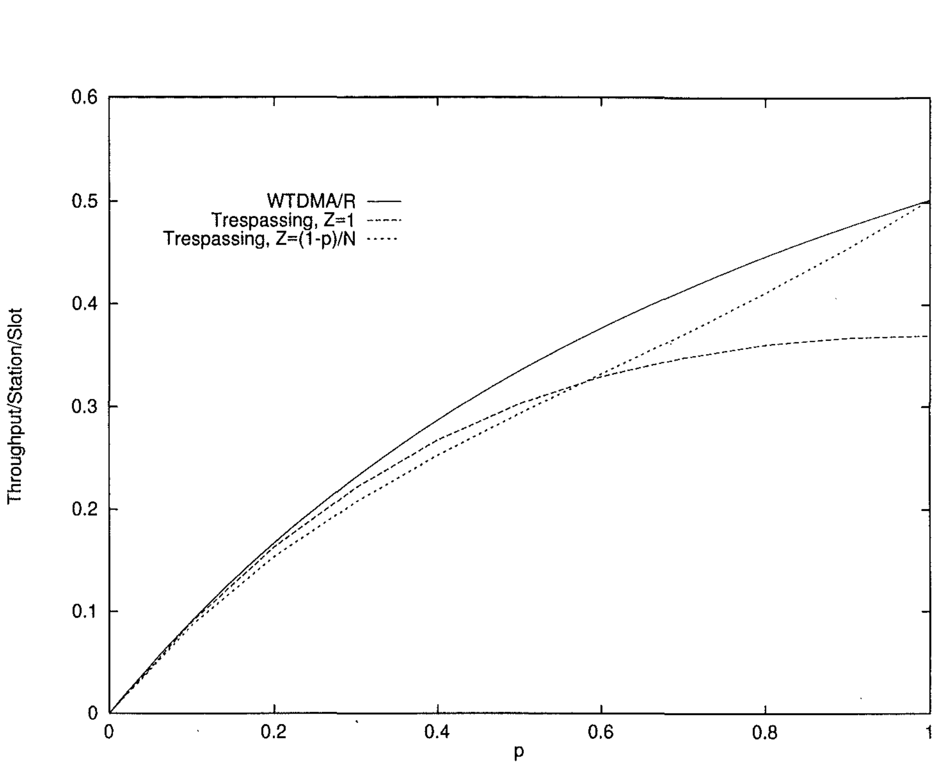

Figure 6.4: Analytical throughput for WTDMA/R, and WTDMA/R with trespassing, $Z = 1$ and $(1-p)/N$; unit buffers.

6.1.2 Numerical Results

The throughput for WTDMA/R and WTDMA/R with trespassing for, $Z = 1$ and $Z = (1-p)/N$ are shown in Figure 6.4. From the figure, trespassing can be seen to have a negative effect on throughput, compared to pure WTDMA/R, if no collision avoidance is implemented.

In Figure 6.5, the throughput per station per slot as a function of the total input load of a station is plotted for WTDMA/R and various trespassing rules. It is of note that the mathematical and simulated throughput match for WTDMA/R, giving a maximum throughput of about 50%. This confirms that the mathematical model is correct.

It can be seen that the deterministic rule of trespassing ($Z = 1$) is clearly the worst. This also matches the results found from mathematical analysis [Figure 6.4]. Note that in this case collisions can get up to 25% of throughput at maximum load. Trespassing with probability inversely proportional to the load and number of stations (i.e. $Z = (1-p)/N$), yields similar results to WTDMA/R, showing that any gain in throughput is undetectable with this scheme.

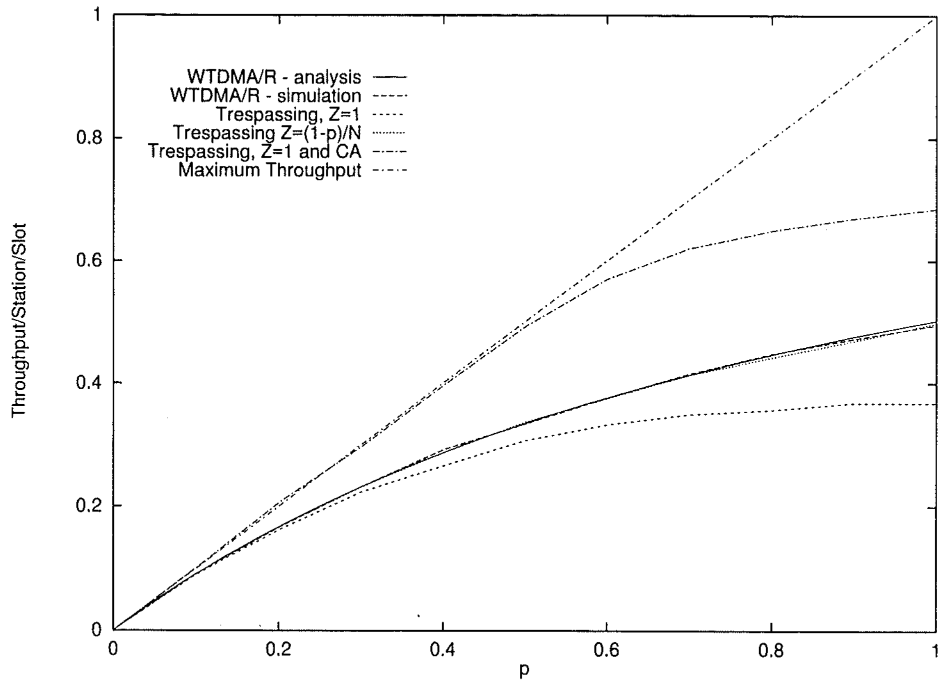

Figure 6.5: Throughput for WTDMA/R with unit buffer and trespassing rules: ($Z = 1$), ($Z = (1 - p)/N$) and ($Z = 1$ with CA).

WTDMA/R with trespassing, $Z = 1$ and CA yields an almost maximal throughput ($\rho \propto p$) for $p < 0.5$, while for $p > 0.5$ the gain is smaller. For $p = 1$, $\rho = 0.68$, which is a gain of almost 15 percentage points above the throughput of WTDMA/R, or an almost 38% gain.

In Figure 6.6, the mean packet delay for two cases is shown. WTDMA/R has a constant delay of around 100 slots. This can be significantly reduced by introducing trespassing, $Z = 1$ and collision avoidance. In the considered case, WTDMA/R with $Z = 1$ and collision avoidance has an almost constant delay of less than five slots for $p < 0.5$. Note that a station can almost always successfully trespass immediately if it does not use its allocated wavelength, meaning the delay is never much more than one slot. When the load gets higher, for $p > 0.5$, it rises to a maximum delay of 42 slots, which is 58% less delay than WTDMA/R.

The throughput given by the analytical model and simulation are compared, in Figure 6.7. This confirms that for WTDMA/R and WTDMA/R with trespassing, $Z = 1$, the mathematical and simulation model match. This is also true for WTDMA/R and WTDMA/R with trespassing, $Z = 1$ if the rate of

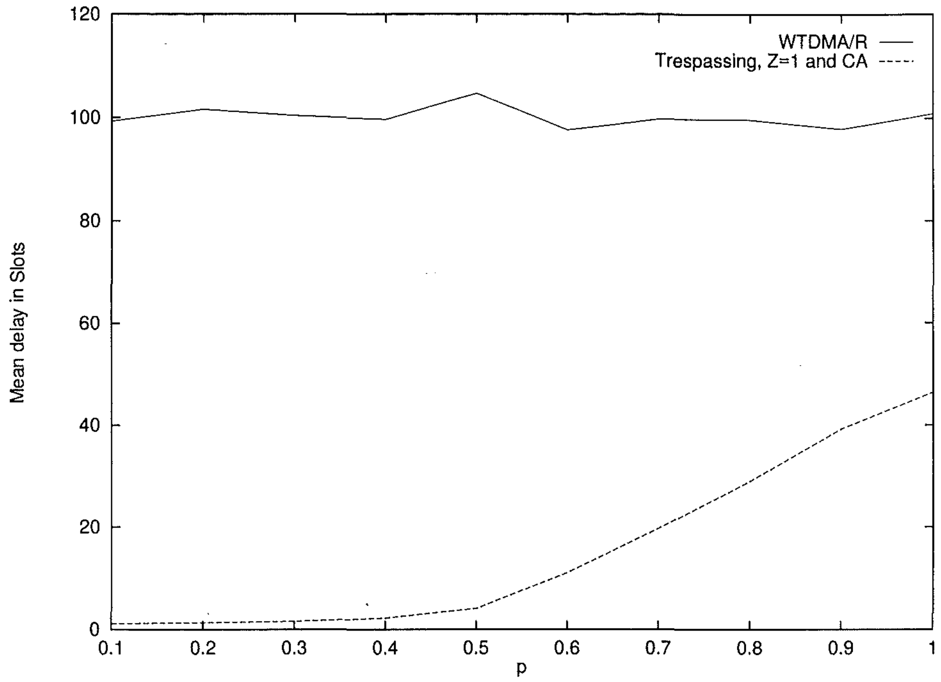

Figure 6.6: Mean delay for WTDMA/R with unit buffers and trespassing rules: $(Z = 1)$, $(Z = (1 - p)/N)$ and $(Z = 1$ with CA$)$, from simulation.

collisions is compared [Figure 6.7]. There is a small difference in results for WTDMA/R with trespassing, $Z = (1 - p)/N$.

If the traffic generated by a network is compared for the mathematical and simulation models, then the results can be seen in Figure 6.8. Here the results for the traffic generated from simulation are estimated as the throughput plus twice the number of collisions in a time slot (i.e., for every collision, there are two packets generating traffic). This slightly underestimates reality, as it is possible (but improbable) that three or more packets are involved in a collision. This explains why, as the offered load increases, the estimated traffic generated from simulation increasingly underestimates the theoretical results, as more instances of more that two packets colliding occur.

6.2 WTDMA/R, Infinite buffer

In this section WTDMA/R with infinite destination buffers is considered. Its performance is compared with WTDMA/R with trespassing, $Z = 1$ and $(1 - p)/N$ and WTDMA/R with trespassing, $Z = 1$ and collision avoidance.

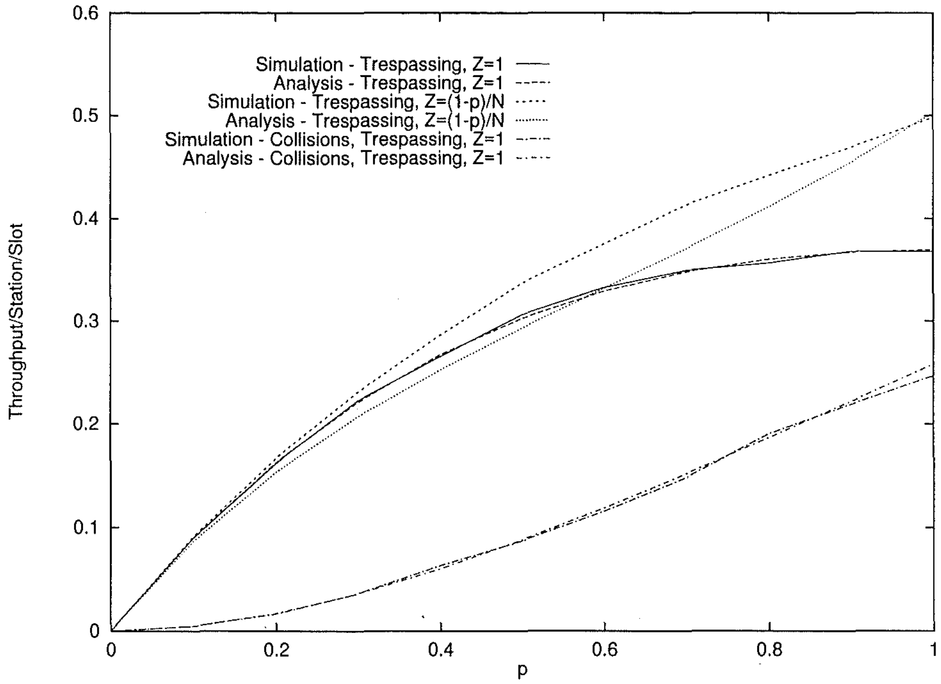

Figure 6.7: Comparison of throughput for WTDMA/R with trespassing, $Z = 1$ and $(1-p)/N$, unit buffers, for simulation and analysis.

6.2.1 Mathematical Analysis

Assuming unlimited buffering capacity of stations, assumptions A2 is replaced with A2$'$, allowing stations to have $N-1$ infinite buffers. This leads to a slightly altered model. Namely, now, contrary to the unit buffer case, packets are no longer dropped. The states of each of the $N-1$ queues at a station, at the beginning of the ith time slot, evolve following the equation:

$$x_i = x_{i-1} - t_{i-1} + a_{i-1} \tag{6.26}$$

Thus, on average:

$$\bar{X}_i = \bar{X}_{i-1} - \frac{\bar{X}_{i-1}}{N-1} + p \tag{6.27}$$

Since:

$$\frac{\bar{X}}{N-1} = p \tag{6.28}$$

Then in steady-state:

$$\bar{X} = p(N-1) \tag{6.29}$$

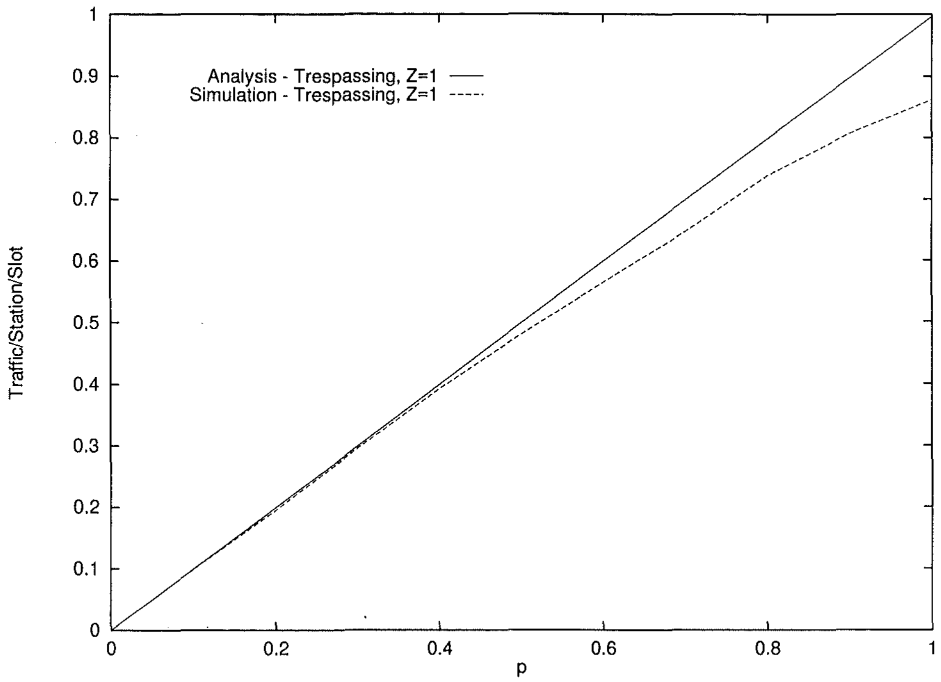

Figure 6.8: Comparison of traffic for WTDMA/R with trespassing, $Z = 1$, unit buffers, for simulation and analysis.

