Comparison of Multiple Access Protocols for Optical Networks

Amy J. Troutman
COMPARISON OF MULTIPLE ACCESS PROTOCOLS FOR OPTICAL NETWORKS

by

Amy J. Troutman

This report is based on the unaltered thesis of Amy J. Troutman submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering at the Massachusetts Institute of Technology Laboratory for Information and Decision Systems with partial support provided by the National Science Foundation (NSF-ECS-8310698).
Comparison of Multiple Access Protocols for Optical Networks

by

Amy J. Troutman

BSEE North Carolina State University (1991)

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1993

© Amy J. Troutman, MCMXCIII. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute copies of this thesis document in whole or in part, and to grant others the right to do so.

Author

Department of Electrical Engineering and Computer Science

December 17, 1992

Certified by

Pierre A. Humblet
Professor of Electrical Engineering
Thesis Supervisor

Accepted by

Campbell L. Searle
Chairman, Departmental Committee on Graduate Students
Comparison of Multiple Access Protocols for Optical Networks

by

Amy J. Troutman

Submitted to the Department of Electrical Engineering and Computer Science on December 17, 1992, in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering

Abstract

In this thesis, I investigated several proposed multiple access protocols for optical passive star local area networks. Each of these protocols utilizes wavelength division multiplexing (WDM) to create many data channels. This allows for the full bandwidth of optical fiber to be used while each channel runs at a speed low enough for electronic processing to be possible. Each of these proposed access protocols differs in the approach taken to access the network, the hardware required at each station, the number of stations supported by the network, and the amount of processing and scheduling each station needed to perform prior to each datagram transmission. This thesis compares these protocols with respect to their throughput and delay capabilities. An evolution is traced through the protocols to see clearly where innovations have been made and where there remains room for further work in the area. Suggestions are offered to improve aspects of several protocols.

Thesis Supervisor: Pierre A. Humblet
Title: Professor of Electrical Engineering
Comparison of Multiple Access Protocols for Optical Networks

by

Amy J. Troutman

Submitted to the Department of Electrical Engineering and Computer Science on December 17, 1992, in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering

Abstract

In this thesis, I investigated several proposed multiple access protocols for optical passive star local area networks. Each of these protocols utilizes wavelength division multiplexing (WDM) to create many data channels. This allows for the full bandwidth of optical fiber to be used while each channel runs at a speed low enough for electronic processing to be possible. Each of these proposed access protocols differs in the approach taken to access the network, the hardware required at each station, the number of stations supported by the network, and the amount of processing and scheduling each station needed to perform prior to each datagram transmission. This thesis compares these protocols with respect to their throughput and delay capabilities. An evolution is traced through the protocols to see clearly where innovations have been made and where there remains room for further work in the area. Suggestions are offered to improve aspects of several protocols.

Thesis Supervisor: Pierre A. Humblet
Title: Professor of Electrical Engineering
Acknowledgments

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the National Science Foundation.

This work could not have been completed without the support of the following people:

- My parents, who have guided me through 17 1/2 years of education (Augh!!!) and 22 years of life

- Kathleen Wage, who has listened to me whine and gone above and beyond the call of duty to keep me laughing (and, along with Lori, Steve, and Lori, keep me passing...)

- Mel Gibson, who was there (on film, at least) to keep me from going completely over the edge when my nineteenth nervous breakdown struck

- and the Rolling Stones, the Beatles, Monty Python, U2 and all the other entertainers who, unbeknownst to themselves, kept me from losing the remaining part of my sanity and provided the soundtrack to this work.
Contents

1 Introduction .................................................. 11

2 Aloha Contention Based Protocols ......................... 13
  2.1 Habbab, Kavehrad, and Sundberg ...................... 14
    2.1.1 Aloha/Aloha ....................................... 14
    2.1.2 Analysis of Aloha/Aloha ......................... 15
    2.1.3 Slotted Aloha/Aloha ............................. 16
    2.1.4 Analysis of Slotted Aloha/Aloha ................. 16
    2.1.5 Delay Analysis of Habbab et al. ................. 20
    2.1.6 Discussion of Slotted Aloha/Slotted Aloha ...... 20
    2.1.7 Nonreliance on Feedback and Use of Acknowledgments 22

2.2 Mehravari .................................................. 23
  2.2.1 Slotted Aloha/Aloha ................................ 23
  2.2.2 Analysis of Mehravari’s Slotted Aloha/Aloha ...... 24
  2.2.3 Bimodal Throughput Analysis ........................ 25
  2.2.4 Delay Analysis of Mehravari ........................ 26
  2.2.5 Probability of Receiver Collisions ................. 31
  2.2.6 Use of Acknowledgments ............................ 32

3 Avoiding Data Collisions .................................. 34
  3.1 Slotted Aloha/N-Server Switch ........................ 34
    3.1.1 Analysis of Slotted Aloha/N-Server Switch ...... 35
  3.2 Sudhakar, Georganas, and Kavehrad ................... 36
3.2.1 Definition of Protocols ........................................ 36
3.2.2 Slotted Aloha Protocols versus Improved Slotted Aloha Protocols 39
3.2.3 Problems of a Single Transmitter/Receiver .......................... 40
3.2.4 Delays Not Considered ........................................... 41
3.2.5 Analysis of Two Types of Protocols ............................... 41
3.2.6 The First Family .................................................. 41
3.2.7 The Second Family ................................................. 43
3.2.8 Delay ............................................................. 45
3.2.9 Use of Acknowledgments ........................................... 50
3.2.10 Reservation Aloha Protocols ..................................... 50

3.3 Jeon and Un ......................................................... 51
   3.3.1 Basic Reservation Protocol ................................... 51
   3.3.2 Finite Population Analysis ..................................... 52
   3.3.3 Infinite Population Results .................................... 52
   3.3.4 Infinite Population Poisson Model .............................. 53
   3.3.5 Performance With Respect to Offered Load ..................... 54
   3.3.6 Gaussian Approximation ......................................... 54
   3.3.7 Throughput With Respect to \( L \) and \( N \) ....................... 56
   3.3.8 Delay of BRP .................................................. 56
   3.3.9 Throughput versus Delay ....................................... 58
   3.3.10 Use of Acknowledgments ..................................... 61

4 Transmission Scheduling Algorithms ................................ 65
   4.1 Slotted Buffered Reservation Protocol ............................ 65
   4.2 Lu and Kleinrock .................................................. 66
      4.2.1 Throughput Analysis of the Queued Protocols ............... 66
   4.3 Delay of Queued Systems .......................................... 67
      4.3.1 Comparison of Queued SBRP Versus BRP ...................... 71
   4.4 Queued Protocols: Jeon and Un versus Lu and Kleinrock ........ 75
      4.4.1 Discussion of Jeon and Un's Queues ......................... 75
4.4.2 Discussion of Lu and Kleinrock’s Queues .................................. 76
4.4.3 Ideas For Handling and Avoiding Collisions ............................ 76
4.4.4 Comparison in Non-Symmetric Traffic .................................... 77
4.4.5 Popular Receiver Case ...................................................... 77
4.4.6 Popular Transmitter Case .................................................. 77

4.5 Probabilistic Scheduling ...................................................... 78
4.5.1 Architecture Needs ......................................................... 78
4.5.2 Dynamic Allocation Scheme .............................................. 79
4.5.3 The Random Scheduling Algorithm ..................................... 79
4.5.4 Signaling Information Required ......................................... 80
4.5.5 Hybrid TDM ................................................................. 80
4.5.6 Signaling Information Required ......................................... 81

4.6 Scheduling to Maximize Network Throughput ............................. 82
4.6.1 Network Architecture ...................................................... 82
4.6.2 Information Maintained by Each User .................................. 83
4.6.3 The Scheduling Algorithm ................................................ 83
4.6.4 Discussion ................................................................. 85
4.6.5 Use of Acknowledgments ................................................ 85

4.7 Jia and Mukherjee: RCA ..................................................... 86
4.7.1 The Receiver Collision Avoidance Protocol .......................... 86
4.7.2 Processing and Delay Increases ......................................... 87
4.7.3 Throughput ............................................................... 87

5 “Announce and Send” Protocols .............................................. 90
5.1 Chen, Dono, and Ramaswami .............................................. 90
5.1.1 The DT-WDMA Protocol ................................................ 91
5.1.2 Analysis ................................................................. 92
5.2 Beyond Single Control Channels ......................................... 94
5.2.1 Drawbacks of a Single Control Channel ............................ 94
5.3 Semaan and Humble: Elimination of Control Channel ................ 95
5.3.1 Types of Traffic ........................................... 96
5.3.2 Datagram Traffic ......................................... 96
5.4 Humblet et al.: Multiple Control Channels ................. 97
  5.4.1 Types of Traffic ....................................... 98
  5.4.2 Datagram Traffic ...................................... 99
5.5 Comparison of Two Protocols: Datagram Traffic ............. 100
  5.5.1 The Use of Acknowledgments ........................... 102
  5.5.2 Semaan and Humblet's Acknowledgments ................. 102
  5.5.3 Humblet et al.'s Acknowledgments ...................... 103

6 Expanding Protocols which Limit the Number of Users to N 104
  6.1 The DT-WDMA Protocol .................................. 105
  6.2 The Chen and Yum Scheduling Protocol ................... 106
  6.3 Problems With Having No Control Channel ................ 107
  6.4 Reducing the Number of Required Channels in the Multi-Control Channel Approach .................. 108

7 Conclusion .................................................. 110
List of Figures

1-1 Optical Star Configuration ........................................... 12

2-1 Throughput of Aloha/Aloha Protocol versus $\frac{L}{N}$, $N = 10, \infty$ ........... 17
2-2 Slotted Aloha Maximum per Channel Throughput versus $\frac{L}{N}$ for $N = 10, \infty$ .................. 18
2-3 Comparison of Throughput for Two Protocols, $N = 5$ .................. 19
2-4 Habbab et al. Delay versus Throughput per Channel, $L = 10$, $N = 5$ 21
2-5 Data Channel Collision ................................................. 24
2-6 Throughput per Channel versus $\frac{L}{N}$ for $N = 5, 10, 20$, and infinity .... 27
2-7 Maximum Throughput per Channel Compared for Mehravari and Habbab, $N = 50$ .................................................. 28
2-8 Maximum Throughput per Channel Compared for Mehravari and Habbab, log scale .................................................. 29
2-9 Delay versus Throughput for Mehravari, $L = 10$, $N = 8$ ............. 30
2-10 Probability of No Receiver Collisions ..................................... 32

3-1 Illustration of Case 1 .................................................. 37
3-2 Illustration of Case 2 .................................................. 38
3-3 Illustration of Case 3 .................................................. 38
3-4 Illustration of Case 4 .................................................. 39
3-5 Illustration of Case 6 .................................................. 40
3-6 Per Channel Throughput of Best Slotted Aloha/Slotted Aloha Case, $N = 5, 10, \infty$ .................................................. 46
3-7 Maximum per Channel Throughput for Case 5, Mehravari, and Habbab 47
3-8 Throughput Comparison of Four Protocols, log scale, $N = 20$ 48
3-9 Delay versus Per Channel Throughput of Case 5 and Mehravari, $L = 10$
and $N = 8$ 49
3-10 Illustration of BRP Channel Assignments 52
3-11 Maximum Throughput of BRP with Respect to $\frac{L}{N}$, $N = 5, 10, 50$
(lowest curve to highest) 57
3-12 BRP Throughput per Data Channel versus $G$ for $\frac{L}{N} = 1, 2, 4, 10, 20$
(lowest curve to highest), $N = 5$ 59
3-13 Delay of BRP Depends Only on Ratio of $\frac{L}{N}$ 62
3-14 Throughput versus Delay for BRP, $N = 10$ 63
4-1 Throughput of BRP versus $\frac{L}{N}$ for $N=10$ Compared to Asymptotes $\frac{L}{e}$
and $N$ 68
4-2 Delay versus Throughput of Queued System from [12] versus BRP 70
4-3 Lu and Kleinrock Reservation Delay versus Throughput 72
4-4 Delay versus Throughput of BRP and Lu and Kleinrock, $L = 10, N = 4$ 73
5-1 Frame Structure 95
5-2 Throughput per Channel of Humblet et al. versus $m$ and $nL$ 101
List of Tables

5.1 Hardware and Wavelength Uses for Control and Data Transmission . 99
Chapter 1

Introduction

The demand for high speed networks is constantly increasing. Optical fiber offers the greatest possibilities for increasing speeds dramatically. However, optical fiber cannot yet be used at its full speed capability, which runs into terabits per second. The speed of fiber is hampered by the speed of the electronics which must be used to interface with it. This is known as the opto-electronic bottleneck. Data cannot be sent at full speed over fiber because the electronic interfaces which must be used to get the data on and off of the fiber cannot process at the high fiber speed.