Hence the throughput of WTDMA/R with infinite buffers is p packets per station per slot.

6.2.2 Numerical Results

With infinite buffers, WTDMA/R can achieve almost maximum throughput [Figure 6.9], until $p > 0.7$, and thus until this point there is no performance gain in using trespassing. There is only a small throughput gain for $p > 0.7$, for trespassing, using collision avoidance. If trespass is used with $Z = 1$, but without collision avoidance, then there is an obvious large loss in throughput.

Simulation was used to determine the mean packet delay of the two non-loss systems (WTDMA/R and WTDMA/R with trespassing, $Z = 1$ and CA). The results are shown in Figure 6.10. Simulations were generally done for $0.1 \leq p \leq 0.9$, as the systems for higher p would require extremely long simulation runs (some results were obtained for $p = 1$). WTDMA/R has a delay quickly increasing from around 100 slots at $p = 0.1$ to over 400 slots for $p = 0.9$. WTDMA/R, with trespassing, $Z = 1$ and CA again yields an almost constant delay of less than six slots, for $p < 0.5$. Even with infinite buffers, the number of

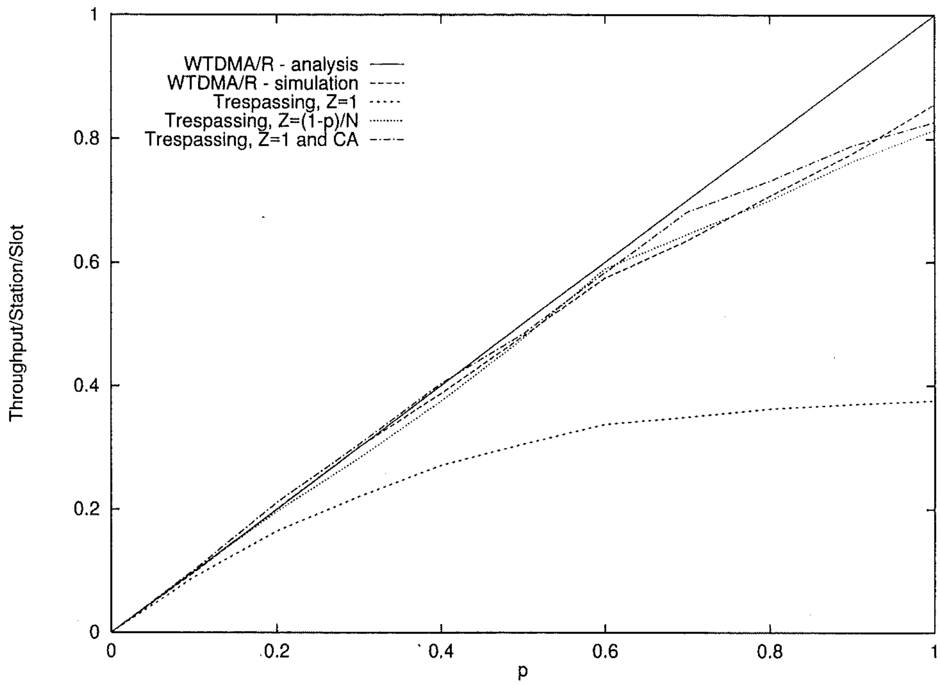

Figure 6.9: Throughput for WTDMA/R with infinite buffers and trespassing rules: ($Z = 1$), ($Z = (1-p)/N$) and ($Z = 1$ with CA), from simulation.

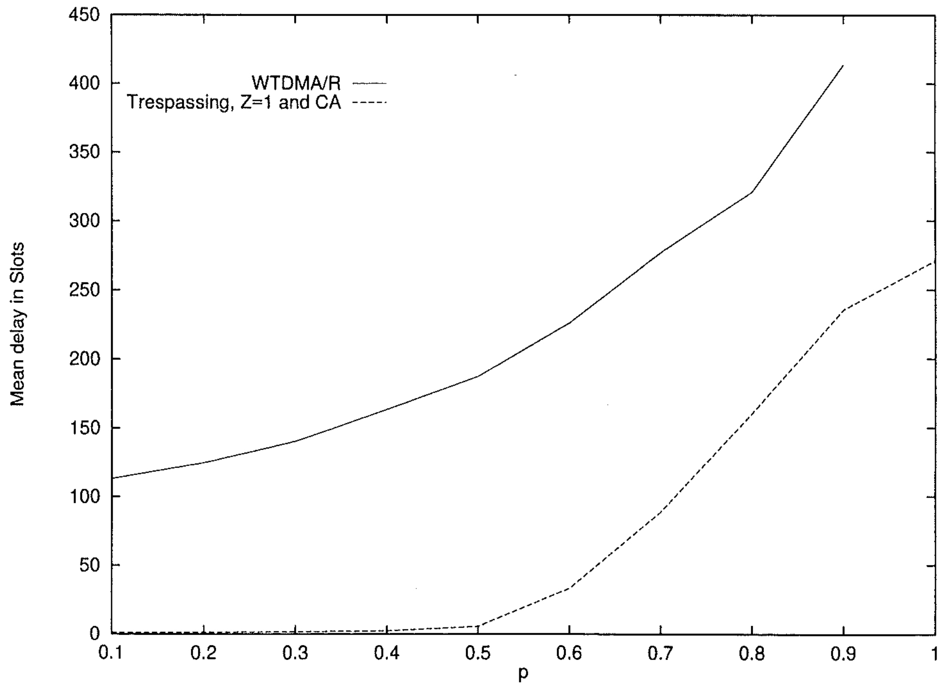

Figure 6.10: Mean delay for WTDMA/R with infinite buffers, with and without trespassing $Z = 1$ with CA, from simulation.

packets waiting in each buffer is very low, so a station can almost always trespass successfully immediately if it does not use its allocated wavelength. When the load gets higher, as the buffers can now have more than one packet queued, the mean delay increases much faster than in the corresponding unit buffer cases. For $p > 0.5$, the mean packet delay in WTDMA/R, with trespassing and collision avoidance rises to a maximum of 236 slots at $p = 0.9$. This is still 43% less than for pure WTDMA/R.

6.3 WTDMA/D, Unit buffer

In this section WTDMA/D with unit destination buffers is considered. Its performance is compared with WTDMA/D with trespassing, $Z = 1$ and $(1 - p)/N$ and with WTDMA/D with trespassing, $Z = 1$, and collision avoidance. It is also compared with the WTDMA/R cases studied in Section 6.1 and with a random-access protocol with collision avoidance.

6.3.1 Mathematical Analysis

For WTDMA/D, the throughput can be calculated simply as the probability that the buffer was full just before the beginning of the allocated slot. Since the probability of it being empty for the $N-1$ slots is the probability that nothing had arrived, the probability of it being full is:

$$P_1 = 1 - \left(1 - \frac{p}{N-1}\right)^{N-1} \tag{6.30}$$

As $N \to \infty$, $(1 - \frac{p}{N-1})^{N-1} \to e^{-p}$, meaning that at the maximum load the throughput is about 0.63.

Using $N-1$ independent Markov chains to model each destination buffer in one station (in Figure 6.11 station N is assumed), performance of the network during a TDMA cycle comprising of $N-1$ slots can be studied. To analyse traffic generated by a station, only one Markov chain needs to be analysed [Figure 6.12].

In a uniform traffic model, every $(N-1)$st slot can be used to send a packet on the allocated wavelength. This means that a packet has to wait on average $(N-1)/2$ slots before being served.

If we assume that the state a buffer is in is measured just after transmission, but before a new arrival is accepted [Figure 6.13], then a buffer of capacity one is guaranteed to be empty on its TDMA slot ($k = 0$). If the buffer was full in

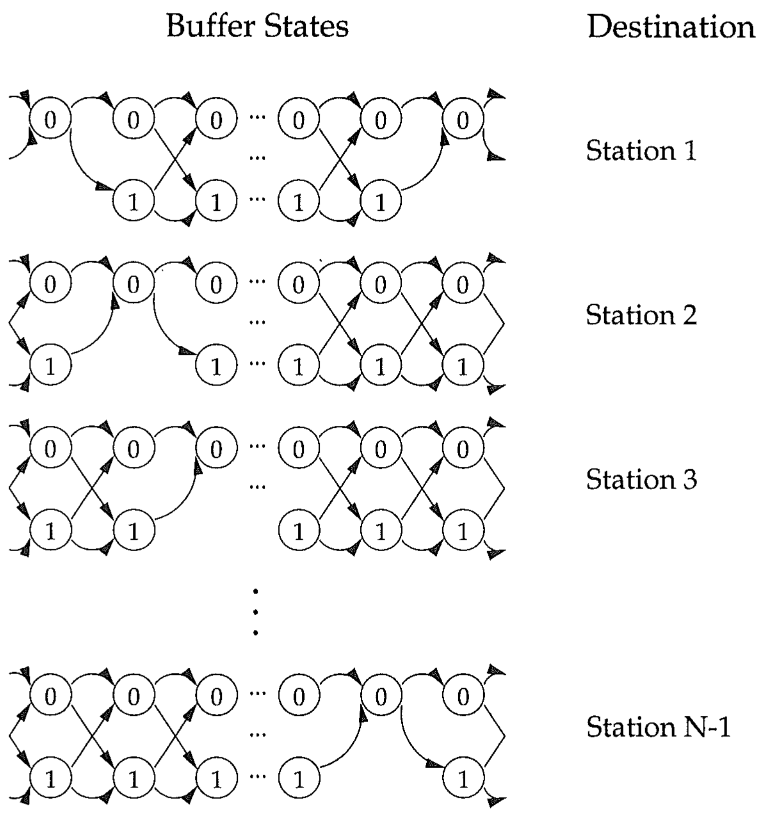

Figure 6.11: $(N-1)$ Independent Markov Chains as a model of $(N-1)$ buffers at Station N in a network operated under WTDMA/D with unit buffers.

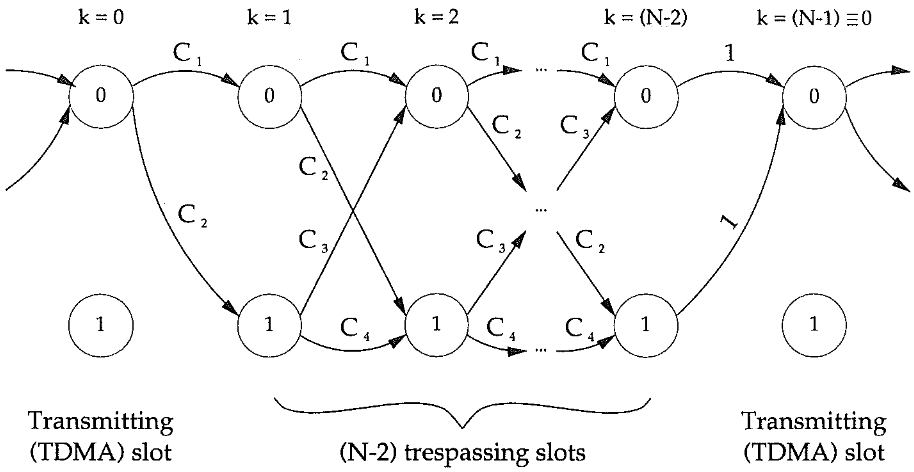

Figure 6.12: The state diagram of a destination buffer at a station over a cycle of $N-1$ slots of a WTDMA/D network.

C_1	$(A=0)\cup(A=1,\text{Trespass}=1)$
C_2	$A=1,\text{Trespass}=0$
C_3	$\text{Trespass}=1$
C_4	$\text{Trespass}=0$

Table 6.1: Conditions for transition of states for Markov chain in Figure 6.12.

the previous slot, then the packet will be transmitted (if a new packet arrived when the buffer was full, it would have been dropped). If the buffer was empty in the previous slot, and no packet has arrived, the buffer will remain empty, and if a packet has arrived, it will be transmitted.

For the next $N-2$ ($k=1,2,...,N-2$) slots, there would be no transmission from that buffer, following TDMA rules. Instead, if trespassing is allowed, then the slots become "trespassing slots". There are four possible transitions between the states, during this stage [Table 6.1].

If the buffer is empty, then it will stay empty if either there is no arrival, or if there is an arrival and the new packet is immediately transmitted as a trespassing packet (C_1, Table 6.1). A full buffer will become empty if its content is transmitted during a trespassing slot (C_3, Table 6.1). Thus:

$$P_0(k+1) = \left(\left(1-\frac{p}{N-1}\right)+\frac{p}{N-1}.\frac{Z}{N-2}\right).P_0(k)+\frac{Z}{N-2}.P_1(k) \tag{6.31}$$

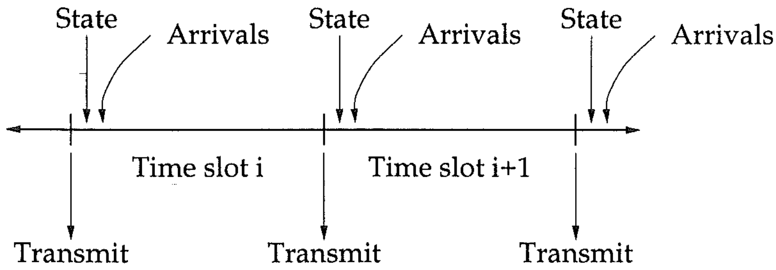

Figure 6.13: Scheduling of events assumed in the Markov chain in Figure 6.12.

If the buffer is empty, then it will become full if there is an arrival, and the new packet is not transmitted as a trespassing packet (C_2, Table 6.1). A full buffer will become empty if its content is transmitted as a trespassing packet (C_3, Table 6.1). Lastly, the buffer will remain full if no trespassing occurs (C_4, Table 6.1).

Thus:

$$P_1(k+1) = \frac{p}{N-1}.\frac{Z}{N-2}.P_0(k) + \left(1 - \frac{Z}{N-2}\right).P_1(k) \tag{6.32}$$

In vector matrix form:

$$\begin{bmatrix} P_0(k+1) \\ P_1(k+1) \end{bmatrix} = \begin{bmatrix} \left(1 - \frac{p}{N-1}\right) + \frac{p}{N-1}.\frac{Z}{N-2} & \frac{Z}{N-2} \\ \frac{p}{N-1}.\frac{Z}{N-2} & \left(1 - \frac{Z}{N-2}\right) \end{bmatrix} \begin{bmatrix} P_0(k) \\ P_1(k) \end{bmatrix} \tag{6.33}$$

To simplify notation, let:

$$\alpha = \frac{p}{N-1}.\left(1 - \frac{Z}{N-2}\right) \tag{6.34}$$

$$\beta = \frac{Z}{N-2} \tag{6.35}$$

$$\Phi = \begin{bmatrix} 1-\alpha & \beta \\ \alpha & 1-\beta \end{bmatrix} \tag{6.36}$$

Since the buffer is always empty at the start of a TDMA cycle, the initial conditions are:

$$\begin{bmatrix} P_0(0) \\ P_1(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{6.37}$$

Thus the solution to (6.33) is:

$$\begin{bmatrix} P_0(k) \\ P_1(k) \end{bmatrix} = \Phi^k \begin{bmatrix} P_0(0) \\ P_1(0) \end{bmatrix} \tag{6.38}$$

The eigenvalues (Anton, 1991) of Φ are 1 and $1-\alpha-\beta$.

The eigenvectors of Φ are $\begin{bmatrix} \beta \\ \alpha \end{bmatrix}$ and $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$ (Anton, 1991). Thus:

$$\begin{aligned} \Phi^k &= \begin{bmatrix} \beta & -1 \\ \alpha & 1 \end{bmatrix} \begin{bmatrix} 1^k & 0 \\ 0 & (1-\alpha-\beta)^k \end{bmatrix} \begin{bmatrix} \beta & -1 \\ \alpha & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \beta & -1 \\ \alpha & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & (1-\alpha-\beta)^k \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -\alpha & \beta \end{bmatrix} \div (\alpha+\beta) \end{aligned} \tag{6.39}$$

$$\begin{bmatrix} P_0(0) \\ P_1(0) \end{bmatrix} = \begin{bmatrix} \beta & -1 \\ \alpha & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & (1-\alpha-\beta)^k \end{bmatrix} \begin{bmatrix} \frac{1}{\alpha+\beta} \\ \frac{-\alpha}{\alpha+\beta} \end{bmatrix} \tag{6.40}$$

This gives us:

$$P_1(k) = \frac{\alpha}{\alpha+\beta}\left(1-(1-\alpha-\beta)^k\right) \tag{6.41}$$

If $k \to \infty$ (a very long TDMA cycle), we get:

$$\begin{aligned} \begin{bmatrix} P_0 \\ P_1 \end{bmatrix} &= \Phi \begin{bmatrix} P_0 \\ P_1 \end{bmatrix} \\ &= \begin{bmatrix} \frac{\beta}{\alpha+\beta} \\ \frac{\alpha}{\alpha+\beta} \end{bmatrix} \end{aligned} \tag{6.42}$$

This result matches that of (6.41), as when $k \to \infty$, $(1-\alpha-\beta)^k \to 0$. Also if $\beta = 0$ (i.e., no trespassing), then $P_1(k) = 1$.

The probability that a packet is transmitted during a TDMA slot (i.e. for $k = 0 \equiv (N-1)$) equals the probability that there was a packet in the buffer in the previous slot, or that the buffer was empty and a new packet arrived:

$$P(\text{Transmission at slot N-1}) = P_1(N-2) + P_0(N-2).\frac{p}{N-1} \tag{6.43}$$

A packet is transmitted during trespassing slot k ($k = 1, 2, ..., N-2$), if there was a packet in the buffer in the previous slot, and that packet was chosen for trespassing:

$$P(\text{Transmission at slot k}) = P_1(k-1).\frac{Z}{N-2} \tag{6.44}$$

Thus, the traffic, or the mean number of packets transmitted per slot per wavelength is:

$$
\begin{aligned}
ET_{trans} &= \frac{1}{N-1}\left(\sum_{k=1}^{N-2} P_1(k-1).\frac{Z}{N-2} + P_1(N-2).1 + P_0(N-2).\frac{p}{N-1}.1\right) \\
&= \frac{1}{N-1}.\frac{Z}{N-1}.\frac{\alpha}{\alpha+\beta}\left(N+2-\frac{(1-\alpha-\beta)^{N-2}}{\alpha+\beta}\right)+ \\
&\quad + \frac{1}{N-1}.\frac{\alpha}{\alpha+\beta}\left(1-(1-\alpha-\beta)^{N-2}\right)+ \\
&\quad + \frac{1}{N-1}.\frac{p}{N-1}\left(1-\frac{\alpha}{\alpha+\beta}\left(1-(1-\alpha-\beta)^{N-2}\right)\right) \qquad (6.45)
\end{aligned}
$$

For WTDMA/D without trespassing, $\alpha = p/(N-1)$ and $\beta = 0$, so (6.45) simplifies to:

$$
\begin{aligned}
ET_{trans} &= \frac{1}{N-1}.1.\left(1-\left(1-\frac{p}{N-1}\right)^{N-2}\right)+ \\
&\quad + \frac{1}{N-1}.\frac{p}{N-1}\left(1-1.\left(1-\left(1-\frac{p}{N-1}\right)^{N-2}\right)\right) \\
&= \frac{1}{N-1}\left(1-\left(1-\frac{p}{N-1}\right)^{N-1}\right) \qquad (6.46)
\end{aligned}
$$

The throughput for WTDMA/D is $(N-1) times larger that that given by$ (6.46), and so matches the result found in (6.30).

6.3.2 Numerical Results

In Figure 6.14, the throughput per station per slot is plotted for the various cases. Note that for WTDMA/D the mathematical and simulated results match, giving a maximum throughput of about 63%. This confirms the correctness of the mathematical model.

Deterministic trespassing ($Z = 1$) is clearly worse, as it was with WTDMA/R. WTDMA/D with trespassing with $Z = (1-p)/N$,produces similar results to WTDMA/D.

WTDMA/D with trespassing, $Z = 1$ and CA yields an almost maximal throughput ($\rho \propto p$) if $p < 0.5$. For $p > 0.5$, the throughput is noticeably higher than for WTDMA/D, reaching $\rho = 0.72$ at $p = 1$.

WTDMA/D performs better than WTDMA/R [Figure 6.15]. However, for deterministic trespassing ($Z = 1$), there is no difference whether a random or deterministic slot allocation is used. The performance of networks with collision avoidance is similar, with WTDMA/D performing slightly better at $p > 0.6$.

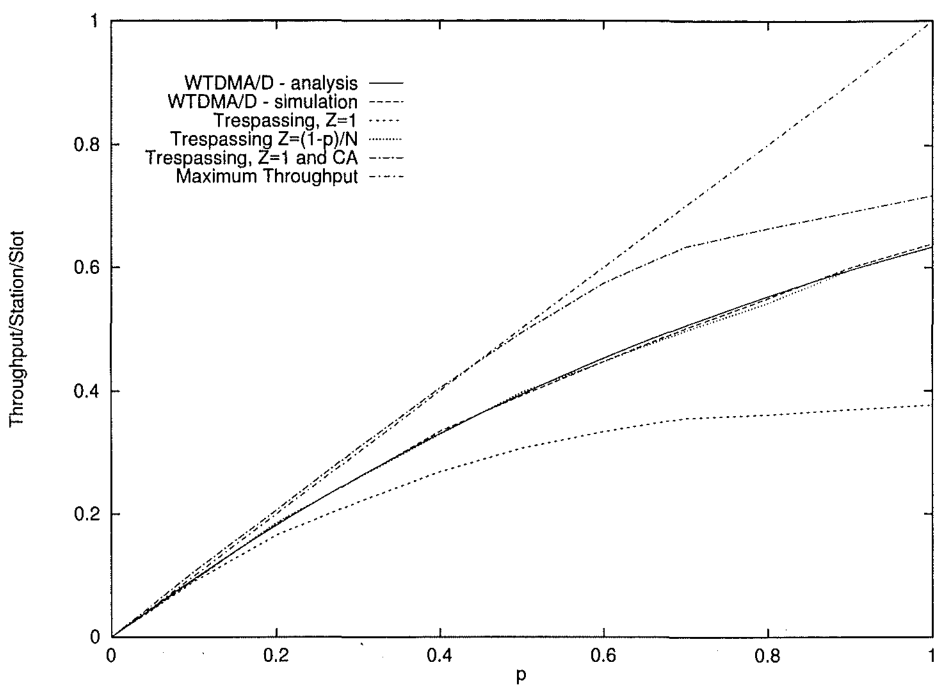

Figure 6.14: Throughput for WTDMA/D with unit buffers and trespassing rules: ($Z = 1$), ($Z = (1-p)/N$) and ($Z = 1$ with CA).

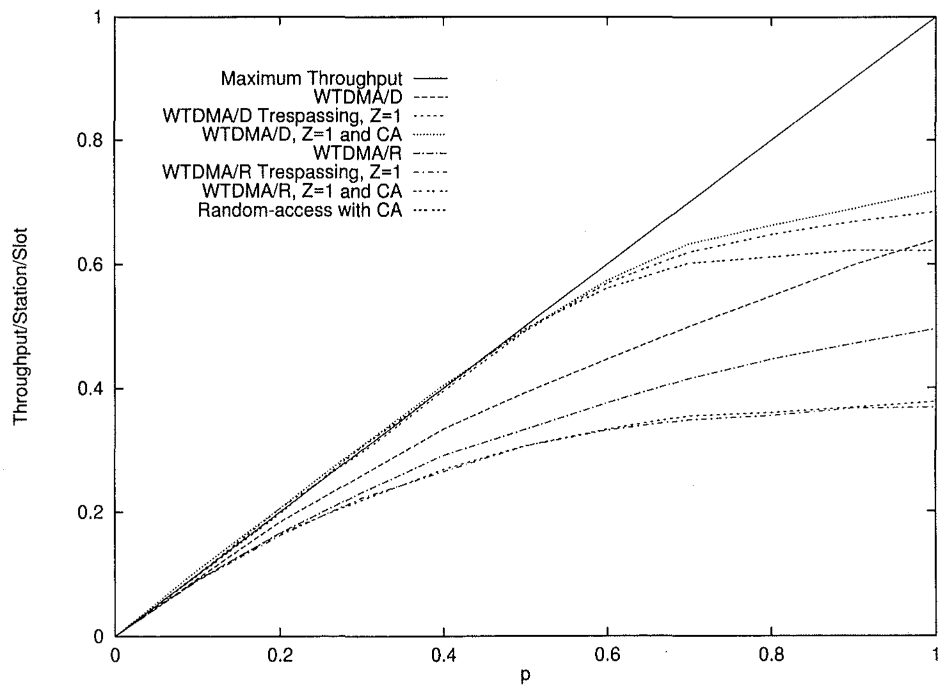

Figure 6.15: Throughput for WTDMA/D and WTDMA/R, for their different trespassing rules: ($Z = 1$), ($Z = (1-p)/N$) and ($Z = 1$ with CA) and a random-access/CA protocol. In all cases unit buffers were considered.

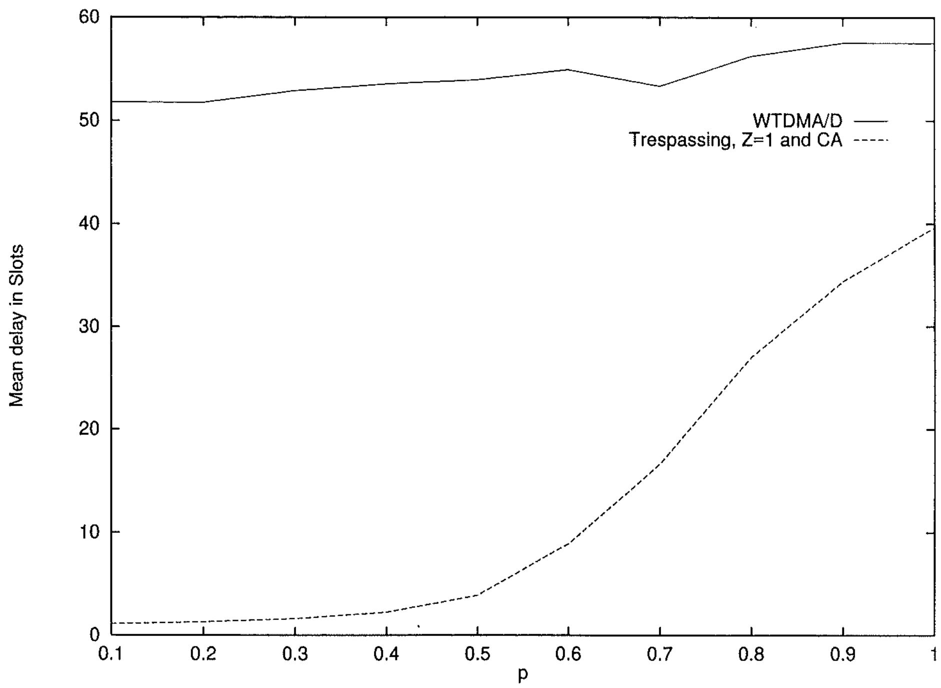

Figure 6.16: Mean packet delay for WTDMA/D with unit buffers, with and without trespassing, $Z = 1$ with CA), from simulation.