Network designers have sought a way to circumvent the opto-electronic bottleneck. One of the most promising techniques is Wavelength Division Multiplexing, or WDM. WDM allows the huge bandwidth of fiber to be utilized. In WDM, the fiber bandwidth is divided into channels with each channel operating on a certain wavelength. These channels are then operated at speeds low enough to allow the electronics to keep up with the optical data. Thus, the entire bandwidth of the fiber is used constructively. The aggregate speed of the channels runs into terabits per second, with each channel operating at around one gigabit per second [15].

With the acceptance of the concept of WDM as a viable transmission technique, there have been many proposed media access protocols offered for optical networks using WDM. These have spanned several topologies, from buses to rings and stars. Most of the work, however, has been concentrated upon the optical star topology.

While many proposals advocate a slightly different exact star architecture, the
basic structure of the optical star remains unaltered. All stations are wired to a central coupler which combines all of the channels onto one fiber. At the receiving end, stations use filters to listen to the wavelengths which they want to hear. The combiner is a passive device which can take no action as a router. Transmitting lasers may be either fixed at a certain wavelength or tunable across a spectrum. The receivers also may be either tunable or fixed. Work has been done to suggest that when the number of users exceeds the number of wavelengths, tunable receivers and filters offer good performance [11].

In this thesis, eleven papers outlining proposed access protocols for WDM passive optical star networks are discussed. The aim of this thesis is to compare the proposed protocols and determine which have the best performance characteristics. The protocols are looked at from an evolutionary point of view. Different proposals can be seen to be improvements over others. From looking at this large group of proposals, one can see where there remains work to be done to improve performances. Conversely, it is also clear which areas have been exhausted and should require no further research.

In general, I have analyzed the performance of the protocols using an infinite population Poisson traffic model. Deviations from this are noted where they occur. This thesis primarily discusses throughput performance, but delay is also considered as a secondary factor. All of the analysis is conducted for datagram packet traffic.
Chapter 2

Aloha Contention Based Protocols

The first paper to address the problem of accessing an optical star passive coupler was that of Habbab, Kavehrad, and Sundberg [6]. In their paper, they investigated several variations on the general access scheme that was to become the basis for almost all subsequent work in the area.

Habbab et al. used the network architecture that so many subsequent authors would also adapt. The basic idea was to allow for there to be many more users of the network than there were WDM data wavelengths available. This allows for flexibility in configuring the network, allowing the number of users to grow and shrink easily as needed.

The basic architecture of the system is as follows. The network is an optical passive star with many wavelengths achieved via wavelength division multiplexing (WDM). One wavelength is restricted for use as a control wavelength. The remaining $N$ are used for transmitting data. Each station on the network is equipped with one tunable transmitter and one tunable receiver. When stations are not busy sending or receiving data, they are able to monitor and access the control channel.

The goal of the protocols proposed in this work is to allow the network to operate in the low throughput/small delay range by using a random access method that is simple [6]. The huge bandwidth of the fiber makes even a low throughput per channel correspond to a huge overall volume of traffic able to be moved.
2.1 Habbab, Kavehrad, and Sundberg

There are five access protocols outlined in this paper. Each consists of one protocol for the control channel and one for the data channel. The basic operation of each is as follows: in order to send a data packet, a user must first transmit a control packet to inform the intended receiver of the packet's transmission. Then the actual data must be sent on a chosen data wavelength. Successful reception of a data packet by the he destination will depend on a combination of successful control and data packets.

The random access schemes for each channel fall into two groups. The first is those based upon the Aloha protocol. The second is comprised of those protocols based upon Carrier Sense Multiple Access (CSMA). In CSMA, a user first senses a channel to see if it is idle before transmitting on that channel. The CSMA protocols exhibit good performance. However, their performance is dependent upon the normalized packet propagation time. In optical networks, this time is quite large. This factor leads to a marked decrease in the performance of CSMA protocols. In the amount of time it takes to monitor the channel for one complete propagation time, other stations may transmit and thus the likelihood of finding an idle channel is reduced. Even if a channel is found to be unused, the delay incurred while waiting for an entire round-trip delay is large. For this reason, all subsequent work based upon that of Habbab et al. addresses only the Aloha-based protocols. The CSMA ones are deemed impractical for implementation in an optical network.

With this in mind, only the Aloha-based protocols will be addressed in this work.

2.1.1 Aloha/Aloha

The first protocol uses Aloha on both the control and data channels. The sender transmits a control packet containing the address of the transmitter, the address of the intended receiver and the wavelength the data will be sent on via the control wavelength. The data packet is then immediately transmitted on the randomly chosen data channel.

Successful reception of the data depends on many factors. First, the intended
destination must have been monitoring the control channel at the time that the control packet was transmitted. Thus, the receiver couldn’t have been occupied receiving other data. Secondly, the control packet cannot have collided on the control channel with any other packet. With Aloha, there is a vulnerability period of twice the length of the packet during which a collision with any other packet would cause them both to be lost. Along this same line, the data packet must also have been transmitted without collision. Also, even if both control and data packets do not collide, the transmission will still fail if the receiver has decided to tune to another data channel at that time to receive another successful packet. In the infinite population case, which is the case analyzed, however, the chance of this happening is negligible.

2.1.2 Analysis of Aloha/Aloha

The performance of the protocol is analyzed according to an infinite population model with Poisson arrivals on the control channel at an offered load of $G$. The length of a control packet is taken to be 1; the length of a data packet is $L$. There are $N$ data channels.

On the control channel, the vulnerable time period for packets to collide is two control packet lengths, or 2. Therefore the probability of success is

$$e^{-2G}.$$  

The vulnerable time period during which collisions may occur for a data packet is $(L - 1)$ lengths of a control packet on either side of the control packet vulnerability period [6]. Thus, the total collision period is $2(L - 1)$ time units. The total offered load to a data channel is $G/N$, so the probability of success on a data channel is

$$e^{-2G/(L-1)}.$$  

The traffic offered to each data channel is $G/N$. The throughput on a data channel
is then

\[ S_d = G \frac{L}{N} e^{-2G} e^{-2 \frac{N}{2}(L-1)}. \]

The maximum value of the per channel throughput can easily be shown to occur where

\[ G = \frac{N}{2(N + L - 1)}. \]

This value corresponds to a maximum per channel throughput of

\[ S_{d,A/A,\text{max}} = \frac{1}{e} \frac{L}{2(N + L - 1)}. \]

The behavior of the throughput with respect to \( \frac{L}{N} \) for \( N = 10 \) and \( N = \infty \) is shown in Figure 2-1.

### 2.1.3 Slotted Aloha/Aloha

The second protocol is a variation of the first. Here the network is synchronized and the control channel is slotted. Users transmit control information at the start of the slots on the control channel. The data is still sent on one of the \( N \) data channels chosen at random. Data packets are again sent immediately following the transmission of control packets.

### 2.1.4 Analysis of Slotted Aloha/Aloha

In slotted Aloha, the throughput of packets arriving in a Poisson manner with offered load \( G \) is known to be \( e^{-G} \) [1]. This gives a maximum throughput on the slotted control channel of 0.36 when \( G = 1 \).

The data channel is however still operating with unslotted Aloha. The vulnerable time period in which data packets may collide is still \( 2(L - 1) \) time units. Therefore, the throughput of a data channel is

\[ S_d = G \frac{L}{N} e^{-G} e^{-2 \frac{N}{2}(L-1)}. \]
Figure 2.1: Throughput of Aloha/Aloha Protocol versus \( \frac{L}{N} \), \( N = 10, \infty \)
Figure 2-2: Slotted Aloha Maximum per Channel Throughput versus $\frac{L}{N}$ for $N = 10, \infty$

It can be shown that the maximum value of this throughput occurs where

$$G = \frac{N}{N + 2(L - 1)}.$$

At this point,

$$S_{d,SA/A,\text{max}} = \frac{L}{e(N + 2(L - 1))}.$$

The maximum throughput is plotted in Figure 2-2 versus $\frac{L}{N}$ for $N = 10$ and infinity. This result is also plotted versus $\frac{L}{N}$ for $N = 5$ as shown in Figure 2-3 together with the per channel maximum throughput of the Aloha/Aloha system. The control channel throughput governs which of the two protocols has the better per channel throughput overall.
Figure 2.3: Comparison of Throughput for Two Protocols, $N = 5$
It can be seen that for $N = 1$ and $L = 1$ the maximum throughput of a channel is $\frac{1}{e}$, the maximum for Slotted Aloha [6]. However, for $N \geq 2$ the maximum throughput becomes $\frac{1}{2e}$, which is the maximum for unslotted Aloha as $L \gg N$ [6]. Thus, the collisions on the data channels are the bottleneck of the protocol's performance.

As the next chapter will show, this bottleneck can be alleviated by altering the protocol slightly.

2.1.5 Delay Analysis of Habbab et al.

All delays in this thesis are calculated in units of $(R + 1)$ where $R$ is the length of one round trip delay. All delay calculations exclude the message transmission and acknowledgment times. Thus, delays are equal to the time until a message starts being transmitted. The minimum delay possible is one.

The delay of Aloha is easily calculated. The delay of the unslotted Aloha system is

$$D_{A/A} = e^{2G[1 + \frac{N-1}{L}]}.$$

The delay of the slotted system is similarly

$$D_{SA/A} = e^{2G[1 + \frac{N-2}{L}]}.$$

The delay versus the throughput per channel of the Aloha and Slotted Aloha systems are plotted in Figure 2-4 for $L = 10$ and $N = 5$. It is possible to achieve a much higher throughput with less delay by using Slotted Aloha on the control channel.

2.1.6 Discussion of Slotted Aloha/Slotted Aloha

A likely next step not covered in this paper would be to use Slotted Aloha on both the control and data channels. This seems logical because it has been shown that introducing Slotted Aloha on the control channel increases throughput performance over simply using Aloha on the control channel. This idea will be explored by Sudhakar
solid = Aloha/Aloha, dash = Slotted Aloha/Aloha

Figure 2.4: Habbab et al. Delay versus Throughput per Channel, $L = 10, N = 5$
et al. in a later chapter.

2.1.7 Nonreliance on Feedback and Use of Acknowledgments

One interesting aspect of the protocols seen in this paper is that they are not dependent upon users receiving any sort of feedback on the control channel for operation. Here the control and data packets are both sent immediately without waiting for any sort of information. No other protocols have this feature. In all subsequent chapters, protocols will be discussed in which it is imperative that the transmitting stations receive some sort of feedback about the success of their control packets, position in a transmission queue, etc. Thus the protocols of Habbab et al. are truly the most simple. They are not affected by propagation delays in the network. They are not reliant on receiving error-free feedback from the control channel.

However, because collisions on both the data and control channels can lead to lost packets, each station must have some way of determining whether or not its data has been successfully received. If they do not get this information via the control channel, then stations must receive some form of acknowledgment from their intended receiver in order to know whether or not their packet has been successful. The potential for loss of these acks due to transmission collisions adds another dimension of problems to the protocol. If acks were transmitted according to the same rules as packets, then their throughput would be low. Consequently, stations would have to wait an unspecified amount of time for their acks to get back to them. It would be hard to tell if the data was unsuccessful or if the ack was. Introducing feedback to the stations on the control channel could eliminate the need to send specific ack packets.
2.2 Mehravari

Mehravari's paper [13] is an improvement of the work of Habbab et al. Mehravari works with the Aloha protocols in [6] and improves them in the following manner: instead of transmitting data immediately following the transmission of the control packet, Mehravari waits until a control packet is successful. This obviously improves the performance of the protocols. Mehravari uses the same single receiver/transmitter hardware.

This paper does not deal further with the carrier sense protocols addressed in the previous ones. That is due to the fact that in a fiber optic network, the propagation delay of a packet far exceeds the transmission delay. CSMA does not exhibit good performance under these conditions [13].

2.2.1 Slotted Aloha/Aloha

Mehravari first works with the Slotted Aloha/Aloha protocol which was shown in the previous paper to exhibit better performance than Aloha/Aloha. Once again, control packets contain the addresses of the transmitter and intended recipient plus the identity of the data channel to be used. Mehravari adds the notion that data packets are only sent following the success of their corresponding control packet. This eliminates data packets being lost because of the rather high probability that a control packet will experience a collision on the control channel.

However, there remains a problem with the protocol. It is possible for two successful control packets to reserve the same data wavelength shortly after one another. Thus, due to the length of the data packets, there would be collisions on the data channel, which is essentially operating in an unslotted Aloha contention mode. Such data channel collisions lower the throughput of the network. It will be shown in a later chapter how such collisions may be avoided by slotting the data channel.
Collision on a data channel following 2 successful control packets, Mehravari

Collision of 2 data packets on same channel

Data Channel

Control Channel

Two successful control packets

Figure 2-5: Data Channel Collision

2.2.2 Analysis of Mehravari's Slotted Aloha/Aloha

Mehravari uses the same assumptions about the network as found in [6]. Again, $G$ represents the average number of control packets transmitted per control slot, $L$ the ratio of the length of a data packet to a control packet, and $N$ the number of data channels [13].

The control channel is operating in standard Slotted Aloha, and therefore has a throughput rate of

$$Ge^{-G}$$

so there are on average $Ge^{-G}$ data packets offered to all the data channels per control slot. It is assumed that data channels are chosen uniformly at random. Data slots are also $L$ times as long as control slots. Therefore it follows that the average number of data packets transmitted per slot on the data channel is found to be

$$\frac{L}{N}Ge^{-G}.$$ 

The vulnerable period during which a data packet may collide with another on the same channel is $(L - 1)$ control slot lengths on either side. Let $P_s$ be the probability that no other data packet whose control packet was successfully transmitted during the vulnerable period of a marked data packet is transmitted on the same data channel.
(say $\lambda_1$) as the marked data packet. Then it can be shown that

$$P_e = \left[1 - \frac{G}{N}e^{-G}\right]^{2(L-1)}.$$

Multiplying this by the average number of data packets transmitted per data slot on a data channel gives the throughput per data channel:

$$S_d = \frac{L}{N}Ge^{-G}\left[1 - \frac{G}{N}e^{-G}\right]^{2(L-1)}$$

### 2.2.3 Bimodal Throughput Analysis

As Mukherjee and Jia discovered in [14], the above per channel throughput equation may have a bimodal characteristic as a function of $G$. It can have three inflection points: one at $G = 1$, corresponding to a throughput of

$$\frac{L}{Ne} \left[1 - \frac{1}{Ne}\right]^{2(L-1)} \approx \frac{L}{Ne} e^{-\frac{2(L-1)}{Ne}}$$

and two where

$$Ge^{-G} = \frac{N}{2L-1},$$

corresponding to throughput of

$$\frac{L}{2L-1} \left[\frac{2(L-1)}{2L-1}\right]^{2(L-1)} \approx \frac{1}{2} e^{-1}$$

These later two optima can be verified to be local maxima. They only exist when $N$ is small relative to $L$; specifically when

$$\frac{N}{2L-1} < \frac{1}{e}$$

and the inflection point located at $G = 1$ becomes a local minimum. If these conditions are not satisfied, then the later two optima vanish, leaving the throughput characteristic unimodal with global maximum at $G = 1$. 

25
When $\frac{N}{L}$ is small, the total throughput is about $\frac{N}{2e}$, which stems from the data channel's unslotted Aloha operation. As $N$ approaches infinity, the total throughput approaches $\frac{1}{2e}$. Thus, in the limiting case, the control channel bottleneck limits the performance of the protocol.

It is interesting to note that for a given number of data channels, $N$, there exists an optimal value of the number of control slots, $L$. The per channel throughput is greatest when

$$L = \frac{N}{2} + 1.$$ 

After this point, further increasing $L$ results in decreasing the throughput until it equals $\frac{1}{2e}$. Conversely, for a fixed value of $L$ there is no optimal value of $N$. Here one encounters the law of diminishing returns. Further increases in $N$ only result in there being so many data channels that there isn't enough traffic to keep them busy. The per channel throughput is reduced, although the total throughput may increase.

For $N$ equal to 5, 10, 20, and infinity the behavior of the per channel throughput is plotted versus $\frac{L}{N}$ in Figure 2-6. For small values on $N$ the throughput reaches a maximum and then decreases as $\frac{L}{N}$ increases. However, there is no such decrease when $N \to \infty$.

2.2.4 Delay Analysis of Mehravari

The delay of this protocol is found in a manner similar to the unimproved ALOha protocols. The resulting delay here,

$$D = \frac{e^G}{1 - \frac{G}{N} e^{-G}},$$

is much higher than the delays of the protocols of Habbab due to the extra time spent waiting for control packet success. This is illustrated in Figure 2-9. In plotting this, $N$ and $L$ were deliberately chosen so that the bimodal characteristic of the throughput equation did not become a factor.
Figure 2.6: Throughput per Channel versus $\frac{L}{N}$ for $N = 5, 10, 20,$ and infinity
Figure 2-7: Maximum Throughput per Channel Compared for Mehravari and Habbab, \( N=50 \)
Figure 2-8: Maximum Throughput per Channel Compared for Mehravari and Habbab, log scale
Figure 2-9: Delay versus Throughput for Mehravari, $L = 10$, $N = 8$
2.2.5 Probability of Receiver Collisions

Because the above analysis is conducted for an infinite population case, the probability of receiver collisions, wherein more than one control packet succeeds during a slot for data packets destined for the same receiver, is rendered negligible. However, in the finite population case, the probability of receiver collisions should be investigated to see if they occur with enough frequency to affect the throughput of the system greatly.

I have calculated the probability of receiver collisions for a network with $M$ users. It is assumed that the data packets are uniformly distributed with probability $\frac{1}{M}$ of being addressed to any particular user. Using the Poisson traffic model seen earlier, the probability that a control packet succeeds in a particular control minislot is $\frac{1}{e}$.

Without loss of generality, let us look at a particular control packet which has succeeded on the control slot and has alerted a receiver, say receiver $A$, that a data packet is coming. It is assumed that receiver $A$ was monitoring the control channel at the correct time and is prepared to receive the data. If in the next $L$ minislots on the control channel, any control packet succeeds whose corresponding data packet is addressed to receiver $A$, that data packet will experience a receiver collision. By the time it reaches receiver $A$, receiver $A$ will already be receiving the first data packet sent to it.

For each control packet, the probability that it does not collide on the control channel and is addressed to receiver $A$ is $\frac{1}{Me}$. Within the $L$ minislots following a successful control packet for a receiver, the number of other successful control packets addressing the same receiver, $k$, is binomially distributed.

\[
P(k = k_0) = \binom{L}{k_0} \left( \frac{1}{Me} \right)^{k_0} \left( 1 - \frac{1}{Me} \right)^{L-k_0}
\]

A receiver collision will occur if there is more than one packet destined for receiver $A$ whose control minipacket is successful in one of the following $L$ minislots.

\[
P_{R\text{coll.}} = \sum_{k=1}^{L} \binom{L}{k} \left( \frac{1}{Me} \right)^{k} \left( 1 - \frac{1}{Me} \right)^{L-k}
\]
Figure 2-10: Probability of No Receiver Collisions

The probability that there is no receiver collision is the probability that $k = 0$,

\[ \left(1 - \frac{1}{Me}\right)^L. \]

The probability of no receiver collision is much greater than the probability of success on the control channel. Indeed, $E[k] = \frac{L}{Me} \ll 1$ when $M$ is significantly larger than $L$.

### 2.2.6 Use of Acknowledgments

The Slotted Aloha/Aloha protocol of Mehravari requires that stations receive an acknowledgment of their data packet. Although senders know their control packets have
not collided prior to their releasing data, they are not assured that the data packets will not collide. By performing additional processing of the information received from monitoring the control wavelength, each station could determine for itself whether its data packet achieved success or not. This would avoid the messy problem of each receiver having to transmit explicit acks.
Chapter 3

Avoiding Data Collisions

The next improvement to be made involved avoiding packet collisions on the data channel. While Mehravari's improved Slotted Aloha/Aloha protocol performed better than Habbab's original, it still suffered from data channel collisions. Mehravari recognized this and sought a way around them. Mehravari's solution is an ad hoc one. All other solutions to the data collision problem involve slotting the data channel.

3.1 Slotted Aloha/N–Server Switch

This protocol of Mehravari's is a modification of the CSMA-based protocols in Habbab et al. [6]. They were all more efficient than the Aloha-based ones [6]. However, as stated before, CSMA is not effective in a high speed network. It can be approximated though [13].

In this protocol the sender listens to the control channel for L consecutive control slots – the length of a data packet. From this, the sender determines which data channels are not in use. When it finds one, it immediately sends a control packet via Slotted Aloha notifying the intended receiver of the chosen data channel. Once this packet has been successful, the data packet is sent on the chosen channel.

The aim of this is to minimize the chance of data channel collisions by trying to isolate those channels which are not in use. Of course, collisions of data may still occur. More than one station may decide to use the same "empty" channel at the
same time. Because the sender cannot have instant knowledge of this, there would be a collision.

This idea will be expanded upon in a later chapter by Sudhakar et al. [16].

3.1.1 Analysis of Slotted Aloha/N-Server Switch

In performing an analysis of the system, it is assumed that users with packets to send are turned away when all the data channels are busy. Therefore, the data channels can be modeled as an N-server loss system wherein the fixed data packet length corresponds to a constant service time [13]. This would tend to lead one to evaluate the probability that all channels are busy with the Erlang-B formula [1]. However, the Erlang-B formula was developed for Poisson traffic. Here, the arrival traffic has a one-to-one correspondence with the output of the Aloha control channel. Results cited in [13] indicate that such processes are less variable than a Poisson process.

In order to approximate system throughput it must be assumed that the input to the N-server switch is Poisson. This assumption leads to a lower bound on the system throughput [13].

The arrival rate to the N-server system is the output rate of the control channel, $Ge^{-G}$. The constant service rate is $\frac{1}{T}$. Thus, using the Erlang-B formula, the blocking probability of the N-server switch is

$$P_B = \frac{(LGe^{-G})^N / N!}{\sum_{i=0}^{N} [(LGe^{-G})^i / i!]}$$

The per channel throughput is then given by

$$S_d = Ge^{-G} \cdot P_B.$$  

Comparisons made by Mehravari show that the throughput of this Slotted Aloha/N-Server Switch protocol is even greater than that of Habbab et al.'s CSMA-based protocol with normalized propagation delay of 1. Thus, this protocol exhibits the best throughput performance seen so far.
3.2 Sudhakar, Georganas, and Kavehrad

As Mehravari [13] was a direct improvement of the protocols of Habbab et al. [6], so Sudhakar et al. [16] directly improved upon Mehravari's slotted Aloha protocol. The novelty here is the slotting of the data channel. Sudhakar et al. present many different cases of protocols in their paper. Each eliminates the data channel overlap possibility that is present in Mehravari [13] by slotting the data channel and introducing the concept of cycles. Successful control packets send their data in the next data cycle.

A cycle lasts as long as the data and control portion of each transmission. For some protocols, all of the control information is sent before the data. However, in some others the control portion of the transmission occurs while the data of the last cycle is being sent. This cuts the overall length on each cycle effectively to be equal to the length of a data packet, plus the tuning time.

Each station is again equipped with one tunable transmitter and one tunable receiver. The system is again slotted and synchronized. As before, the control packets contain the source address, receiver address, and data channel. However, as will be seen, some protocols in this paper determine the data channel automatically from the slot position of the successful control channel packet. Therefore the data channel is not specified in the control slot in these protocols.

3.2.1 Definition of Protocols

Sudhakar et al. present three groups of protocols: slotted Aloha protocols, improved slotted Aloha protocols, and reservation Aloha protocols. The improved slotted Aloha protocols are exactly the same as the slotted Aloha protocols except that transmission of data is held until a control packet has been successful.

Within the first two groups are presented six cases each of Aloha protocols. Case 1 allows for \( N \) control minislots to be transmitted directly ahead of the data slots (which are of length \( L \)). Control slots are taken to be of length 1 [16]. There is a direct mapping from control minislot to data channel: success in the first minislot corresponds to data being transmitted on \( \lambda_1 \), success in the second minislot to \( \lambda_2 \), et
Case 1

![Diagram showing one cycle with data channel N, data channel 1, and control channel]

Figure 3-1: Illustration of Case 1

cetera. Thus, control channel success guarantees data packet success. If there is no successful control packet in a minislot, the corresponding data channel remains idle.