6.3.2.1 Random-access protocol

Also included in Figure 6.15 is the throughput gained by a random-access protocol, using collision avoidance. Here a packet is chosen at random from one of the full buffers and transmitted. If there are any collisions, then one of the packets is forwarded to the destination, and all other colliding packets are reflected. This protocol could be considered a purely trespassing protocol, without any prior slot allocation.

As can be seen from Figure 6.15, this scheme works as well as the trespassing with collision avoidance schemes for $p < 0.6$. The throughput then levels off to a value of just over $\rho = 0.62$.

This result is of interest, as it shows that is possible to get a higher throughput using a hybrid of TDMA and random-access, rather than using either pure random-access, or pure TDMA, i.e. hybridizing the two protocols has a synergistic effect.

When analysing mean packet delay [Figure 6.16], it can be seen that WTDMA/D has the mean packet delay of between 50 and 60 slots. This is about half the length of a frame in slots, and so is the expected result. With WTDMA/D with

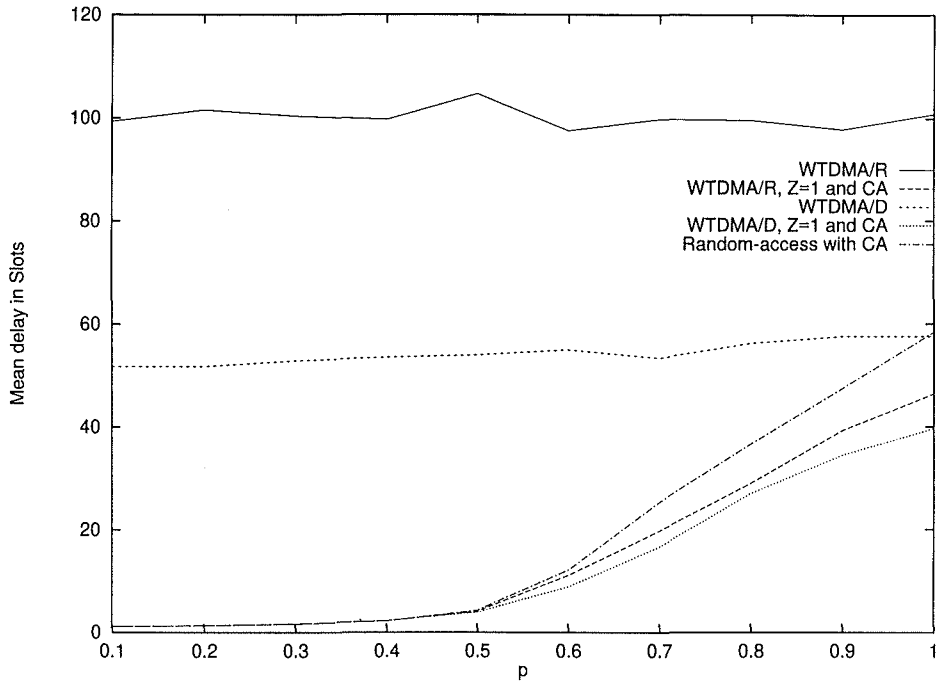

Figure 6.17: Mean packet delay for WTDMA/D and WTDMA/R, with and without trespassing, $Z = 1$ with CA, and a random-access/CA protocol, all cases assuming unit buffers.

trespassing, $Z = 1$ and collision avoidance, the mean delay is almost constant, below four slots for $p < 0.5$. For $p > 0.5$, it rises constantly to a maximum of around 40 slots delay, which is about 69% of the mean delay under WTDMA/D without collision avoidance.

WTDMA/D has half the mean packet delay of WTDMA/R [Figure 6.17]. This is because random slot assignment causes (on average) a packet to be served once every $N - 1$ slots. However, if the assignment is done deterministically, then the wait is only half of that. With collision avoidance, WTDMA/D has only a slightly smaller mean delay than WTDMA/R, since the effects of trespassing on mean delay much outweigh that of the WTDMA component.

Additionally, the mean packet delay of the random-access protocol with collision avoidance is shown in Figure 6.17. At low loads ($p < 0.5$), its performance is inseparable to that of WTDMA with trespassing. As the load increases to $p > 0.5$, the mean delay linearly converges to the mean delay of WTDMA/D without trespassing. This is worse than both WTDMA/R and WTDMA/D with trespassing and CA protocols.

The estimation of traffic generated by a network by the mathematical and

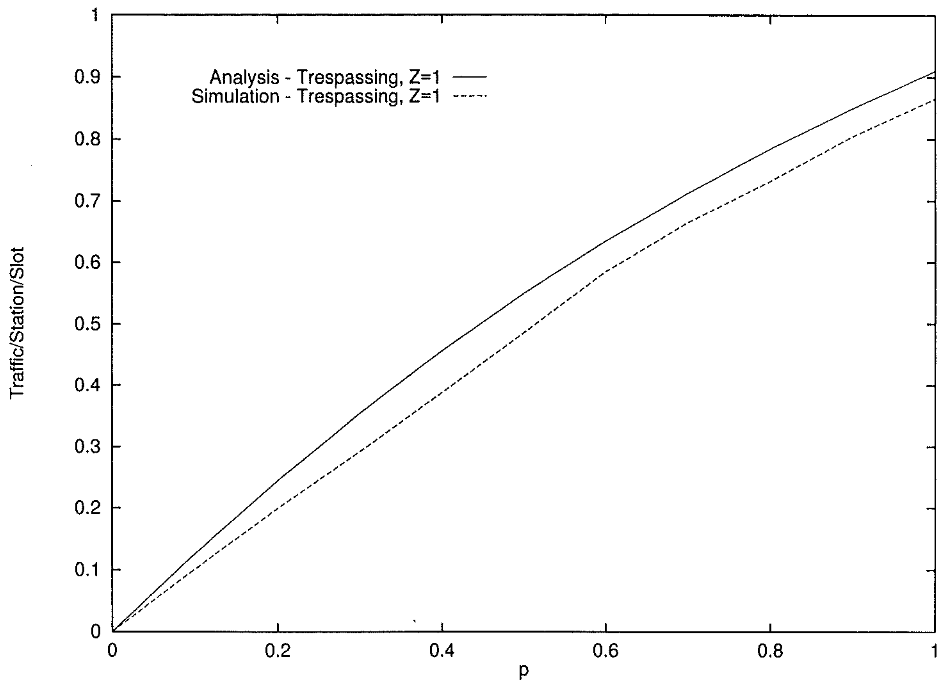

Figure 6.18: Comparison of traffic for WTDMA/D with trespassing, $Z = 1$, unit buffers, for simulation and analysis.

simulation models, is shown in Figure 6.18. From the figure, it can be seen that the analytical and simulation models basically agree. Some refinement of the analytical model (especially to include throughput analysis) should be done to.

6.4 WTDMA/D, Infinite buffer

In this section WTDMA/D, with infinite destination buffers is considered. Its performance is compared with WTDMA/D with trespassing, $Z = 1$ and $(1 - p)/N$ and WTDMA/D with trespassing, $Z = 1$, and collision avoidance. It is also compared with the WTDMA/R cases, considered in Section 6.2.

6.4.1 Numerical Results

WTDMA/D (and WTDMA/R) with infinite buffers, can achieve almost maximum throughput [Figure 6.19], so there is no performance gain in throughput from using trespassing. If trespassing is used deterministically ($Z = 1$) and without collision avoidance, then there is an obvious large loss in throughput.

Figure 6.20 shows the mean packet delay for WTDMA/D. Mean packet

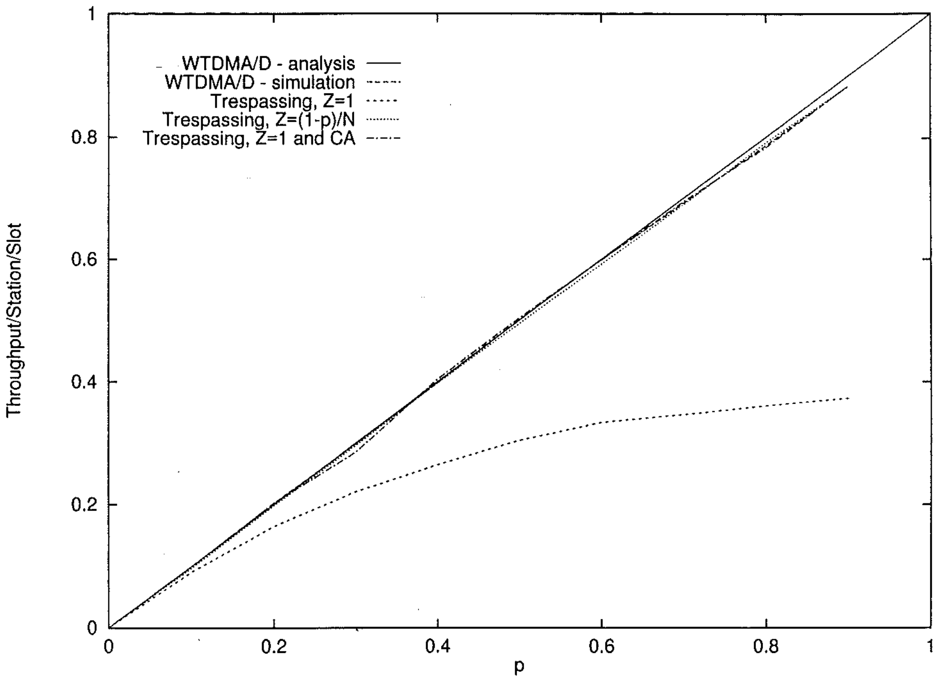

Figure 6.19: Throughput for WTDMA/D, WTDMA/R and their equivalent trespassing rules: $(Z = 1)$, $(Z = (1 - p)/N)$ and $(Z = 1$ with CA) with infinite buffers.

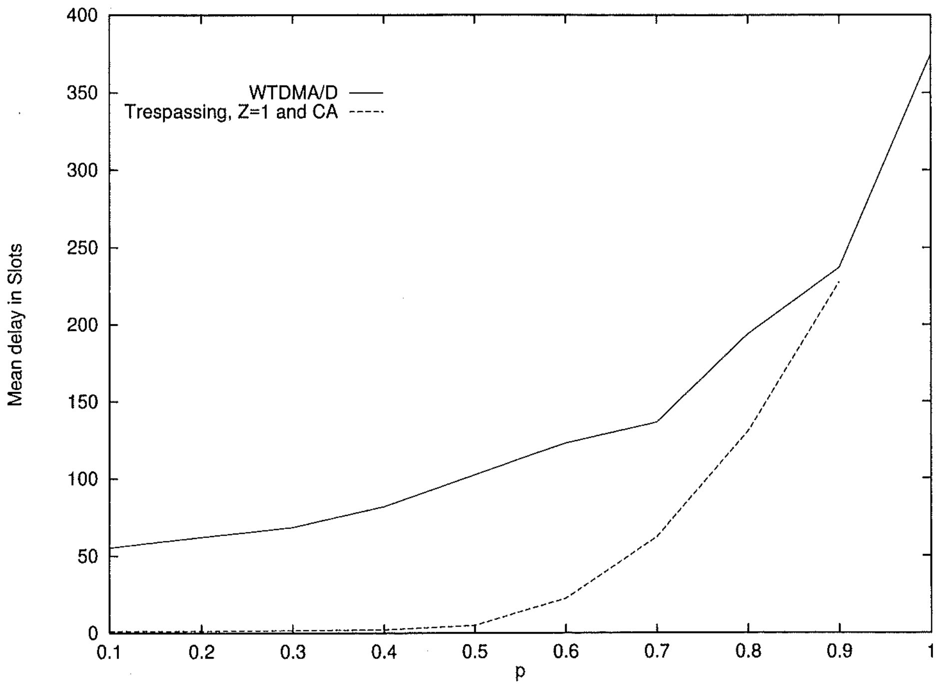

Figure 6.20: Mean packet delay for WTDMA/D with and without trespassing, $Z = 1$ with CA, infinite buffers, from simulation.

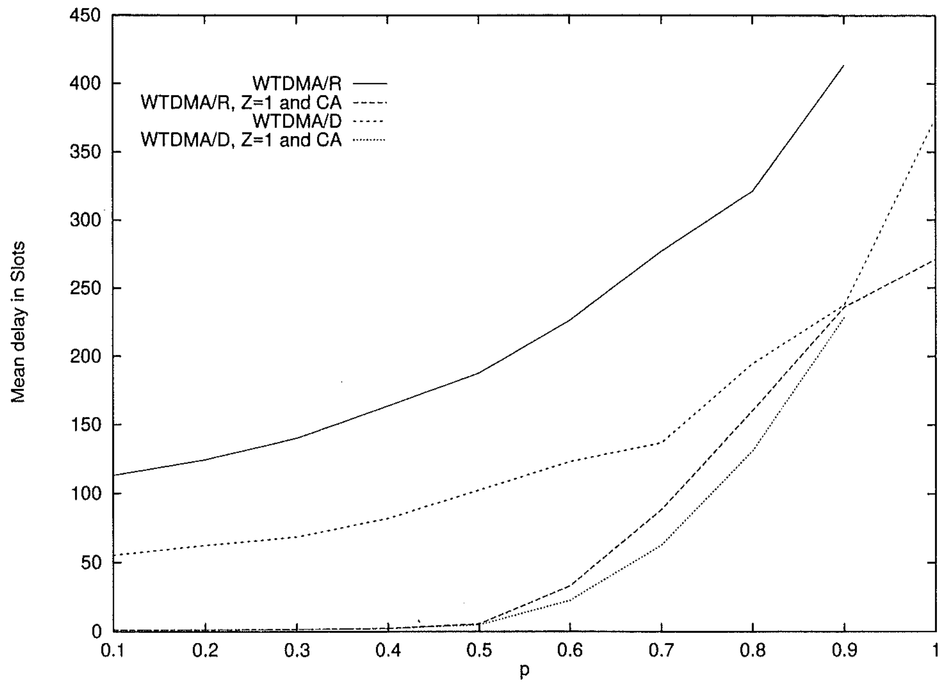

Figure 6.21: Mean packet delay for WTDMA/D and WTDMA/R with and without trespassing, $Z = 1$ with CA, infinite buffers, from simulation.

delay quickly increases from around 50 slots at $p = 0.1$ to 375 slots for $p = 1$. For $p < 0.5$ WTDMA/D with trespassing, $Z = 1$ and collision avoidance again gives an almost constant mean delay of less than five slots. For $p > 0.5$, it rises to a 228 slot mean delay at $p = 0.9$. At $p = 0.9$, the mean delay is four percent less than in WTDMA/D. As the load approaches maximum, the mean delay characteristics converge.