In Case 2 the N control minislots are themselves in a frame of length L, so that after the Nth minislot has passed, there is a delay of \((L - N)\) time slots until the data slots begin for that set of control minislots. The data slot from the previous cycle overlaps the control minislots of the current cycle, so not as much time is wasted. Again, there is a direct mapping of minislots to data channels so data channel assignment is implicit.

In Case 3 there is no preassignment of minislots to data channels, as there are in the first two cases. Here there are z minislots on the control channel. They are accessed in the familiar Aloha manner. After they pass, the data slots begin, similar to case 1. A user wishing to transmit on data channel \(\lambda_i\) sends a control packet in any of the z minislots and sends data on \(\lambda_i\). There is contention on both the control and data channels.

Case 4 operates like Case 2, except with z minislots, again accessed by Aloha contention. Case 4 is modified so that \(x = L\) to form Case 5. Again in both these cases, a user wishing to send data on \(\lambda_i\) will transmit a control packet in any minislot.
Case 2

Figure 3-2: Illustration of Case 2

Case 3

Figure 3-3: Illustration of Case 3
Case 4

Figure 3.4: Illustration of Case 4

and follow this by sending data on $\lambda_i$. In Case 6 each data channel runs on its own cycle and begins immediately after the transmission of its corresponding minislot out of the $N$ control channel minislots. If a sender is successful in a control minislot then its data packet will not collide on the data channel.

3.2.2 Slotted Aloha Protocols versus Improved Slotted Aloha Protocols

The original set of six protocols in [16] may be immediately dispensed with. Except for the cases where a control packet success guarantees a data packet success (Cases 1, 2, and 6), there will obviously be improvement if the transmitter waits to see if a control packet is successful before sending data. This was shown by the protocol of [13] improving the performance demonstrated by [6]. Therefore, the unimproved Aloha protocols were not looked at in this analysis.
3.2.3 Problems of a Single Transmitter/Receiver

When the authors analyze the performances of these protocols, they do so under the assumption that each node only has one tunable transmitter and receiver. Thus, a station is only able to monitor and access the control channel when it is not occupied sending or receiving data. This problem is not unique to Sudhakar et al. The same architecture is used by the two previous papers discussed.

The authors perform their analysis using an infinite population model which reduces the probability of receiver collisions to zero. They also assume that the probability of a station simultaneously trying to access the data and control channels is negligible. However, for practical applications, such situations could occur at a frequency that would render the single transmitter/receiver pair less practical. They would cut performance by at most fifty per cent. Therefore, in such cases, because of the limited hardware at each station, Cases 2, 4 and 5 of the Aloha protocols would not give performances as well as those presented for the ideal case here.

As has been seen previously, with a single receiver stations can only monitor the
control channel or receive data. They cannot do both simultaneously. Again there is the problem of having a control packet succeed without physical collision in the control minislots only to have it not be seen if the receiver it is trying to reach is busy on a data channel.

3.2.4 Delays Not Considered

In looking at the performance of the protocols, Sudhakar et al. also fail to consider the effects of the round trip propagation delay needed to determine whether there has been a successful control packet transmission or not. This delay must be accounted for to truly determine how the protocols will perform. Users must be able to have time to get feedback on the control wavelength. However, Sudhakar et al. base their analysis in [16] on the assumption that data may be transmitted instantly following control packets.

The results given in this paper also ignore the effects of propagation delays. It is assumed that they are not large enough to make a significant difference.

3.2.5 Analysis of Two Types of Protocols

Before analyzing the protocols mathematically, certain ones can be seen to be superior over others. It is also easy to see that the protocols fall into two categories: those where a control packet's success guarantees a data packet's and those where it does not. I will address the best ones out of each of two “families.”

3.2.6 The First Family

Cases 1, 2 and 6 insure that a data packet will be successful following the success of a control packet. Therefore these protocols can be expected to perform well. Of the two of them, Case 1 wastes an amount of time equal to the length of the minislots prior to each data frame. Case 2 wastes an amount of time equal to \((L - N)\) prior to the start of each data frame.

In the idealized version of Case 6 presented in the paper, the transmitters send
data immediately after a control minislot, so no time is wasted. Therefore, one would expect Case 6 to outperform Cases 1 and 2 by default. In Case 6 the transmitter/receiver does not waste time monitoring the control channel during all of the control slots. However, because of the definition of a frame as having length \( L \), Case 2 becomes idealized. The control slots are seen as taking no time at all, for they are perceived as being accessed while data is being sent in previous frames. From this, Case 2 has a throughput simply equal to the number of successful control packets from the previous frame. The network took no time to stop sending data and solely handle control information. Therefore, the throughput per channel for Case 2 is just the standard for Slotted Aloha,

\[
Ge^{-G}.
\]

In Case 6, the network halts data transmissions for only one minislot length on each data cycle. The throughput of the data channel is the throughput of the control channel, \( Ge^{-G} \), times the ratio of the length of a data frame to the length of the entire data plus control cycle. Thus Case 6 has the per cycle throughput of

\[
\frac{L}{L+1}Ge^{-G}
\]

- the standard for slotted Aloha, which is what the protocol amounts to in the limiting case as \( L \) increases.

Similar calculations show that Case 1's throughput per cycle is

\[
\frac{L}{L+N}Ge^{-G}.
\]

So as the number of data channels, \( N \), increases, Case 1 becomes less and less efficient. More time is spent waiting for the data portion of the transmission to begin. Cases 2 and 6 are unaffected by the number of data channels.

The maximum throughput of these three cases occurs when \( G = 1 \), providing a maximum throughput of \( \frac{1}{e} \) in all in the limiting case as \( L \rightarrow \infty \).

In each of these Cases, \( N \) must be less than \( L \). For this reason, there are no plots
given to compare this family to any of the other protocols. The restriction on $N$ makes comparisons unjust.

3.2.7 The Second Family

Cases 3, 4 and 5 form another like group. Here there are no preassigned minislots as in the first family. This family is in essence a Slotted Aloha/Slotted Aloha protocol, with the added provision that data packets are only sent after their corresponding control packets have succeeded. These protocols resemble the Slotted Aloha/Aloha protocol presented by Mehravari [13] with the prevention of data channel overlaps added by slotting the data channels as well as the control. Therefore, their performance is similar to this. Indeed, the throughput equations for these three cases take on the same form as Mehravari's. Again, they have the bimodal characteristic found by Mukherjee and Jia [14].

The total throughput for Case 3 is

$$\frac{L}{L + x N} Ge^{-G} \left(1 - \frac{G}{N} e^{-G}\right)^{x-1}.$$

This has optima at $G = 1$ which exists when $\frac{x}{N}$ is small and corresponds to a throughput of

$$\frac{L}{L + x N} e \left(1 - \frac{1}{Ne}\right)^{x-1} \approx \frac{L}{L + x Ne} e^{-\frac{(x-1)}{Ne}},$$

and optima where

$$Ge^{-G} = \frac{N}{x},$$

corresponding to throughput of

$$\frac{L}{L + x} \left(1 - \frac{1}{x}\right)^{x-1} \approx \frac{L}{L + x} e^{-1}$$

which exists only when $x > Ne$.

It can be seen that in the limiting case, as the number of minislots goes to infinity, the maximum throughput becomes zero. the behavior in this case, when $\frac{x}{N}$ is large, is
similar to Case 1. The slotted Aloha contention on the control channel always limits the performance ability of the protocol. In this case, a successful control packet does not insure a successful data packet. Naturally, the performance of this Case is poorer than the Cases 1 and 6 where data packet success follows control packet success.

Cases 4 and 5 behave similarly to Case 3. The total throughput per cycle for Case 4 can be found to be

\[ \frac{Gx}{N} e^{-G} \left[ 1 - \frac{G}{N} e^{-G} \right]^{x-1}. \]

Again, this behaves bimodally. When \( \frac{x}{N} \) is small the maximum throughput occurs when \( G = 1 \) and equals

\[ \frac{x}{Ne} \left[ 1 - \frac{1}{Ne} \right]^{x-1} \cong \frac{x}{Ne} e^{-\frac{x}{Ne}}. \]

When \( x > Ne \) the maximum occurs where

\[ Ge^{-G} = \frac{N}{x} \]

and equals

\[ \left[ 1 - \frac{1}{x} \right]^{x-1} \cong e^{-1}. \]

From this it can be seen that when \( x \gg N \) Case 4 provides the same result as Case 2, so it works out just as well to use Case 2.

Looking at the frame structure of Case 4, one may ask the question: is it ever better to make \( x < L \)? It is not. Case 5, wherein Case 4 is altered so that \( x = L \), provides the best throughput performance of the second family. Here the throughput per cycle may be found to be

\[ \frac{GL}{N} e^{-G} \left[ 1 - \frac{G}{N} e^{-G} \right]^{L-1}. \]
Where $\frac{L}{N}$ is small the maximum value occurs where $G = 1$ and equals

\[ \frac{L}{Ne} \left[ 1 - \frac{1}{Ne} \right]^{(L-1)} \approx \frac{L}{Ne} e^{-\frac{(L-1)}{Ne}}. \]

When $L > Ne$ the maximum occurs at

\[ Ge^{-G} = \frac{N}{L} \]

and equals

\[ \left[ 1 - \frac{1}{L} \right]^{(L-1)} \approx e^{-1}. \]

This is the best throughput of this family. It is graphed in Figure 3-6 versus $\frac{L}{N}$ where $N = 5, 10$ and infinity.

The throughput, like that of Mehravari’s protocol, has a peak value for small $N$’s. Once $N \to \infty$ this peak disappears and the throughput no longer decreases as $\frac{L}{N}$ increases.

The throughput has higher peaks as the number of minislots is increased. The peak occurs at lower values of offered load, $G$, as the number of minislots goes up, as the above formulas indicate.

The maximum throughput per channel of the Aloha-based schemes of Habbab, Mehravari, and Sudhakar’s Case 5 are compared in Figure 3-7 on a linear scale and in Figure 3-8 on a log scale for the case where $N = 20$.

### 3.2.8 Delay

The delay of Sudhakar’s Case 5 is considered. The average delay, assuming the ideal case of infinitely quick tuning, can be expressed by

\[ D_{case5} = (r - 1) \cdot \left( \frac{L + 1}{2} + 1 \right) \]
Figure 3-6: Per Channel Throughput of Best Slotted Aloha/Slotted Aloha Case, $N = 5, 10, \infty$
Figure 3-7: Maximum per Channel Throughput for Case 5, Mehravari, and Habbab
Figure 3-8: Throughput Comparison of Four Protocols, log scale, N = 20
Figure 3-9: Delay versus Per Channel Throughput of Case 5 and Mehravari, $L = 10$ and $N = 8$

where $r$ is the average number of retransmissions of a control packet required. Here, because of Aloha on the control channel,

$$r = e^G.$$

The delay versus per channel throughput performance for Case 5 is compared with that of Mehravari for $L = 10$ and $N = 8$ in Figure 3-9. One can see that the increase in throughput performance slotting the data channel provides comes at a cost of higher delay.
3.2.9 Use of Acknowledgments

In Cases 1, 2 and 6 where a successful control packet guarantees a successful data packet, senders do not need to get acknowledgments of their transmissions. However, in the Cases where data success is not inherent senders must again be sent some form of ack to determine if they were successful. Monitoring the control channel and deriving whether data was successful from the information contained on the control channel is again possible here.

3.2.10 Reservation Aloha Protocols

Reservation Aloha is a derivative of standard Aloha wherein a station that succeeds in getting a reservation on a data channel is given exclusive use of that channel until it is finished transmitting its data on it. For example, if a sender manages to successfully transmit a control packet requesting the use of data channel λ₅, then it not only is allowed to use λ₅ for sending one data packet in the next cycle, but it may continue to send data on λ₅ as long as it needs to without having to send control packets to reserve the channel each time. When that sender is finished transmitting, other users will see that λ₅ is no longer in use. They may then contend for it via the control channel. This sort of access scheme is of use when stations are sending isochronous traffic which demands that channels be reserved for fixed amounts of time.

Cases 1 and 6 are expanded to Reservation Aloha in the final section of the paper. Again, since Case 6 exhibited superior performance over Case 1 in the original presentation of the protocols, one would expect its R-Aloha version to be superior to Case 1’s, and it indeed is. In performing their calculations, Sudhakar et al. conclude that the Case 6 – based R-aloha protocol exhibits better throughput and lower delay than the Case 1 – based version.

Once again, however, they neglect to include the feedback delay necessary in the network in their calculations. Their results do not reflect the true operational characteristics of such a network and, moreover, may not be entirely accurate in predicting which has the best performance.
3.3 Jeon and Un

In Jeon and Un's paper [8, 9], several protocols are presented. They all stem from the Basic Reservation Protocol (BRP) originally published in [8]. The hardware configuration differs slightly from the previous papers. Those assumed that each station only had 1 tunable transmitter and receiver, and that idle stations monitored the control channel. In [9] Jeon and Un assign to each station a tunable receiver and transmitter for data and a fixed transmitter/receiver pair to monitor the control channel. Thus, Jeon and Un avoid the problem which occurred in the previous papers, namely the inability of a user to access the control channel simultaneously with a data channel. Of course, this is only seen mathematically for a finite user population, and it is only significant in practice for users that are very busy.

Jeon and Un set up the size of their data and control slots to take into account the propagation delay in the network. However, this assumption does not appear necessary. The previous papers ignore this altogether. Slotted Aloha is again used to access the control channel. Control channel success is defined as a control packet not colliding with others, not having the same destination as other successful packets, and the number of successful reservations being less than the number of data channels, $N$. This restriction on the number of successful control packets is the key difference. Control packets need only contain the source and destination addresses.

3.3.1 Basic Reservation Protocol

The BRP operates in the following manner. A non-collided control packet which is the $i$th non-collided packet out of all the control packets transmitted by one of the $M$ users ($M \geq N$) corresponds to data being transmitted on the $i$th wavelength in the next cycle, with cycles overlapping each other so that there is no wasted time. Data packets are sent after one round-trip propagation delay during which potential senders learn if they had one of the $N$ successful control packets.
3.3.2 Finite Population Analysis

Jeon and Un analyze the BRP’s performance in [9] using a finite user analysis, unlike all the previous papers which did an infinite population case. They numerically investigate the delay-throughput characteristics of the BRP for different values of $N$, $M$, and the number of control minislots, $L$ [9, 8].

Due to the Aloha control channel, the maximum average number of successful uncollided control packets is $\frac{L}{e}$. Thus, when $\frac{L}{e} \geq N$ the maximum throughput can be $N$. But even if there are enough minislots the maximum throughput can still be small due to destination conflicts [9].

3.3.3 Infinite Population Results

Jeon and Un state that when the user population is relatively high in comparison to the number of channels, the system throughput becomes independent of the user population. Using the infinite population model, the number of users, $M$, would be so high as to render the probability of destination conflicts negligible. Thus, maximum throughput is limited only by the throughput of the control minislots and the number of data wavelengths. The maximum throughput per channel is then bounded by

$$\leq \min \left[ \frac{L}{Ne}, 1 \right]$$
where \( L \) is the number of minislots on the control channel per frame and \( N \) is the number of data channels. If \( \frac{L}{Ne} \gg 1 \), the protocol can achieve a throughput of 1 per channel per cycle, or a total system throughput of \( N \) data packets per cycle. Intuitively, if there were many more control minislots than data channels, there would be a good chance of a successful reservation being made for every one of the \( N \) data wavelengths, so a total throughput of \( N \) would be achievable.

On the control channel, the maximum number of average successful transmissions without a collision is \( \frac{L}{\epsilon} \). Per channel then, the throughput becomes \( \frac{L}{Ne} \), the same maximum throughput result as the other protocols exhibited in the limiting case.

### 3.3.4 Infinite Population Poisson Model

I performed an infinite population analysis of the BRP based upon a Poisson model of arrivals. Because of the infinite population model, the probability of destination conflict is zero, reducing the requirements for control packet success to 1) no collision on the control channel, and 2) the number of currently successful control packets \(< N\), the number of data channels. Due to slotted Aloha, the probability of no collision in the control minislots is \( Ge^{-G} \) [1].

The probability that there are \( k \) successful control packets out of \( L \) possible minislots is

\[
\binom{L}{k} [Ge^{-G}]^k [1 - Ge^{-G}]^{L-k}.
\]

The probability that the number of successful control packets is less than the number of data channels is

\[
\sum_{k=0}^{N-1} \binom{L}{k} [Ge^{-G}]^k [1 - Ge^{-G}]^{L-k}.
\]

Therefore, the total throughput is the expected number of successes:

\[
S = N - \sum_{k=0}^{N-1} (N - k) \binom{L}{k} [Ge^{-G}]^k [1 - Ge^{-G}]^{L-k}.
\]

This throughput must be examined with respect to values of \( G, L, \) and \( N \) to
determine what the optimal characteristics of the protocol are. First the behavior was examined with respect to the offered load, $G$.

### 3.3.5 Performance With Respect to Offered Load

It is easily shown that the maximum value of the throughput occurs where $G = 1$. This corresponds to a maximum system throughput of

$$S_{\text{max}} = N - \sum_{k=0}^{N-1} (N - k) \binom{L}{k} e^{-k} \left(1 - \frac{1}{e}\right)^{(L-k)}.$$  \hspace{1cm} (3.2)

The sender sees the same effect whether a packet on the control channel collides with another or does not collide but must be discarded because there are already more than $N$ successful reservations. In either case, that control packet has been wasted and the station incurs a delay of one round-trip propagation time before it must retransmit another reservation packet to the control channel. Therefore, there is nothing to be gained by lowering the offered load to allow fewer reservation packets to be offered to the control channel, thus lowering the number of uncollided control packets. Moreover, there is no tradeoff in efficiency between allowing packets to succeed on the control channel only to have their reservations denied due to an excess of successes and lowering the number of control packets offered, and thus increasing the delay incurred by a data packet while it waits for its control packet to be generated.

### 3.3.6 Gaussian Approximation

The above system throughput can be simplified as $N$ approaches infinity. Using the Central Limit Theorem the binomial probability density function can be approximated by a Gaussian density [5]. The throughput per channel equals

$$S = 1 - \sum_{k=0}^{N-1} \frac{1}{N} \binom{N-k}{k} [Ge^{-G}]^k [1 - Ge^{-G}]^{L-k}.$$
The Gaussian normal probability density function is

\[ f_k(\beta) = \frac{e^{-\frac{(\beta - \mu)^2}{2\sigma_k^2}}}{\sqrt{2\pi\sigma_k}}. \]

Here,

\[ E[k] = L \cdot Ge^{-G} \]

and

\[ \sigma_k = \sqrt{L \cdot Ge^{-G} \cdot (1 - Ge^{-G})}. \]

For large \( N \) the total throughput per channel becomes

\[ S \simeq 1 - \sum_{k=0}^{N-1} \frac{(N - k)}{N} \cdot f_k(k) \]

The sum can be replaced by an integral:

\[ S \simeq 1 - \int_{k=0}^{N-1} \frac{(N - k)}{N} \cdot [f_k(k)] \, dk. \]

Finally,

\[ S \simeq 1 - \int_{k=0}^{N-1} \frac{(N - k)}{N\sqrt{2\pi} \sqrt{L Ge^{-G} (1 - Ge^{-G})}} \cdot e^{-\frac{(k-Le^{-G})(k-Le^{-G})}{2L Ge^{-G} (1 - Ge^{-G})}} \, dk. \]

Where \( G = 1 \),

\[ S \simeq 1 - \int_{k=0}^{N-1} \frac{e(N - k)}{N\sqrt{2\pi} L(e - 1)} \cdot e^{-\frac{(k-L)^2}{2L(e - 1)}} \, dk. \]

It is possible to express this result in terms of \( \frac{N}{L} \). Let

\[ k' = \frac{k}{L}. \]

Then

\[ dk = Ldk'. \]
and

\[ S \simeq 1 - \int_0^L \frac{N}{NL} \frac{e^{(N/L - k')}}{\sqrt{2\pi L(e - 1)}} \cdot e^{-\frac{(L' - \frac{1}{2})^2}{2L(e - 1)}} \cdot Ldk'. \]

\[ S \simeq 1 - \int_0^\frac{N}{L} \frac{e^{(N/L - k')}}{\sqrt{2\pi L(e - 1)}} \cdot e^{-\frac{\left(\frac{L'}{e} - \frac{1}{2}\right)^2}{2L(e - 1)}} \cdot dk'. \]

When \( L \to \infty \) the variance of this result, \( \frac{\varepsilon - 1}{L} \), goes to zero. When \( \frac{N}{L} \) is less than the mean, \( \frac{1}{e} \), the throughput goes to one. If \( \frac{N}{L} \) is greater than the mean, the throughput becomes

\[ 1 - \frac{e^{\left(\frac{N}{L} - \frac{1}{e}\right)}}{\frac{N}{L}} = \frac{L}{Ne}. \]

This result is confirmed in the following section.

### 3.3.7 Throughput With Respect to \( L \) and \( N \)

Figure 3-11 shows the behavior of the per channel maximum throughput with respect to \( \frac{L}{N} \) for values of \( N = 5, 10, \) and 50. The curves converge with little variation. As \( N \to \infty \) the throughput converges to the asymptotic bounds \( \frac{L}{e} \) and \( N \). Thus, the throughput can really be described by the ideal bound.

It can be shown for Equation 3.2 that for \( L \) larger than \( \approx Ne \), \( S_{\text{max}} = N \). \( S_{\text{max}} \) is linear increasing with slope \( \frac{1}{e} \) while \( \frac{L}{e} < N \) as a function of \( L \). Once \( \frac{L}{e} > N \), the throughput is no longer linear, but is very close to linear with a slope of \( \approx \frac{1}{e} \) while \( L \sim 2N \). Once \( L \sim 2N \) the slope of the throughput curve decreases from a value of \( \frac{1}{e} \) to a value approaching 0 as the throughput approaches its maximum value of \( N \) asymptotically approaches \( N \).

### 3.3.8 Delay of BRP

In [9] the delay of the BRP is not calculated for an infinite population case. Such an analysis is undertaken here.

The propagation delay is taken to be \( R \) and the transmission delay to be 1. The probability of retransmitting a failed packet in the next slot is also 1. The total delay,
Figure 3.11: Maximum Throughput of BRP with Respect to $\frac{E}{N}$, $N = 5, 10, 50$ (lowest curve to highest)
\[ D, \text{ is given by} \]
\[ D = D_R + (R + 1) \]

where \( D_R \) is the reservation delay. Thus, the average packet delay is

\[ E[D] = E[D_R] + (R + 1). \]

The following section will concentrate on \( E[D_R] \), as \((R + 1)\) is a constant factor. The reservation delay will be expressed in units of \((R + 1)\).

### 3.3.9 Throughput versus Delay

It is easy to plot the behavior of the BRP throughput in Equation 3.1 with respect to varying values of \( G \). The maximum throughput will always occur where \( G = 1 \).

The larger the value of the ratio of \( \frac{L}{N} \), the higher the value of maximum throughput per data channel will be. This value approaches one for increasing values of \( \frac{L}{N} \). The behavior of the throughput per channel is plotted for \( \frac{L}{N} = 1, 2, 4, 10 \) and 20 in Figure 3-12 where \( N = 5 \).

The question of stability arises at this point. Aloha protocols are always unstable. However, stability reigns when the output rate of the system exceeds the input rate. Because the Aloha channel performance is limited to be at most \( N \) total packets transmitted per cycle, or 1 per data channel, the traditional Aloha throughput performance curve is flattened out at \( N \) (or 1 in the per channel throughput case). This flat region extends, for large values of \( \frac{L}{N} \), beyond the \( G = 1 \) point which is the traditional beginning of the unstable region. Therefore, the region of stability is extended. The amount of extension is increased with higher \( \frac{L}{N} \) values.

For example, if \( \frac{L}{N} \) were equal to four, and the desired throughput of the system were 0.8, the stable region extends to around \( G \approx 2.25 \). Because \( G \) is the offered load per control minislot, \( G \approx 2.25 \) is equivalent to 2.25 attempted transmission per minislot. Because \( \frac{L}{N} = 4 \) this system would be able to handle four minislots per data channel, each with 2.25 transmissions in it, or nine transmissions per data channel every slot and still remain stable for a long time, in fact forever if the number of
Figure 3-12: BRP Throughput per Data Channel versus G for $\frac{L}{N} = 1, 2, 4, 10, 20$ (lowest curve to highest), $N = 5$
active users never exceeds $9N$.