6.5 No out-of-order Packets

In this section WTDMA/D with trespassing, $Z = 1$ and collision avoidance with unit destination buffers and a non-zero propagation delay is considered. Its performance is compared with that of WTDMA/D. The propagation delays considered are the equivalent to 25 and 40 slots for the 100 station network.

6.5.1 Numerical Results

The cases previously considered do not take into account the effect propagation delay has on throughput and mean delay, when collision avoidance is

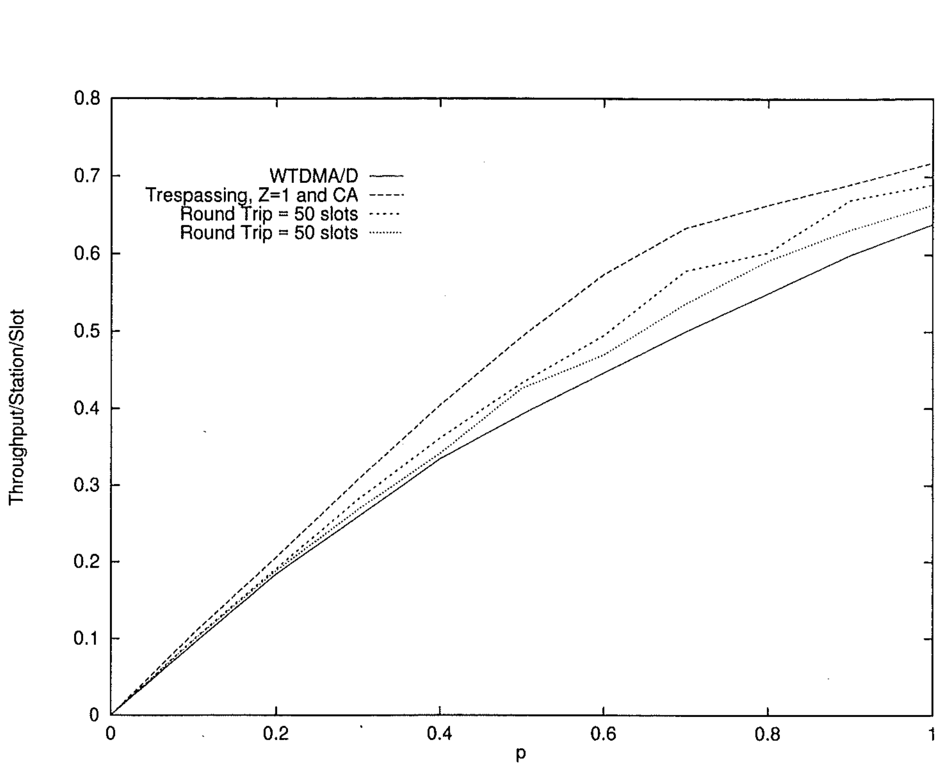

Figure 6.22: Throughput for WTDMA/D with unit buffers and trespassing, $Z = 1$ and CA, for round-trip delays of 50 and 80 slots, with protection against out-of-order transmission.

implemented. If a packet collides, then it is immediately reflected back to the sender, and put back into the appropriate destination buffer, without suffering the full round trip delay between source and destination station.

If the propagation delay was taken into account, then a packet that collides would take twice the propagation delay to return to its sender. This would have an obvious effect on mean packet delay.

To avoid out-of-order transmission of packets, each a packet must not be transmitted as a trespassing packet if another packet on the same wavelength could be delivered to its destination before an earlier trespassing packet. Thus a packet can not be transmitted as a trespassing packet if it is within twice the propagation delay of its allocated slot. A packet also can not be transmitted as a trespassing packet if a trespassing packet on the same wavelength has been transmitted during the time corresponding to two propagation delays. This will have an effect on the number of packets available for trespassing, lowering the throughput gain in trespassing.

Figure 6.22 shows the throughput achieved with a propagation delay of 25 and 40 slots. This means that a packet can not be transmitted if it is within 50 (or 80) slots from its allocated slot. Given a frame in this case is 99 slots long,

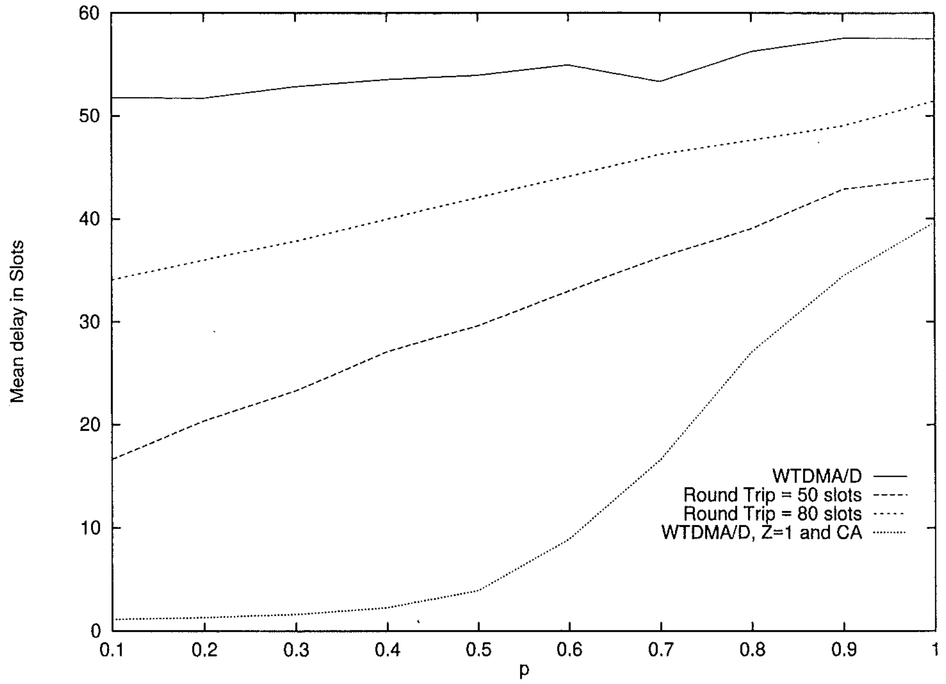

Figure 6.23: Mean packet delay for WTDMA/D with unit buffers and trespassing, $Z = 1$ and CA, for round-trip delays of 50 and 80 slots, with protection against out-of-order packets.

this means that a packet does not have the opportunity to trespass for 51%, or 81% respectively, of the destinations in any slot. A station can also never trespass more than once in a TDMA cycle, as the round-trip delay is more than half the number of slots in a cycle, in both cases.

A propagation delay of 25 slots over halves the throughput gained in using WTDMA/D with trespassing and CA, over that of WTDMA/D [Figure 6.22], e.g. at $p = 0.6$, WTDMA/D has a throughput of 0.45, while trespassing and CA, without a propagation delay has a 28% higher throughput of 0.57. If a propagation of 25 slots is taken into account, then the throughput is only 0.5, a gain of only 11%. A propagation delay of 40 slots cuts the gain in throughput by about a sixth (a 5% gain).

There is a quite noticeable effect on mean packet delay as soon as the propagation delay is factored into calculations [Figure 6.23]. The mean delay for WTDMA/D with trespassing and CA at $p = 0.1$ is 6.72% of that with propagation delay of 25, and 3.29% of that with propagation delay of 40. This is still favourable compared to the 2.16% of the mean delay of WTDMA/D experienced by WTDMA/D with trespass and CA.

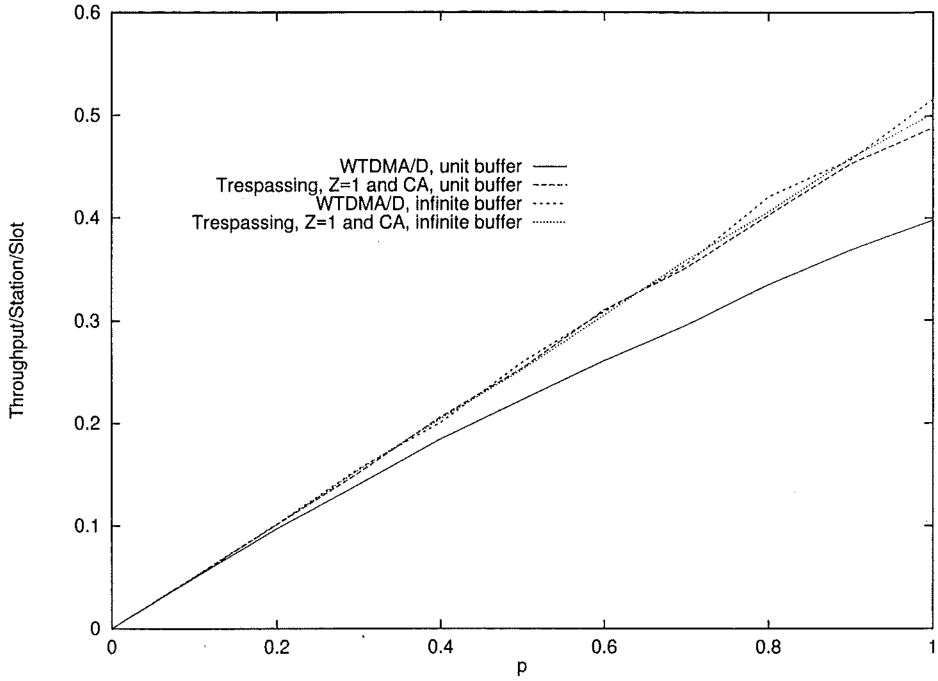

Figure 6.24: Throughput to non-server stations for WTDMA/D with trespassing, $Z = 1$ and CA, with unit buffers, for the Server model.

For $p > 0.5$, the differences become smaller, and when $p = 1$, WTDMA/D with trespassing and CA has 90.26% of the mean delay caused by a propagation delay of 25, and 77.03% of the delay caused with propagation delay of 40. This compares with WTDMA/D with trespassing and CA having 68.92% of the delay of WTDMA/D.

6.6 Non-uniform traffic

In this section a non-uniform traffic model is considered. It is assumed that one station is the server in a network. 50% of the load generated at any station is destined to the server, the other half is evenly distributed among the remaining $N - 2$ stations.

6.6.1 Numerical Results

Given that the probability a station has a packet arriving to transmit to the server in a slot is $0.5p$, this means that for WTDMA/D, the probability of there not being a packet to transmit when the TDMA slot comes around is very low. If $p = 1$ and $N = 100$, then the probability of there being nothing to transmit

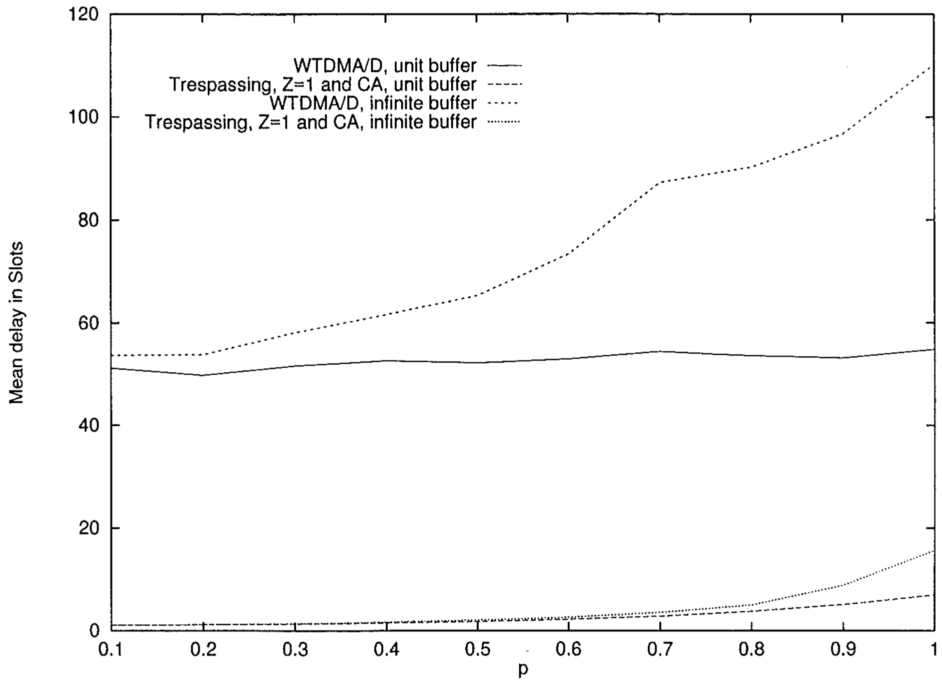

Figure 6.25: Mean delay for packets arriving at non-server stations for WTDMA/D with trespassing, $Z = 1$ and CA, with unit buffers, for the Server Model.

to the server would be 1.58×10^{-30}, even if $P = 0.1$, then the probability of there being nothing to transmit would be 6.23×10^{-3}. This means that the server's wavelength will effectively be at 100% utilisation at almost all loads. This means that there is no gain in using trespassing on this wavelength. Any gain would have to be in the throughput caused by stations transmitting to other non-server stations.

The throughput of non-server station's wavelengths is shown in Figure 6.24. As the non-server load at a station is on average $0.5p/(N-2)$, this means the maximum traffic (for $N = 100$) would be 51%. In the unit buffer case, WTDMA/D has linear rise in throughput as load increases, but can only achieve a maximum of under 40%. This compares to WTDMA/D with trespassing, Z=1 and CA, which has an almost maximal throughput of 49%. In the infinite buffer case, the differences in throughput are indistinguishable, as WTDMA/D has the optimal performance.