However, if $\frac{L}{N}$ is, say, smaller than two then this stability region does not exist. Here, in order to keep the Aloha system stable there must be some monitoring of the traffic and some alteration of the frequency with which every user retransmits.

The investigation into the throughput/delay characteristics for the protocol now continue. From Little’s Law it is known that the throughput times the average delay equals the number of packets in the system, $x$ [1]. If we look at the system as being one data channel then the throughput

$$S = \frac{S_{\text{total}}}{N}.$$ 

The number in the system is $\frac{x}{L}$, where $x$ is the total number of packets generated. So from Little’s Law,

$$S_{\text{total}}D_R = x$$

so

$$SD_R = \frac{x}{N}.$$ 

Moreover, $\frac{x}{L} = G$ because all users that have packets are transmitting. Thus,

$$SD_R = \frac{L}{N}G$$

so the average retransmission delay, $D_R$, equals

$$D_R = \frac{L}{N}G.$$ 

This $D_R$ is the reservation delay per $(R + 1)$.

We know that

$$S \simeq \min(1, \frac{L}{N}Ge^{-G}).$$

For small values of $G < 1$ the $S$ versus $G$ curve is close to linear. In this region $S$ can
be approximated as a linear function of $G$, where

$$e^{-G} \approx 1 - G$$

when $G$ is small. Thus

$$S \approx \frac{L}{N} G(1 - G).$$

This rough approximation leads to

$$D_R \approx \frac{1}{1 - \frac{L}{N} S},$$

$$D_R \approx 1 + \frac{N}{L} S.$$

It is possible to use the linear approximation $S \approx \frac{L}{N} G$ for small $G$ to start finding the actual value of $G$ corresponding to a given value of $S$. A quick iterative algorithm accomplishes this task. Given a value of $S$, an initial value of $G_{\text{guess}}$ is found from $G_{\text{guess}} = \frac{N}{L} S$. From this $G_{\text{guess}}$ value the corresponding throughput value, $S_{\text{guess}}$, is found using the throughput formula, Equation 3.1. If it does not equal the given $S$ value, a new $G_{\text{guess}}$ is calculated using a linear approximation. The $x$ coordinate of the point on the line from $(0,0)$ to $(G_{\text{guess}},S_{\text{guess}})$ having $y$ component equaling $S$ is the new $G_{\text{guess}}$. This algorithm produces the true $G$ value in a few iterations.

Thus, the actual value of delay, $D_R = \frac{L G}{N S}$, may be found easily for any given throughput per channel value $S$. It depends, of course, on $\frac{L}{N}$.

Graphing throughput versus delay shows that the delay is only affected by the ratio $\frac{L}{N}$, and not by actual values of $L$ and $N$, as shown in Figure 3-13.

It can also be seen in Figure 3-14 that increasing values of $\frac{L}{N}$ lower the delay incurred reaching higher throughput values.

### 3.3.10 Use of Acknowledgments

The BRP can be operated without the need for specific acknowledgments. All users have information from the control channel about their transmissions. There is enough
$L/N = 10$: solid: $N = 5, L = 50$, dash: $N = 3, L = 30$

Figure 3-13: Delay of BRP Depends Only on Ratio of $\frac{L}{N}$
solid: L/N = 3, dot: L/N = 5, dash: L/N = 10

Figure 3-14: Throughput versus Delay for BRP, N = 10
information for receiver collisions to be avoided. Stations do not have to devote resources to handling acknowledgments.
Chapter 4

Transmission Scheduling Algorithms

The next class of proposed protocols requires that transmissions be scheduled. Potential senders announce their desire to transmit and are assigned a time and wavelength on which to do so by a scheduling algorithm. There are two main issues to focus on in this technique. The first involves how users announce that they wish to send a packet. This information must reach the network before any assignment of transmission times can take place. The second issue is the scheduling itself. Two papers in this chapter take up the first issue, those of Jeon and Un and Lu and Kleinrock. Scheduling algorithms are addressed in the second half of this chapter.

4.1 Slotted Buffered Reservation Protocol

The Slotted Buffered Reservation Protocol (SBRP) [9] introduces queuing. All reservation packets sent to the control channel which do not collide with others are considered successful. They are then queued, one queue per destination. Queued packets are transmitted on a first come, first served basis. This eliminates destination conflicts. However, a station may now find itself scheduled to transmit to different destinations simultaneously. Source conflicts have now arisen. Analyzing performance using an infinite population model renders their probability negligible.
4.2 Lu and Kleinrock

Lu and Kleinrock also incorporate the idea of queuing successful control packets into their protocol [12]. Their protocol is the same as Jeon and Un's SBRP, using the same hardware, with a few minor alterations. Lu and Kleinrock divide slots on the control channel into reservation and tuning minislots. There are $L$ reservation minislots. Here stations contend for reservations via slotted Aloha. All reservations which do not collide with others are then queued in a common distributed queue. The only information necessary in a reservation minislot would be the identity of the station wishing to be allowed to transmit data.

There are $N$ tuning minislots, wherein the first $N$ stations waiting in the common distributed queue send a tuning "minipacket" to announce the destination of the data packet they will send in the next slot. The position of the reservations in the queue determines the assignment of tuning minislots. The position of the tuning minislot uniquely determines the data wavelength to be used for transmission, i.e., there are $N$ data channels and the tuning minislots are preassigned to them. However, this tuning minipacket is not really needed in practice as the receiver and transmitter are located at the same station and thus can share the same information.

Because they are only using one queue, Lu and Kleinrock may encounter destination conflicts in which a receiving station sees more than one packet addressed to it in the upcoming slot. In this case, the receiver chooses one to be successful, and the other packets destined for it must restart the process by securing a reservation and queuing once again.

4.2.1 Throughput Analysis of the Queued Protocols

The two queued protocols give exactly the same throughput in the infinite population case. Both arrive at a maximum throughput of the $L$ Aloha channels is $\frac{L}{e}$ so the total system throughput will be $min\left[\frac{L}{e}, N\right]$ where $N$ is the number of data channels [12]. In the limiting case, the total throughput of the nonqueued BRP also equaled this result. In actuality, the BRP does not achieve this bound, but the throughput performance
is not too different as seen in Figure 4-1.

4.3 Delay of Queued Systems

It has been shown that the queued systems of Jeon and Un and Lu and Kleinrock have the same maximum throughput. This throughput is better, but not greatly so, than the maximum throughput for Jeon and Un's unqueued BRP. In Section 4.3.1, the delay of the two queued systems will be compared with that of the nonqueued.

Under the infinite population model, the SBRP's destination queues behave as one large queue. The probability of more than one packet being addressed to the same destination is negligible. Therefore, the destination queues reduce to one queue where up to $N$ packets may leave the queue at each cycle. Thus, the queuing delay may be modeled as Lu and Kleinrock model theirs.

As with my throughput calculations, I investigated the delay performance under an infinite population assumption. Lu and Kleinrock calculate the delay of their protocol under such an assumption. Their result is the sum of reservation, queuing, propagation, and transmission delays. The round trip propagation delay is set to be $R$; the transmission delay is normalized to be 1. The reservation delay is the expected delay incurred due to the possibility of collisions on the slotted Aloha reservation channel. Lu and Kleinrock use a batch queue model to arrive at the expected value of the queuing delay.

The total delay is:

$$D = D_R + D_Q + (R + 1)$$

where $D_R$ is the reservation delay, equaling the time between the packet arrival and the instant the success of the reservation is known $[12]$. $D_Q$ is the queuing delay, or the time from the instant the success of the reservation is recognized until the beginning of successful transmission of the data. The propagation delay is $R$; the transmission delay is 1.
Figure 4-1: Throughput of BRP versus $\frac{L}{N}$ for N=10 Compared to Asymptotes $\frac{L}{e}$ and $\frac{L}{N}$
The average packet delay is therefore

\[ E[D] = E[D_R] + E[D_Q] + (R + 1). \]

The reservation delay is calculated to be

\[ E[D_R] = (R + 1) \left( e^G \right). \]

The average queuing delay is given in [12] by

\[ E[D_Q] = \left[ \frac{E[C] + E[X]}{N} \right] \]

where

\[ C = \lim_{t \to \infty} C_t \]

is the length of the queue after a slot in steady state, and \( X \) is the position of a tagged successful reservation among others in the same frame. If there are \( A_{t+1} \) successful requests in slot \( t + 1 \) then

\[ C_{t+1} = \max[0, C_t + A_{t+1} - N]. \]

From this, it is possible to calculate \( C(z) \), the \( z \)-transform of the \( C \) function. From this, Lu and Kleinrock determine that

\[ E[C] = - \sum_{i=1}^{L-N} \frac{1}{1 - z_i}, \]

where the \( z_i \)'s are the poles of \( C(z) \). These poles can then be found. Finding \( E[X] \) just requires the first and second moments of \( A \):

\[ E[X] = 1 + \frac{1}{2} (L - 1) Ge^{-G}. \]

Thus, the total \( E[D] \) can be found [12].
Figure 4-2: Delay versus Throughput of Queued System from [12] versus BRP

Lu and Kleinrock provide a graph in [12] showing the delay/throughput characteristic of their protocol for $M = 500$, $N = 4$, $L = 10$, $R = 10$, and the probability of retransmitting failed packet in the next slot $p = 0.2$. This plot is reproduced in Figure 4-2 after being normalized by $(R+1)$. Figure 4-2 also plots the delay/throughput curve for Jeon and Un's BRP for the same $L$ and $N$. However, this curve was generated assuming the probability of retransmission in the next slot, $p$, equaled one. Because Lu and Kleinrock use $p = 0.2$ their delay will be higher, although not by much as this has been normalized by $R + 1$.

Figure 4-2 shows that the two systems have very similar delays. From this it can be surmised that the queuing delay of Lu and Kleinrock is negligible compared to the reservation delay. It is interesting to compare the reservation delay and throughput.
of Lu and Kleinrock to that of Jeon and Un.

We may define Lu and Kleinrock's reservation delay as the number of retransmissions needed before a control packet is successful. This is $e^G$ for Slotted Aloha. The throughput of the reservations is also familiar. The throughput per Aloha control slot is

$$Ge^{-G}$$

so the throughput for the $L$ control slots is

$$LGe^{-G}.$$ 

Thus the throughput per data channel of the system is

$$\min \left( \frac{L}{N} Ge^{-G}, 1 \right).$$

This throughput and delay are plotted in Figure 4-3. For $N = 4$ and $L = 10$ this result is compared to Jeon and Un's BRP in Figure 4-4.

4.3.1 Comparison of Queued SBRP Versus BRP

As has been shown in Figure 4-1, the throughput of the queued protocol is slightly higher than that of the nonqueued BRP. However, as $N$ becomes large, the BRP throughput converges to equal that of the queued system. Therefore, any throughput advantage the queued system has over the BRP is slight. The next area of comparison for the two systems is in delay performance. In Section 3.3.9 the throughput versus delay characteristics of the BRP were analyzed. This result may be compared with the result given in their paper by Lu and Kleinrock.

Looking at the graphs of delay versus throughput for the BRP and the queued system (Figure 4-2), it is apparent that the systems have the same delay characteristics when both are normalized by $(R + 1)$. Because Lu and Kleinrock's delay behaves with so similarly to the unqueued system, it can be concluded that the queuing delay is negligible with respect to the retransmission delay incurred trying to get into the
solid: $L/N = 2.5$, dash: $L/N = 1$

Figure 4-3: Lu and Kleinrock Reservation Delay versus Throughput
solid = Lu and Kleinrock, dot = BRP; L = 10, N = 4

Figure 4-4: Delay versus Throughput of BRP and Lu and Kleinrock, L = 10, N = 4
queue. When $R$ is significantly large, this is the case.

What is of real interest is the behavior of the system when $L \gg N$. In this case, the control minislots are not saturated. The bottleneck of both systems occurs at the data channels. In the queued system, this causes the queue to become backed up. Queuing theory states that the delay of a queue will blow up to infinity in this situation. However, as shown in Figure 3-14, the delay of the unqueued system remains constant when $L \gg N$. Thus in this case it would be concluded that the unqueued system provides much lower delay. However, the question of the stability of Aloha must arise again at this point. It is known that as Aloha approaches saturation, it is unlikely to remain in the low delay region. When it becomes more unstable, the delay of an Aloha system becomes infinite. Therefore, to achieve the constant delay performance predicted by the model, the Aloha system must be kept stable. There are techniques for doing this involving adjusting the probability of retransmission in the next slot, $p$. As the system approaches saturation, the best performance is no longer achieved when $p = 1$. The probability $p$ must be lessened in order to retain stability. This effectively lessens the offered load to the system which enables it to remain in the stable region.