The mean delay of packets addressed to non-server stations is shown in Figure 6.25. In both the unit and infinite buffer cases, it can be seen that WTDMA/D with trespassing, $Z = 1$ and CA has a vastly better performance

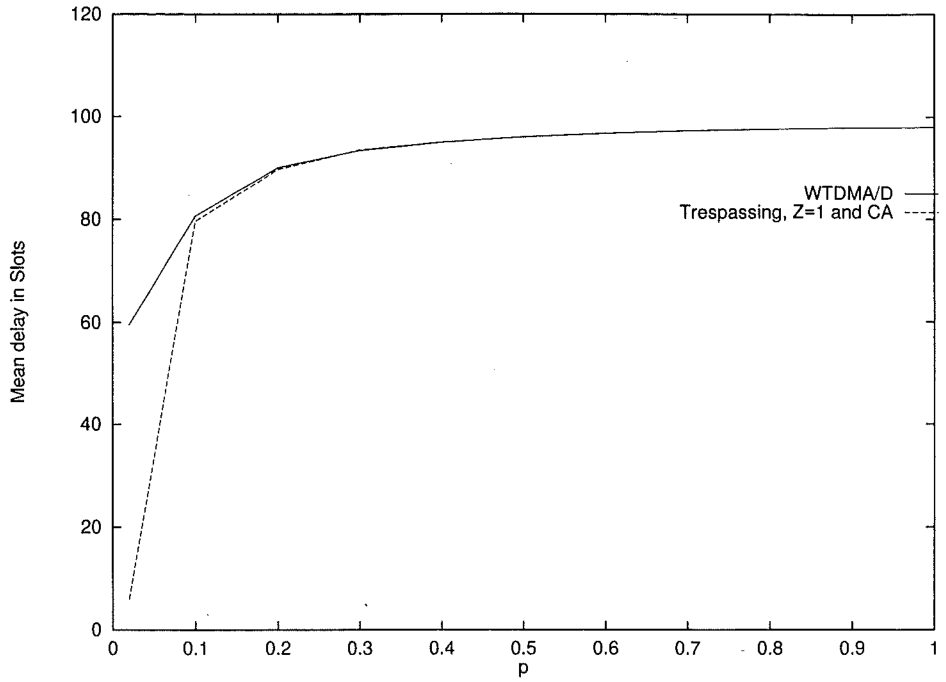

Figure 6.26: Mean delay for packets arriving at the server station for WTDMA/D with trespassing, $Z = 1$ and CA, with unit buffers, for the Server Model.

than WTDMA/D.

The mean delay for packets arriving at the server station, in the unit buffers case is shown in Figure 6.26. At high loads, because a packet will arrive almost as soon as one is transmitted, the mean delay will be close to a TDMA cycle. The mean packet delay at high loads is almost 99 slots. There is little difference between the performance of WTDMA/D and WTDMA/D with trespassing as there is little opportunity for a packet to trespass into the server's wavelength. The only difference is at very low loads, when there is more of an opportunity to trespass, as the allocated wavelength remains unused more often.

When the buffers have unlimited capacity, the load to the server must be restricted, as otherwise the buffers will become unstable. For WTDMA/D, on average $0.5p$ packets arrive in each time slot, which means $0.5p < 1/(N-1)$. In a $N = 100$ station network, this limits the load delivered to a single buffer to $p < 0.020$, if the network is to remain stable. This case in analysed in Figure 6.27. As can be seen, in this case, WTDMA/D with trespassing, $Z = 1$ and CA has a vastly lower mean delay.

If the mean delay of packets bound to the server is compared with that

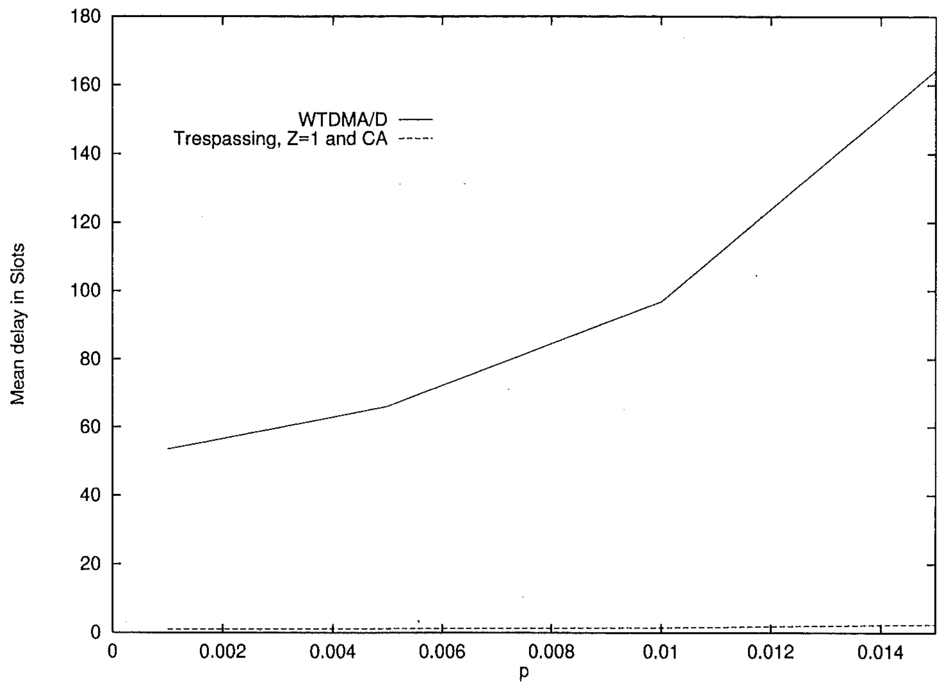

Figure 6.27: Mean delay for packets arriving at the server station for WTDMA/D with trespassing, $Z = 1$ and CA, with infinite buffers, for the Server Model.

of non-server packets, it can be seen that for unit buffers, the mean delay for non-server traffic is half that of server traffic. In the infinite buffer case, the mean delay of server traffic will never reach steady state if $p > 0.020$, and for $p = 0.015$ the mean delay has already reached over 160 slots. Even at maximum load (50 times higher than maximum load possible in the server case), a non-server packet never has to wait more than 110 slots on average.

6.7 Conclusions

On the basis of the presented results, it can easily be seen that WTDMA/D is superior to WTDMA/R in both throughput and mean delay performance. In practice WTDMA/D would be more appropriate since it better satisfies the quality of service requirements of delay sensitive traffic. Also, the perfect synchronisation required is not practical. Thus, in the following discussion, WTDMA/D is taken as the default case.

WTDMA with trespassing without collision avoidance has no better throughput than WTDMA. Any gain from trespassing packets appear to be lost by a similar number of colliding packets. The mean delay would be better, as packets are guaranteed to be transmitted with at least as much delay as WTDMA, but this result is artificial as colliding packets are lost, and so do not contribute to the mean delay.

WTDMA with trespassing and collision avoidance, in the unit buffer case does perform better than the WTDMA benchmark. In fact it has optimal performance for loads of $p < 0.5$. The maximum throughput was 0.72 packets per station per slot. This compares with the 0.63 packets per station per slot of WTDMA (14% higher throughput).

For the infinite buffer cases, WTDMA provides optimal, or near optimal throughput. Thus, there is no gain in using trespassing to increase the throughput.

In mean packet delay analysis, the unit buffer case is significantly better. For $p < 0.5$ the mean number of slots a packet is delayed, when trespassing and collision avoidance is used, is less than four slots. This compares to the almost constant 50 slots delay of WTDMA. Even for $p > 0.5$, the mean delay never gets to more than 69% of WTDMA, which is a significant difference.

If the mean packet delay is considered for stations with infinite buffers, then, for loads of $p < 0.5$, the mean delay for trespassing and collision avoidance is similar to that of stations with unit buffers, with the mean delay being always less than five slots. This compares with WTDMA, where the mean delay rises from 55 slots, to over 102 slots per packet. Once loads get higher, both trespassing and non-trespassing cases both have their mean delays rapidly increase, towards a common limit.

If WTDMA with trespassing and collision avoidance is compared with a random-access scheme, where there is always an attempt to transmit a packet, then trespassing performs slightly better, both in the sense of throughput and mean packet delay. If there was no difference, this would mean that there would be no point in implementing trespassing, when random-access works just as well.

The propagation delay of optical networks is relatively large, if compared with the transmission times. This means that it has a considerable effect on both throughput and mean delay. A round trip delay of half a TDMA cycle halves the throughput gain of trespassing over pure WTDMA. The mean packet delay is also drastically increased, but the gains from trespassing still lead to a mean packet delay of between 33% and 50% of that experienced under

WTDMA. If you care about receiving no out-of-order packets and the round trip delay is larger than a TDMA cycle, then you can never trespass, if colliding packets are reflected. Storing colliding packets at the central arbiter would improve mean delay, and prevent such problems from occurring.

If a server model is considered, where the majority of the load is generated for the server, then the throughput to the server is identical for WTDMA with and without trespassing. Such high load means for all but the smallest values of p, the throughput tends towards one. This means that if all the load was destined for the server, there would be no gain in using trespassing, except for minute loads. For non-server traffic and unit buffers, there is a 25% higher throughput with trespassing at maximum load. The mean delay of non-server traffic under trespassing is very good too, never going above a seven slot delay, compared with a 55 slot mean delay for WTDMA. If infinite buffers are considered, then for trespassing, the mean packet delay never goes above 15 slots, compared with 110 slots for WTDMA.

The novel technique of trespassing improves throughput of a WTDMA network, if collision avoidance is used. It also improves the mean packet delay in such networks, due to its random-access nature. WTDMA with trespassing can be said to be a successful hybrid of both WTDMA and random-access techniques.

Chapter 7

Conclusions and Future work

7.1 Conclusions

There are a dearth of WDM MAC protocols for star LANs (Chapter 2). These apply techniques ranging from fixed-assignment to random-access.

Hybrid protocols (Section 2.5), such as HTDM proved that combining protocols could produce a hybrid that performs better than its components.

The newly proposed concept of Trespassing (Section 3.2), in effect combines a TDMA and random-access protocol, to produce a hybrid. If a station does not have a packet to transmit on its allocated TDMA wavelength, then it will use trespassing to transmit a packet on another wavelength, with a probability that can be altered to produce the optimum results.

The performance evaluation of WTDMA with trespassing (Chapter 6) showed that, if collision avoidance was used and a station had unit buffers, trespassing offered significantly better throughput and mean delay performance than both WTDMA and pure random-access. If the station had infinite buffers, then trespassing with collision avoidance offered smaller mean delays. WTDMA already offers optimal throughput in the infinite buffer case. Even with the effects of propagation delay and prevention of out-of-order packets, trespassing with collision avoidance out-performed WTDMA.

If a non-uniform traffic server model was used, then traffic to non-server stations performed better under WTDMA with trespassing and collision avoidance than WTDMA. There was also an improvement in throughput and mean delay for traffic to the server under low loads.

WTDMA with trespassing is an effective hybrid of WTDMA and random-access techniques. The results show that it offers significant performance improvement in comparison with WTDMA without trespassing.

7.2 Future Work

The following courses of study could be worthwhile to be further investigated:

1. Extending the simulation results to include collision avoidance where reflected packets can not collide would be of interest. This could include an investigation into the effect of buffering of collided packets at the central-arbiter, instead of reflection, as this could well provide better throughput and mean packet delay performance.

2. Apply better analytical models to include formulating the throughput generated by a station running under WTDMA/D with trespassing.

3. The more realistic situation where the number of wavelengths available is smaller than the number of stations could be investigated, as could the affect of transmitter tuning times on trespassing.

4. The analytical models could be improved by applying more realistic traffic models of dependent streams of teletraffic generated by various types of communication services (computer data integrated with voice, video etc.), by assuming modulated Bernoulli, Poisson arrival processes, or more complex models of integrated teletraffic based on self-similar models.

Many hybrid protocols prove to be complex, especially since they use reservation based techniques. Full investigation of simpler random-access techniques used by trespassing would need to be done.

It should also be possible to apply the concept of trespassing to non-WDM networks, such as mobile networks. Trespassing could possibly be effective in systems where some loss of packets is acceptable. Trespassing could be applied to any form of network that uses TDMA. It would be beneficial however to use it with some form of collision avoidance, as this is where trespassing proves most valuable.

Acknowledgements

I would like to thank Associate Professor Krzysztof Pawlikowski, for his supervision of my studies. His more than able assistance in providing technical knowledge, proof-reading, and suitable literature was invaluable to the completion of this thesis.

I would also like to thank my co-supervisor Associate Professor Harsha Sirisena, who kept me on the right path during Krzysztof's absence, and who provided excellent help with my mathematical modelling.

All those that helped develop and improve Akaroa (especially Dr. Greg Ewing and Dr. Victor Yau) deserve acknowledgement, as it proved an essential tool in performing simulation analysis quickly and efficiently.

Professor Biswanath Mukherjee of the University of California in Davis deserves acknowledgement for all the help and encouragement he gave me during his all too short stay in the department.

I would like to thank my father for his help in proof-reading this thesis, and my family for supporting me throughout my studies.

Finally I would like to thank all my friends and colleagues for preventing me from winding up in a padded room, looked after by people in white coats. They might be coming for me yet, so keep up the good work ...

References

Anton, H. 1991. *Elementary Linear Algebra*. John Wiley and Sons, Inc.

Arthurs, E., Goodman, M., Kobrinski, H., & Vecchi, M. 1988. HYPASS: An Optoelectronic Hybrid Packet-Switching System. *IEEE Journal in Selected Areas in communication*, **6**(December), 1500–1510.

Arthurs, E. et al. 1986. Multiwave Optical Crossconnect for Parallel-Processing Computers. *Electrical Letters*, **24**, 119–120.

Azizoglu, M., Barry, R., & Mokhtar, A. 1995. The Effects of Tuning Time in Bandwidth-Limited Optical Broadcast Networks. *IEEE Infocom '95*, **1**, 138–145.

Bogineni, K., & Dowd, R. 1993. Impact of propagation delay on media access protocol performance for star-coupled EDM local area networks. *Milcom '93*, **1**, 298–302.

Bogineni, K., Sivalingam, K., & Dowd, P. 1993. Low-Complexity Multiple Access Protocols for Wavelength-Division Multiplexed Photonic Networks. *IEEE Journal on Selected Areas in Communications*, **11**(4), 590–604.

Borella, M., & Mukherjee, B. 1995. Efficient Scheduling of Nonuniform Packet Traffic in a WDM/TDM Local Lightwave Network with Arbitrary Transceiver Tuning Latencies. *IEEE Infocom '95*, **1**, 129–137.

Brackett, C. 1990. Dense Wavelength Division Multiplexing Networks: Principles and Applications. *IEEE Journal on Selected Areas in Communication*, **8**(August), 947–964.

Chen, M., & Yum, T. 1991. A Conflict-free Protocol For Optical WDMA Networks. *IEEE Infocom '91*, 1276–1281.

Chen, M-S., Dono, N., & Ramaswami, R. 1990. A media-access protocol for packet-switched wavelength division multi-access metropolitan area networks. *IEE Journal on Selected Areas in communications*, **8**(6), 1048–1057.

Cheung, K., Choy, M, & Kobrinski, H. 1989. Electronic wavelength tuning using acoustooptic tunable filter with broad continuous tuning range and narrow channel spacing. *Photonic Technological Letters*, **1**, 38–40.

Chipalkatti, R., Zhang, Z., & Acampora, A. 1992. High Speed Communication Protocols for Optical Star Coupler Using WDM. *Infocom '92*, 2124–2133.

Chipalkatti, R., Zhang, A., & Acampora, A. 1993. Protocols for Optical Star-Coupler Network Using WDM: Performance and Complexity Study. *IEEE Journal on Selected Areas in Communications*, **11**(4), 579–589.

Chlamtac, I., & Fumagalli, A. 1994. Quadro: A solution to packet switching in optical transmission networks. *Computer Networks and ISDN Systems*, **26**, 945–963.

Chlamtac, I., & Ganz, A. 1988a. Channel Allocation Protocols in Frequency-Time Controlled High Speed Networks. *IEEE Transactions in Communications*, **36**(April), 430–440.

Chlamtac, I., & Ganz, A. 1988b. A Multibus Train Communication Architecture. *IEEE Journal on Selected Areas in Communications*, **6**(6), 903–912.

Coquin, G., Cheung, K., & Choy, M. 1988. Single- and Multiple-Wavelength Operation of Acousto-Optically Tuned Lasers at 1.3 microns. *Proceedings 11th IEEE International Semiconductor Laser Conference*, 130–131.

Dono, N. et al. 1990. A Wavelength Division Multiple Access Network for Computer Communication. *IEEE Journal on Selected Areas in Communication*, **8**(August), 983–994.

Ewing, G., McNickle, D., & Pawlikowski, K. 1995. Credibility of the Final Results from Quantitative Stochastic Simulation. *Pages 189–194 of: Proc. European Simulation Congress.* Vienna, Austria: Elsevier, for ESC '95.

Ewing, G., Pawlikowski, K., & McNickle, D. 1996 (August). *Akaroa II Version 1.2.1 User's Manual.* University of Canterbury.

Goodman, M. et al. 1990. The LAMBDANET Multiwavelength Network: Architecture, Applications and Demonstrations. *IEEE Journal on Selected Areas in Communication*, **8**(August), 995–1004.

Green, P. 1992. An All-Optical Computer Network: Lessons Learned. *IEEE Network*, March.

Green, P. 1993. *Fiber Optic Networks.* Prentice Hall.

Hammond, J., & O'Reilly, P. 1986. *Performance Analysis of Local Computer Networks.* Addison-Wesley.

Hou, C., Wang, B., & Han, C. 1996. Design and Analysis of a WDMA Protocol for Passive Star-Coupled Lightwave Networks. *IEEE INFOCOM '96*, **3**(March), 1225–1233.

Humblet, P., Ramaswami, R., & K., Sivarajan. 1993. An Efficient Communications Protocol for High-Speed Packet Switched Multichannel Networks. *IEEE Journal on Selected Areas in Communications*, **11**(4), 568–578.

Illek, S., Thulke, W., Schanen, C., Lang, H., & Amann, M. 1990. Over 7 nm (875 GHz) continuous wavelength tuning bt tunable twin-guide (TTG) laser diode. *Electronic Letters*, **26**, 46–47.

Janneillo, F., Ramaswami, R., & Strinbery, D. 1992. A Prototype Circuit-Switched Multiwavelength Optical Metropolitan Area Network. *Submitted to ICC.*

Jia, F., & Mukherjee, B. 1991 (December). *The Receiver Collision Avoidance (RAC) Protocol for a Single-hop WDM Lightwave Network.* Tech. rept. Division of Computer Science, University of California, Davis.

Kannan, B., Fotedar, S., & Gerla, M. 1994. A Protocol for WDM Star Coupler Networks. *IEEE Infocom '94*, **3**, 1536–1542.

Karol, M, & Glance, B. 1991. Performance of the PAC Optical Packet Network. *Proceedings IEEE GLOBECOM*, December, 1258–1263.

Karol, M., & Glance, B. 1994. A collision-avoidance WDM optical star network. *Computer Networks and ISDN Systems*, **26**, 931–943.

Kazovsky, L., & Poggiolini, P. 1993. STARNET: A Multi-gigabit-per-second Optical LAN Utilising a Passive WDM Star. *Journal of Lightwave Technology*, **11**(5/6), 1009–.

Kobrinski, H., Vecchi, M., Goodman, M., Goldstein, E., Chapuran, T., Cooper, J., Tur, M., Zah, C., & Menocal, S.ereCe .C, .nd Tu. 1990a. Fast wavelength-switching of laser transmitters and amplifiers. *IEEE Journal in Selected Areas of Communications*, **8**(6), 1190–1202.

Kobrinski, H., Vecchi, M., Chapuran, T., Georges, J., Zah, C., Caneau, C., Menocal, S., Lin, P., Gozdz, A., & Favire, F. 1990b. Simultaneous fast wavelength switching and intensity modulation using a tunable DBR laser. *Photonic Technological Letters*, **2**, 139–142.

Levine, D., & Akyildiz, I. 1995. PROTON: A Media Access Control Protocol for Optical Networks with Star Topologies. *IEEE/ACM Transactions on Networking*, **3**(2), 158–168.

McNickle, D., Pawlikowski, K., & Ewing, G. 1996 (December). Experimental Evaluation of Confidence Interval Procedures in Sequential Steady-State Simulation. *In: Proc. Winter Simulation Conference*. WSC '96, San Diego.

Mukherjee, B. 1992a. WDM-Based Local Lightwave Networks Part I: Single-Hop systems. *IEEE Network*, May, 12–26.

Mukherjee, B. 1992b. WDM-Based Local Lightwave Networks Part II: Multihop Systems. *IEEE Network*, July, 20–32.

Mukherjee, B., & Meditch, J. 1987. The p_i-persistent protocol for unidirectional broadcast bus networks. *Proceedings IEEE International Conference on Communication 1987*, 563–567 (see also 1277–1286).

OFC. 1992 (February). *Technical Digest*. San Jose, California.

Pawlikowski, K., & Yau, V. 1992. On Automatic Partitioning, Runtime Control and Output Analysis Methodology for Massively Parallel Simulations. *Pages 135–139 of: Proc. European Simulation Symp. ESS '92 (Dresden, Germany, November 1992)*. So. Computer Simulation.

Pawlikowski, K., Yau, V., & McNickle, D. 1994 (December). Distributed Stochastic Discrete-Event Simulation in Parallel Time Streams. *Pages 723–730 of: Proc. Winter Simulation Conf. WSC '94*. IEEE Press, Orlando, Florida.

Poggiolini, P, & Kazovsky, L. 1991. STARNET: An integrated services broadband optical network with physical star topology. *Advanced Fibre Communications Technologies*, **SPIE Volume 1579**, 14–29.

Ramaswami, R. 1990. *Tunability Needed in Multi-Channel Networks: Transmitters, Receivers, or Both?* Tech. rept. IBM Research Division, T.J. Watson Research Center.

Semaan, G. 1993 (June). MARKAB: A slotted protocol for a WDM star network. *Pages 230–234 of: Proceedings EFOC/N '93.*

Shimosaka, N. et al. 1989. Photonic Wavelength-Division and Time-Division Hybrid Switching System Utilising Coherent Optical Detection. *Proceedings ECOC '89*, September, 292–295.

Sivalingam, K., & Wang, J. 1996. Performance of a MAC Protocol for WDM Networks with On-Line Scheduling. *IEEE INFOCOM '96*, **3**(March), 1234–1241.

Sivalingam, K., Bogineni, K., & Dowd, P. 1992. Pre-allocation media access control protocols for multiple access WDM photonic networks. *ACM Sigcomm '92*, August, 235–246.

Sneh, A., Johnson, K., & Liu, J. 1995. High-Speed Wavelength Tunable Liquid Crystal Filter. *IEEE Photonics Technology Letters*, **7**(4), 379–.

Stacey, C. 1993. *Application of Independent replications in Automatic Steady-State Simulation.* Tech. rept. Computer Science Department, University of Canterbury.

Toba, H., Odu, K., Nakanishi, K., Shibata, N., Nosu, K., Takato, N., & Sato, K. 1990. 100-channel optical FDM transmission/distribution at 622 Mbit/s over 50 km utilising waveguide frequency selection switch. *Electronic Letters*, **26**, 376–377.

Tridandapani, S., Meditch, J., & Somani, A. 1994. The MaTPi Protocal: Masking Tuning Times Through Pipelining in WDM Optical Networks. *IEEE Infocom '94*, 1528–1535.

Wong, P., & Yum, T. 1988. Design and analysis of a contention-based lookahead reservation protocol on a multichannel local area network. *IEEE Transaction s in communications*, **36**(2), 234–238.

Yau, V. 1996. *WDM Network Design and Destination Conflicts.* Ph.D. thesis, Computer Science Department, University of Canterbury.

Yau, V., & Pawlikowski, K. 1993 (February). AKAROA: a Package for Automatic Generation and Process Control of Parallel Stochastic Simulation. *Pages 71–82 of: Proc. of the 16th Australian Computer Science Conference*, vol. A. ASAC '93, Brisbane, Australia.

Chapter A

Simulation Source code

This chapter contains the source code, written in C++, used for running simulations under Akaroa2, and the altered procedures used for the various versions of the protocols.

A.1 Basic program

The following is the basic source code that all the versions of the protocols are based on.

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include "akaroa.H"
#include "akaroa/process.H"
#include "akaroa/distributions.H"
#include "akaroa/queue.H"

int N = 100; // The number of stations
real p = 1;  // The offered load

class Packet {
public:
  int destination;   // The destination station of a packet
  int source;        // The source station of a packet
  Time arrival_time; // The arrival time of a packet
};
```

```
Packet *packet;
Queue<Packet> packet_queue[100][100];
Queue<Packet> trans_queue[100];

// Transmission Process for station j
void packetTransmit(int j, int i) {
  double z;
  real z_test;

  // Check allocated buffer for a packet
  if(packet_queue[j][(i+j) % N].Length() == 0) {
    // Allocated buffer is empty, so Trespass
    // Cycle through the stations
    for (int k = N-1; k > 0 ; k--) {
      // Prevent trying to transmit to self
      if((i+j+k) % N == j % N)
        k--;
      // If a buffer is full, then try to trespass from it
      if(packet_queue[j][(i+j+k) % N].Length() > 0) {
        // Trespass probability:
        // z = 0 for WTDMA, z = 1 for deterministic trespassing
        z = (1-p)/N;
        z_test = Uniform(0,1);
        // Test to see if trespassing is allowed
        if(z_test <= z) {
          // Trespassing is allowed, so trespass:
          // Remove packet from buffer
          packet = (packet_queue[j][(i+j+k) %N].Next());
          // Transmit packet to station (i+j+k) mod N
          trans_queue[(i+j+k) % N].Insert(packet);
        }
        // Trespassing can only be attempted once,
        // so prevent any more attempts
        k = 0;
      }
    }
  } else {
    // Allocated buffer is full, so transmit
    // Remove packet from buffer
    packet = packet_queue[j][(i+j) %N].Next();
    // Transmit packet to station (i+j) mod N
    trans_queue[(i+j) % N].Insert(packet);
  }
}
```

```
// Arrival process for station j
void packetArrival(int j) {
  // Check for arrivals destined for each station
  for (int i = 0; i < N; i++) {
    // Make sure no arrivals to self
    if(i == j)
      i++;
    if(i != N) {
      // A packet arrives with probability p/(N-1)
      if(Uniform(0,1) <= p/(N-1)) {
        // Check if the unit buffer is full
        if(packet_queue[j][i].Length() == 0) {
          // Buffer is empty, so a packet can arrive
          Packet *packet = new Packet;
          packet->destination = i;
          packet->source = j;
          packet->arrival_time = CurrentTime();
          // Insert packet into buffer
          packet_queue[j][i].Insert(packet);
        }
      }
    }
  }
}

// Analyse throughput for the network
void thruputAnalysis(){
  int none = 0;
  int single = 0;
  int collision = 0;
  int queue = 0;
  Time delay;

  // Results are only taken for one station
  // Check for the number of packets transmitted to station 0
  switch (trans_queue[0].Length()) {
  case 0:
    // Nothing transmitted
    none++;
    break;
  case 1:
    // One packet transmitted
    single++;
```

```
      break;
    default:
      // More than one packet transmitted
      collision++;
  }

  // Calculate the number of packets waiting in buffers at station 0
  for (int j = 1; j < N; j++)
    queue = queue + packet_queue[0][j].Length();

  // Calculate the delay of a non-colliding packet arriving at station 0
  if(trans_queue[0].Length() > 0) {
    packet = trans_queue[0].Head();
    // Delay equals current time minus arrival time
    delay = CurrentTime() - packet->arrival_time;
    // Send observation of delay to Akaroa for analysis
    AkObservation(1, delay);
  }

  // Send number of unused wavelengths to Akaroa for analysis
  AkObservation(2,none);
  // Send number of successful transmissions to Akaroa for analysis
  AkObservation(3,single);
  // Send number of collisions to Akaroa for analysis
  AkObservation(4,collision);
  // Send number of packets in buffer to Akaroa for analysis
  AkObservation(5,queue);
}

// Clear the transmitted packets from the system
void clearQueues(){
  Packet *packet;
  int queue_length;

  for (int k = 0; k < N; k++)
    while(trans_queue[k].Length() > 0)
      trans_queue[k].Next();
}

int main(int argc, char *argv[]) {
  // Up to five parameters are analysed using Akaroa
  AkDeclareParameters(5);
  for(;;) {
    // Slot number i
```

```
    for (int i = 1; i < N; i++) {
      // Station j
      for (int j = 0; j < N; j++) {
        // Transmission process from station j to allocated station
        // (i+j) mod N
        packetTransmit(j,i);
        // Arrival process for station j
        packetArrival(j);
      }
      // All stations have had transmission and arrivals, so analyse
      // the results and clear the transmitted packets from the system
      thruputAnalysis();
      clearQueues();
      // The end of a slot, so go to the next slot, adding one to the
      // time so the delay can be correctly calculated
      Hold(1);
    }
  }
}
```

A.1.1 Infinite buffer

The following is the altered procedure used to model infinite capacity buffers in each station.