With the queued systems, network maintenance becomes harder. It is more complicated to bring users on line. Stations are more dependent than ever on the correctness of maintaining their knowledge of where they stand in the queue to know when they are supposed to transmit. Whether the queues provide enough performance enhancement to be worth the effort of maintaining them must be decided. If the underlying Aloha reservation system in the BRP does not go unstable quickly, then the BRP will give virtually the same delay performance and only slightly inferior throughput performance when compared to the queued system.
4.4 Queued Protocols: Jeon and Un versus Lu and Kleinrock

Because Jeon and Un's SBRP is so close to Lu and Kleinrock's protocol, it is interesting to compare the two ideas. Both have the same premise: all successful reservations are placed into a queue. Then, each cycle, a number of packets equal to the number of data channels are selected for transmission. The difference in the two protocols lies in the concept of the queues.

4.4.1 Discussion of Jeon and Un's Queues

Jeon and Un envision there being $M$ data queues, where $M$ is equal to the number of nodes in the system. Thus, from these $M$ queues, $N$ - being the number of data channels - would be selected to send their data each cycle. Jeon and Un do not specify the manner in which such a selection would be made. \(^1\)

Clearly, if the receiver queues were selected according to some preset scheme, i.e., select receiver queues $1 - N$ in the first cycle, queues $N + 1 - 2N$ the next, and so on, there would be a possibility that queues would be selected which contained no data at that time. Therefore, there would be data channels which would be idle in that cycle. Moreover, the throughput of the network would be less than the optimal value due to this waste of channel resources.

However, if the queue transmission selection scheme insures that only receiver queues containing some data are selected then the throughput would be able to reach optimality. The possibility of idle data channels would be reduced. This would minimize waste in the overall network. Such a system will be seen in Section 4.6.

In the case of symmetric traffic, which is the case dealt with in the analysis of the protocol's performance, the exact method of selecting receiver queues to transmit is not important, as long as each queue has an equal opportunity to transmit to ensure fairness. However, in the case of unsymmetric traffic, the selection algorithm can

\(^1\)It is usually assumed that $M \geq N$
be arranged to allow for higher throughput. Severely backed up queues can also be avoided.

In the case where one receiver is especially popular, the selection algorithm can be adjusted to select that destination's receiver queue more often to avoid a large backup at that particular queue. This could, however, cause questions of fairness to arise and so would have to be managed carefully.

### 4.4.2 Discussion of Lu and Kleinrock's Queues

Lu and Kleinrock's protocol places all successful reservations into one common, distributed queue. Then, at each cycle, the first $N$ are selected for transmission. This protocol allows for the possibility of receiver collisions to occur. Among the packets selected for transmission, there may be more than one destined for the same receiver. Jeon and Un's SBRP encounters the complementary problem of collisions at the transmitter. That is, when packets are selected to be transmitted, the possibility exists for a transmitter to be scheduled to send data to more than one receiver simultaneously.

### 4.4.3 Ideas For Handling and Avoiding Collisions

Lu and Kleinrock mention in [12] a technique for dealing with collided packets. They state that the receiver should choose one packet to be successful and the other one should be dropped. This collided packet would then have to begin the transmission process over again from scratch, starting with endeavoring to obtain a reservation and get into the queue again. This would require that all stations keep copies of their data packets until their reception has been assured. Such a protocol adds to the overhead that all stations must manage. ²

It is always preferable to halt transmissions at the transmitter. It would be possible to avoid potential receiver collisions by introducing a slightly more complicated technique for selecting data packets for transmission. Some schemes for this are discussed in the next sections. Rather than simply selecting the first packets at the

²Chen and Dono advocate a similar collision scheme in [3].
head of the queue, an algorithm could be introduced to look at the destination of each packet and insure that only the first $N$ packets having different destinations are selected. As long as there were at least as many different destinations as data channels, no channel would ever sit idle with such a scheme. To avoid the possibility of requests getting passed over multiple times for transmission, their time of arrival can be easily considered. Geometrically, the requests can be visualized as being a part of a cube; they are held in a “matrix of queues” positioned by their sender, their intended destination, and their time of arrival.

### 4.4.4 Comparison in Non-Symmetric Traffic

In the case of nonsymmetric traffic, the two queuing approaches do not exhibit exactly the same performance any longer. There are two basic ways to classify nonsymmetric traffic to get a simple understanding of their differences. The first is the case wherein some stations expect to receive more traffic than others. The second case occurs when certain stations are busier transmitting data than others.

### 4.4.5 Popular Receiver Case

As mentioned above, the multiple receiver queues approach of Jeon and Un [9] is more flexible in allowing for a popular receiver to be selected to receive data more often. In Lu and Kleinrock’s protocol [12] there would be a highly increased likelihood of receiver collisions in such a case. In increasingly extreme scenarios, even additional scheduling algorithms would not greatly improve the performance of one distributed queue. There would in all likelihood be idle data channels as a large number of stations attempted to talk to one particular receiver. However, such a result would be seen with the multiple receiver queue approach as well in such a drastic case.

### 4.4.6 Popular Transmitter Case

In this case neither protocol is superior. Both will be limited by a transmitter’s inability to send more than one packet at once.
4.5 Probabilistic Scheduling

The previous section outlined the need for scheduling algorithms to process the order in which the transmission requests are fulfilled. Three are discussed in the following sections. The first approach is found in the paper by Chipalkatti, Zhang, and Acampora [4]. In these scheduling protocols, the control channel is heavily relied upon to disseminate information necessary to the operation of the protocol. Like the RCA protocol [10], the protocols proposed in this paper also avoid the problem of receiver collisions on the data channel by ensuring that each receiver may be sent only one data packet at a time.

4.5.1 Architecture Needs

The architecture used here is the same as in Jeon and Un's protocol. All nodes are outfitted with a fixed transmitter and tunable receiver to access the data channels. Additionally, though, each station is provided with a fixed transmitter/receiver pair locked onto the control wavelength. In the Random Scheduling Algorithm (RSA) of [4], the information on the control channel is so crucial to the correct operation of the algorithm that each station must have constant access to it. Therefore the stations must have the extra transmitter/receiver set.

In any of the protocols requiring stations to work out their transmissions from either logical queue information maintained by each user or a probabilistic algorithm as used here, errors on the control channel can lead to drastic problems for transmission. The probabilities and consequences of such errors have not yet been quantified.

Because each user sends data on a preassigned frequency with a fixed transmitter, the maximum number of users of the network is equal to the number of data channels. Proposed protocols which impose similar size restrictions on the network will be seen in later chapters.

As with the other protocols, this called for a time synchronized system and fixed length packets. The duration of a slot is defined to equal a packet's transmission time.
4.5.2 Dynamic Allocation Scheme

The Dynamic Allocation Scheme (DAS) is the first scheduling algorithm presented by Chipalkatti et al. Here, it is assumed that all newly generated data packets at each node are placed into separate destination buffers, which is reminiscent of Jeon and Un. This requires that each station maintain $N$ receiver queues, where $N$ is the number of users on the network.

All stations additionally maintain the queue state for every other station on the network. This information is broadcast to all via the control channel. Jeon and Un did this also. All users must also ensure that they update the state of each of their destination buffers after each transmission. Each station maintains a complete “snapshot” of the entire state of the network. This differs slightly from the required information maintained by the users of Lu and Kleinrock’s protocol. There users need only keep track of their positions in the distributed queue along with the reservations which entered and left the queue in each cycle.

The DAS requires an algorithm to be executed simultaneously by every user prior to the transmission of every data packet. All stations run an identical algorithm to arrive at a mutual transmission decision. This again is done by the previously discussed queuing algorithms. This eliminates the need for explicit acknowledgments, as all stations know the decision of all others regarding transmissions. Additionally, all receivers know at the same time where to tune to receive their upcoming data packet.

4.5.3 The Random Scheduling Algorithm

The algorithm executed prior to each slot is the RSA. It eliminates the possibility of more than one data packet arriving at any receiver during a frame. It begins by selecting a transmitter at random from all the users. From this chosen transmitter’s non-empty receiver queues, one is randomly selected. This receiver cannot be selected by any other transmitter in the network. If the selected transmitter has no unempty receiver queues, no receiver is selected and the data channel remains unused.
The algorithm then moves on to select a different transmitter at random, and for it a different receiver to send data to. The algorithm continues until it has assigned all transmitters to either send data to a distinct receiver or remain idle.

### 4.5.4 Signaling Information Required

All of the queue state information necessary to the operation of the RSA is broadcast via the control wavelength, $\lambda_0$. This channel is broken into frames of the same length as a data packet. Frames are then further slotted into $N$ minislots which are assigned to the $N$ users via Time Division Multiplexing (TDM).

At the end of each data slot, each user broadcasts on $\lambda_0$ which of its receiver queues experienced arrivals. This can be done with $N$ bits (1 if an arrival, 0 if not). Thus, every frame there are $N$ bits transmitted per minislot by $N$ users, for $N^2$ bits total in the $N$ minislots combined. Thus, the signalling channel capacity required is

$$capacity = \frac{N^2}{\Delta}$$

where $\Delta$ is the length of a frame. So, if the frame size were of length $\approx 1000$ bits and the channel capacity 1 Gps, $N^2 = 1000$. Chipalkatti et al. state that this would severely limit the size of the network to $N \leq 30$ [4], a ridiculously small limit. To get around this serious restriction, the Hybrid TDM protocol is introduced.

### 4.5.5 Hybrid TDM

In standard TDM each frame is divided into $N$ slots and one is preassigned to each user for transmitting to each other user. In Hybrid TDM, the frames are divided into $N + M$ slots, where $N$ is the number of users and $M$ is an integer [4]. The $N$ slots are preassigned as in TDM. During each of these TDM slots, a transmitter sends a packet to its designated destination. If the receiver queue for that particular destination is empty, the slot goes unused and is therefore wasted.

After every $\frac{N}{M}$ slots, one slot is left "open." During these slots, the Random Scheduling Algorithm is used to choose which transmitters send data to which re-
receivers. Thus, the RSA is run only $M$ times per $N + M$ slots in Hybrid TDM, whereas with the DAS it was run $N$ times per every $N$ slots. It would be possible to run the RSA only once for the entire frame of $N + M$ slots. All of the $M$ free slots could be scheduled at once, seeing as it is known what stations will have in their receiver queues for the entire frame.

### 4.5.6 Signaling Information Required

A slightly different signalling system is proposed here, as stations need only know the state of the network every $\frac{N}{M}$ slots. At the end of each "open" slot, each transmitter broadcasts the state – empty or not – of each of its receiver queues. This information takes into account the packets they will send via TDM in the next TDM period. One bit per queue is sufficient for this, so each of the $N$ stations sends $N$ bits. In the Hybrid TDM system, this information need only be sent every $\frac{N}{M} \Delta$ seconds. Therefore, the signalling channel capacity needed is

$$\text{capacity} = \frac{N^2}{\frac{N}{M} \Delta} = \frac{NM}{\Delta}$$

With a packet length of $\approx 1000$ bits and channel capacity of 1 Gps, $NM = 10^3$. Therefore, a slightly larger number of stations than in DAS can be supported, i.e., $N = 100, M = 10$ or $N = 200, M = 5$. [4]. However, this is still a protocol which is not taken seriously because of its limitations.
4.6 Scheduling to Maximize Network Throughput

Another proposed protocol which makes use of a scheduling algorithm is presented by Chen and Yum in [2]. Like the protocol of Chipalkatti et al. and the queued protocols of Jeon and Un and Lu and Kleinrock, this protocol uses an algorithm to schedule the transmission of data packets to avoid collisions. Like the RCA protocol, this protocol allows for receiver collisions to be avoided. The idea which separates this from the previous papers is that the scheduling algorithm here seeks to maximize network throughput. With this algorithm, a throughput of $N$ packets per transmission time is possible.

4.6.1 Network Architecture

As in the protocol by Chipalkatti et al., each station in the network is assigned its own unique data wavelength. Thus, if there are $N$ WDM data channels present, there is a maximum on $N$ stations that can be a part of the network. Each station has two transmitters: one set at the assigned data wavelength of the station and one on the control channel wavelength. Each station has two receivers: one on the control channel and one tunable over the data wavelengths. Chen and Yum point out that if each station has two tunable receivers, tuning time of the receiver can be omitted from the calculations of delay and throughput.