```
// Arrival process for station j, with infinite buffers
void packetArrival(int j) {
  // Check for arrivals destined for each station
  for (int i = 0; i < N; i++) {
    // Make no arrivals to self
    if(i == j)
      i++;
    if(i != N) {
      // A packet arrives with probability p/(N-1)
      if(Uniform(0,1) <= p/(N-1)) {
        // Note: no check is made for a full buffer here, as it can
        // never reach capacity, as it is infinite in size
        Packet *packet = new Packet;
        packet->destination = i;
        packet->source = j;
        packet->arrival_time = CurrentTime();
        // Insert packet into buffer
        packet_queue[j][i].Insert(packet);
      }
    }
  }
}
```

A.1.2 Collision Avoidance

The following are the altered procedure used to simulate a collision avoidance scheme for trespassing.

```
// Transmission Process for station j, with collision avoidance
void packetTransmit(int j, int i) {
  double z;
  real z_test;

  // Check allocated buffer for a packet
  if(packet_queue[j][(i+j) % N].Length() == 0) {
    // Allocated buffer is empty, so Trespass
    // Cycle through the stations
    for (int k = N−1; k > 0 ; k−−) {
      // Prevent trying to transmit to self
      if((i+j+k) % N == j % N)
        k−−;
      // If a buffer is full, then try to trespass from it
      if(packet_queue[j][(i+j+k) % N].Length() > 0) {
        // Another packet has transmitted on this wavelength, so do
        // not trespass as this will cause a collision
        if(trans_queue[(i+j+k) % N].Length() == 0) {
          // Trespass probability:
          // z = 0 for WTDMA, z = 1 for deterministic trespassing
          z = 1.0;
          z_test = Uniform(0,1);
          // Test to see if trespassing is allowed
          if(z_test <= z) {
            // Trespassing is allowed, so trespass:
            // Remove packet from buffer
            packet = (packet_queue[j][(i+j+k) %N].Next());
            // Transmit packet to station (i+j+k) mod N
            trans_queue[(i+j+k) % N].Insert(packet);
          }
        }
        // Trespassing can only be attempted once,
        // so prevent any more attempts
        k = 0;
      }
    }
  } else {
```

```
    // Allocated buffer is full, so transmit

    // Another packet has already transmitted on this wavelength. As
    // this is the allocated station, it gets guaranteed service, so
    // the trespassing packet is placed back into its queue
    if(trans_queue[(i+j) % N].Length() != 0) {
      // Remove the packet from the transmitted wavelength
      packet = trans_queue[(i+j) % N].Next();
      // As a packet may have arrived while the packet that was
      // transmitted was in transit, this must be dropped
      while(packet_queue[packet->source][packet->destination].Length() > 0)
        packet_queue[packet->source][packet->destination].Next();
      // Place the transmitted packet back into the buffer
      packet_queue[packet->source][packet->destination].Insert(packet);
    }
    // Remove packet from buffer
    packet = packet_queue[j][(i+j) %N].Next();
    // Transmit packet to station (i+j) mod N
    trans_queue[(i+j) % N].Insert(packet);
  }
}
```

A.1.3 WTDMA/R

The following is the altered procedure to simulate the WTDMA/R type network.

```
// Main procedure for WTDMA/R type network
int main(int argc, char *argv[]) {
  int i;

  // Up to five parameters are analysed using Akaroa
  AkDeclareParameters(5);
  for(;;) {
    // The allocated slot is choosen at random
    i = UniformInt(1,(N-1));
    // Station j
    for (int j = 0; j < N; j++) {
      // Transmission process from station j to allocated station
      // (i+j) mod N
      packetTransmit(j,i);
      // Arrival process for station j
      packetArrival(j);
    }
    // All stations have had transmission and arrivals, so analyse
    // the results and clear the transmitted packets from the system
    thruputAnalysis();
    clearQueues();
    // The end of a slot, so go to the next slot, adding one to the
    // time so the delay can be correctly calculated
    Hold(1);
  }
}
```

A.1.4 Propagation delay and out-of-order protection

The following are the altered procedures used to simulate propagation delay and out-of-order packet protection.

```
// Transmission Process for station j, with round trip delay of 50 slots
void packetTransmit(int j, int i) {
  double z;
  real z_test;

  // Check allocated buffer for a packet
  if(packet_queue[j][(i+j) % N].Length() == 0) {
    // Allocated buffer is empty, so Trespass
    // Cycle through the stations, but only those who will not become
    // the allocated slot within a round trip delay
    for (int k = N−1; k > 49 ; k−−) {
      // Prevent trying to transmit to self
      if((i+j+k) % N == j % N)
        k−−;
      // Check that this wavelength has not had a trespassing packet
      // within the past round trip delay slots
      if(trespass[j][(i+j+k) %N] > 0) {
        // Too soon to trespass
      } else {
        // If a buffer is full, then try to trespass from it
        if(packet_queue[j][(i+j+k) % N].Length() > 0) {
          // Trespass probability:
          // z = 0 for WTDMA, z = 1 for deterministic trespassing
          z = (1−p)/N;
          z_test = Uniform(0,1);
          // Test to see if trespassing is allowed
          if(z_test <= z) {
            // Trespassing is allowed, so trespass:
            // Remove packet from buffer
            packet = (packet_queue[j][(i+j+k) %N].Next());
            // Make sure that no packet can transmit for the next
            // round trip delay slots
            trespass[j][(i+j+k) %N] = 50;
            // Check to see if the trespassing packet has collided
            // with another packet
            if(trans_queue[(i+j+k) % N].Length() == 0) {
              // No collision
```

```
            trans_queue[(i+j+k) % N].Insert(packet);
          } else {
            // The packet collided, and so is reflected. It will
            // take the next round trip delay slots for the packet
            // to arrive back at the source station
            tres_queue[50].Insert(packet);
          }
        }
        // Trespassing can only be attempted once,
        // so prevent any more attempts
        k = 0;
      }
    }
  }
} else {
  // Allocated buffer is full, so transmit

  // Another packet has already transmitted on this wavelength. As
  // this is the allocated station, it gets guaranteed service, so
  // the trespassing packet is reflected, and will arrive back at
  // its source station after a round trip delay
  if(trans_queue[(i+j) % N].Length() != 0) {
    // Remove the packet from the transmitted wavelength
    packet = trans_queue[(i+j) % N].Next();
    // Place in a queue that simulated a round trip delay
    tres_queue[50].Insert(packet);
  }
  // Remove packet from buffer
  packet = packet_queue[j][(i+j) %N].Next();
  // Transmit packet to station (i+j) mod N
  trans_queue[(i+j) % N].Insert(packet);
  }
}

void clearQueues(){
  Packet *packet;
  int queue_length;

  // The following simulates a packet returning to its source slot
  // after being reflected. It does not get inserted back into its
  // source stations buffer until it has experienced a round trip
  // delay

  // Remove all packets that have experienced a round trip delay, and
```

```
  // place them back in their appropriate queue
  while(tres_queue[0].Length() > 0) {
    packet = tres_queue[0].Next();
    // Drop any packet that is in the queue, and replace it with the
    // reflected packet
    while(packet_queue[packet->source][packet->destination].Length() > 0)
      packet_queue[packet->source][packet->destination].Next();
    packet_queue[packet->source][packet->destination].Insert(packet);
  }
  // Decrement the count of the number of slots a packet is delayed
  for (int i = 1; i <= 50; i++)
    while(tres_queue[i].Length() > 0)
      tres_queue[(i-1)].Insert(tres_queue[i].Next());

  // The following decrements the variable used to make sure that a
  // wavelength is not trespassed on again within a round trip delay
  for (int i = 0; i < N; i++)
    for (int j = 0; j < N; j++)
      if(trespass[i][j] > 0)
        trespass[i][j]--;

    for (int k = 0; k < N; k++)
    while(trans_queue[k].Length() > 0)
      trans_queue[k].Next();
}
```

A.1.5 Server model

The following are the altered procedures used in the simulation of a server model.

```
// Arrival process for station j, under the server model
void packetArrival(int j) {
  // If the station is the server, then it has an equal probability of
  // transmitting to all other stations
  if(j == 0) {
    for (int i = 1; i < N; i++) {
      if(Uniform(0,1) <= p/(N-1)) {
        if(packet_queue[j][i].Length() == 0) {
          Packet *packet = new Packet;
          packet->destination = i;
          packet->source = j;
          packet->arrival_time = CurrentTime();
          packet_queue[j][i].Insert(packet);
        }
      }
    }
  } else {
    // If the station is not the server, then it transmits to the
    // server with probability p/2, and to a non-server station with
    // probability p/2(N-2)
    // A packet to the server with probability p/2
    if(Uniform(0,1) <= 0.5*p) {
      // Check if the unit buffer is full
      if(packet_queue[j][0].Length() == 0) {
        // Buffer is empty, so a packet can arrive
        Packet *packet = new Packet;
        packet->destination = 0;
        packet->source = j;
        packet->arrival_time = CurrentTime();
        // Insert packet into buffer
        packet_queue[j][0].Insert(packet);
      }
    }
    for (int i = 1; i < N; i++) {
      // Make sure no arrivals to self
      if(i == j)
        i++;
```

```
      if(i != N) {
        // A packet to a non-server station with probability p/2(N-2)
        if(Uniform(0,1) <= 0.5*p/(N-2)) {
          if(packet_queue[j][i].Length() == 0) {
            // Buffer is empty, so a packet can arrive
            Packet *packet = new Packet;
            packet->destination = i;
            packet->source = j;
            packet->arrival_time = CurrentTime();
            // Insert packet into buffer
            packet_queue[j][i].Insert(packet);
          }
        }
      }
    }
  }
}

// Analyse throughput for the network
void thruputAnalysis(){
  int none = 0;
  int single = 0;
  int collision = 0;
  Time delay;

  // Analyse the delay of non-colliding packets arriving at the server
  if(trans_queue[0].Length() > 0) {
    packet = trans_queue[0].Head();
    // Delay equals current time minus arrival time
    delay = CurrentTime() - packet->arrival_time;
    // Send observation of delay to Akaroa for analysis
    AkObservation(1,delay);
  }

  // Analyse the throughput of packets arriving to a non-server
  // station
  switch (trans_queue[1].Length()) {
   case 0:
     // Nothing transmitted
     none++;
     break;
  case 1:
    // One packet transmitted
    single++;
```

```
      break;
    default:
      // More than one packet transmitted
      collision++;
    }
    // Send number of successful transmissions to Akaroa for analysis
    AkObservation(2,single);

    // Analyse the delay of non-colliding packets arriving at a
    // non-server station
    if(trans_queue[1].Length() > 0) {
      packet = trans_queue[1].Head();
      // Delay equals current time minus arrival time
      delay = CurrentTime() - packet->arrival_time;
      // Send observation of delay to Akaroa for analysis
      AkObservation(3,delay);
    }
}
```

A.1.6 Random-access

The following are the altered procedures used in the simulation of a random-access protocol with collision avoidance.

```
// Transmission Process for station j, under random access with
// collision avoidance
void packetTransmit(int j) {
  double z;
  real z_test;
  int sum = 0;
  int dest;

  // Check to see if there are any packets in the stations buffers to
  // transmit
  for(int i = 0; i < N; i++)
    sum = sum + packet_queue[j][i].Length();
  // If there is a packet to transmit, then search through the buffers
  // at random, until a packet is found
  if(sum > 0) {
    // There is at least one packet to transmit
    dest = UniformInt(0,N-2);
    if(dest >= j)
      dest++;
    // Search through the buffers at random until a packet is found
    while(packet_queue[j][dest].Length() == 0) {
      dest = UniformInt(0,N-2);
      if(dest >= j)
        dest++;
    }
    // Transmit the packet, if it will not collide with another packet
    // that has already been transmitted
    // Note: This is slightly unfair on stations further down the
    // cycle, as they are more likely to have packets rejected
    if(trans_queue[dest].Length() == 0) {
      // Remove packet from buffer
      packet = (packet_queue[j][dest].Next());
      // Transmit packet to station "dest"
      trans_queue[dest].Insert(packet);
    }
  }
}
```

```
int main(int argc, char *argv[]) {
  // Up to five parameters are analysed using Akaroa
  AkDeclareParameters(5);
  for(;;) {
    // Note in this case there is no allocated slot
    // Cycle through the stations
    for (int j = 0; j < N; j++) {
      // Transmission process fo station j
      packetTransmit(j);
      // Arrival process for station j
      packetArrival(j);
    }
    // All stations have had transmission and arrivals, so analyse
    // the results and clear the transmitted packets from the system
    thruputAnalysis();
    clearQueues();
    // The end of a slot, so go to the next slot, adding one to the
    // time so the delay can be correctly calculated
    Hold(1);
  }
}
```