The system is slotted and synchronized. The slots on the data channels are the length of a data packet. The control channel is subdivided into $N$ minislots per slot. Each of these minislots is assigned uniquely to a user, i.e., slot $i$ is assigned to the $i$th user. These minislots are used by the stations to broadcast data packet destination addresses.
4.6.2 Information Maintained by Each User

As with the other papers in which each station must run an algorithm in order to determine transmissions or maintain a queue, each station on the network must maintain certain information about the state of all stations in the network. Chen and Yum envision this information being held in matrix form, which would certainly also be applicable to all the other scheduling protocols. Chen and Yum allow each station to maintain a "Backlog Matrix",

$$B = [b_{ij}]_{N \times N}$$

where $b_{ij}$ indicates the number of packets at station $i$ waiting to be transmitted to station $j$.

The $b_{ij}$ entries are updated every slot. When a new packet arrives for transmission at a station, the station announces its destination address in the station's preassigned control minislot. Thus, every slot each user receives an update about the new packets in the network. Unlike the protocol of Chipalkatti et al. the stations do not send a complete "snapshot" of the state of the entire network each cycle. Sending only the changes each time cuts down on the signaling required. Sending only the updates is the approach taken by the queuing systems of Lu and Kleinrock and Jeon and Un.

All stations use the same matrix $B$ to run the same scheduling algorithm so that all stations know simultaneously which packets are being sent to which destination.

4.6.3 The Scheduling Algorithm

From the $B$ matrix, every slot each station runs an algorithm to schedule data packets for transmission in the next slot. The end result of the algorithm is the "Transmission Matrix",

$$T = [t_{ij}]_{N \times N}$$
where \( t_{ij} = 1 \) if station \( i \) should transmit a packet to station \( j \) and \( t_{ij} = 0 \) otherwise, \( \forall i, j \). During the algorithm, each station generates the matrix

\[
D = [d_{ij}]_{N \times N}
\]

where \( d_{ij} = 1 \) if \( b_{ij} > 0 \) and \( d_{ij} = 0 \) otherwise, \( \forall i, j \).

A \( T \) matrix has at most one nonzero element in every row and column. Thus, to maximize throughput and schedule as many packets for transmission in each slot as possible, the \( T \) matrix with the largest rank should be found. Chen and Yum show in [2] that this is equivalent to solving the integer linear program:

\[
\text{Maximize} U(T) = \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} \cdot t_{ij}
\]

subject to

\[
\sum_{i=1}^{N} t_{ij} \leq 1, 1 \leq j \leq N
\]

\[
\sum_{j=1}^{N} t_{ij} \leq 1, 1 \leq i \leq N
\]

\( \forall t_{ij} \in \{0, 1\} \).

This integer linear program can be solved to yield the \( T \) matrix with maximum rank. However, the number of operations required to do so is too high for the application needed here. Therefore Chen and Yum propose an algorithm which finds a suboptimal solution to the problem of finding a \( T \) with the greatest rank. The heuristic they propose successively chooses non-zero entries of \( D \) that result in the greatest number of remaining non-zero entries. These chosen entries become the entries in \( T \). They argue that the rank of a matrix is likely to be higher the more non-zero entries it has.

The algorithm proposed by Chen and Yum often produces optimal \( T \)'s. Letting \( L_{\text{max}} \) be the maximum loss of a station's scheduling efficiency with the use of the suboptimal algorithm, \( D_{\text{max}} \) be the difference in rank between the optimal and sub-
optimal T's, and δ be the percentage of times the ranks of the two T's are different,

\[ L_{\text{max}} = \frac{D_{\text{max}} \delta}{N}. \]

As an example of the efficiency of their algorithm, Chen and Yum calculate \( L_{\text{max}} = 8.7 \cdot 10^{-4} \) for \( N = 40 \).

### 4.6.4 Discussion

The queuing algorithms of Lu and Kleinrock [12] and Jeon and Un [9] also use the idea of notifying all stations only of updates to the overall network state, in the form of successful control packets. However, when they transmit from their queues they do not utilize any maximum throughput algorithm like the one described by Chen and Yum. Using such an algorithm, which is easy to implement in the matrix form, would improve the throughput of the queued systems while allowing there to be more users of the network than data channels, thus circumventing Chen and Yum's size restriction.

This protocol also clearly produces better throughput results than that of Chipalkatti et al. [4]. Their probabilistic scheduling approach did not maximize throughput. With their random RSA algorithm, stations could be forced to sit idle although they had packets to send because of the scheduling assignment procedure. The protocol of Chen and Yum lessens this probability. Now, each slot, as many packets are sent as possible, minimizing the probability of a station which has data to send being forced to sit idle.

### 4.6.5 Use of Acknowledgments

There is no need for explicit acknowledgments to be sent to transmitters with this protocol. Because of the nature of the scheduling algorithm there exists no possibility of receiver collisions. Because each station has a unique data wavelength there is no opportunity for channel collisions. Thus all stations know that, baring some error which would be handled at a higher level, the data will be delivered as scheduled.
4.7 Jia and Mukherjee: RCA

Scheduling can eliminate the potential for receiver collisions. This would improve the theoretical throughput of the network. However, such scheduling algorithms require more time to be spent processing the algorithm and managing the schedule. This increases the overhead of the network. However, if such increases in processing requirements and delay are tolerable, then the efficiency of the network may be improved by the use of scheduling to avoid receiver collisions.

In [10], Jia and Mukherjee seek direct improvement over Mehravari's protocol [13]. They point out that while Mehravari ignored the effect of receiver collisions by using an infinite population model for his analysis, they in fact cause a significant degradation in the protocol's performance. Hence, Jia and Mukherjee's work employs a scheduling algorithm to avoid them [10]. However, as has been shown, the probability of receiver collisions is really quite small. Therefore this protocol is really going to great lengths to avoid only a minor problem.

4.7.1 The Receiver Collision Avoidance Protocol

The name of this protocol is the Receiver Collision Avoidance Protocol (RCA). Jia and Mukherjee use the same hardware configuration as [13, 6, 16]: a passive star with \( N \) WDM channels where each station is equipped with only one tunable transmitter and one tunable receiver. The RCA protocol seeks to avoid two kinds of collisions. The first are receiver collisions which occur when the receiver of the intended destination is not listening to the control channel at the time of an arrival of a control packet from the sender because it is listening to other data. The second is the event of two data packets which are sent following successful control packets overlapping on a data wavelength.

The basic operation of the protocol may be summarized as follows. Each station in the network maintains a list, called the Node Activities List (NAL), of all of the traffic that has been sent over the control channel during the last cycle in which the station was monitoring the control channel. From this, the sending station can
determine whether the intended receiver will be busy receiving another packet or idle, and thus monitoring the control channel.

A sender first determines whether the intended receiver is idle by consulting its (the sender's) latest NAL. If it appears that the receiver is idle, the sender transmits a control packet on the control channel. If this packet does not collide with another, its success will automatically determine what wavelength the data will be sent on. Much like Jeon and Un's BRP in [8], a success at the first "control point" corresponds to the first data wavelength, \( \lambda_1 \), at the second to \( \lambda_2 \) et cetera. The "control points" act like Aloha minislots seen in previous protocols. The sender keeps monitoring the control channel while the control packet is propagating to the receiver to listen for other successful control packets which may arrive at the destination first. If there is a collision or conflict, the sender will start over later. If not, the data packet is scheduled for transmission by the sender and receiver. There will be no form of receiver collisions with this scheme.

### 4.7.2 Processing and Delay Increases

The obvious drawback of using the RCA protocol is that it increases the amount of processing each station must perform. Now, not only does every station have the task of trying to monitor the control channel as often as possible to determine the outcome of its transmissions, but it must also try to calculate the status of an intended destination for data via the NAL. The control channel becomes increasingly crucial to the operation of the RCA protocol. While the delay each data packet sees due to receiver collisions is eliminated, the pretransmission delay is increased.

### 4.7.3 Throughput

As they did in [14], Jia and Mukherjee analyze the throughput of the RCA protocol using an finite population model that ignores the traffic interdependency of each station and assumes that traffic is instead uniformly distributed. Each station transmits a control packet with probability \( p \) at each "control point." They assume there are
$M$ users.

The throughput per control point on the control channel, which operates basically according to Aloha, is

$$S_c = M p(1 - p)^{M-1}.$$  

There are $N$ data channels. Therefore at most $N$ control packets can be sent during a cycle. Thus, the aggregate system throughput on all $N$ data channels is

$$S_t = N S_c \underbrace{P_{nrc}}_{\text{no receiver collision}}.$$  

where $P_{nrc}$ is the probability that there will be no receiver collision after a control packet is received.

$P_{nrc}$ is determined to be $1 - (\text{the probability that there are any more successful control packets with no receiver collisions on any of the control points between the transmission and arrival of the first successful control packet})$. Letting $C$ be the number of control points in this period,

$$P_{nrc} = 1 - C \frac{1}{M} S_c P_{nrc}.$$  

Therefore,

$$P_{nrc} = \frac{M}{M + C S_c}.$$  

Substituting into the equation for $S_t$ gives

$$S_t = N M p(1 - p)^{M-1} \frac{1}{1 + C p(1 - p)^{M-1}}.$$  

It can be easily shown that $S_t$ is unimodal and exhibits its maximum at $p = \frac{1}{m}$.

Thus,

$$S_{t,\text{max}} = N \left(1 - \frac{1}{m}\right)^{M-1} \frac{1}{1 + C \frac{(1 - M)^{M-1}}{M}}$$  

$S_t$ reaches its maximum simultaneously with the control channel. Consequently, the system throughput is limited by the throughput of the control channel. This is unlike the original protocol, wherein throughput was limited by collisions on the
control channel and on the data channels. Jia and Mukherjee show that the RCA throughput is higher than that of the Mehravari protocol [10], as would be expected. However, the gain in throughput is not worth the expense of increasing complexity, processing, and delay which the RCA protocol demands.
Chapter 5

“Announce and Send” Protocols

The next type of protocols to be discussed share a common feature. In each, stations wishing to send a data packet announce their intention in some reserved space and then send their packet. Therefore there is no contention. Moreover, each of these protocols requires stations to have dedicated wavelengths for their exclusive use. This dictates that the size of the network be held to no larger than the number of WDM wavelengths available.

5.1 Chen, Dono, and Ramaswami

This paper, [3], has a similar general structure to the Aloha-based protocols presented by Habbab et. al and those who modified the original Aloha protocols, as seen in earlier chapters. However, the protocol presented by Chen et al. has features also seen in other papers. It can be viewed rather like a cousin to all of the other papers presented in this thesis, for it is related to each in some way.

The hardware of this protocol varies slightly with others seen so far. Here each station is equipped with two transmitters: one fixed on the control channel and one fixed on a data wavelength unique to that station. Thus, there is a size restriction to the network: there can be no more users than data wavelengths. Each station is also equipped with two receivers. One is fixed upon the control wavelength while the other may be tuned to any data wavelength to receive incoming packets. As with the
other protocols, the system is assumed to be slotted and synchronized.

5.1.1 The DT-WDMA Protocol

The control and data channels are all divided into slots. Control slots are further subdivided into \( N \) minislots, where \( N \) is equal to the number of users of the network. Like the early Aloha papers, stations wishing to send a data packet must first announce this intention to their chosen recipient. This is as usual done over the control channel. Control packets sent in the control minislots consists of the address of the intended data packet destination. They also may include information about the delay the packet has experienced since its arrival at the source (used in contention resolution wherein packets that have been waiting longer are chosen to be successful when there is a conflict) and whether the source is in packet-switched or circuit-switched mode. However, herein the access to the control channel is via Time Division Multiplexing (TDM) and not slotted Aloha. TDM has the disadvantage of making the number of users in the network fixed and very hard to change. The maximum number of users is set by the number of channels which are provided from WDM. Thus, such a protocol is of use only in a small network.

The Dynamic Time-Wavelength Division Multiple Access (DT-WDMA) protocol operates as follows. Each of the \( N \) users is assigned a minislot out of the \( N \) control minislots per slot. For example, station \( i \) is assigned the \( i \)th minislot. A station wishing to transmit a data packet first sends a control minipacket in its assigned slot announcing the address of its intended receiver. All stations monitor the control wavelength constantly with their fixed receivers. When a station sees a minislot with its address in it, it knows there is a data packet intended for it in the next slot. The destination tunes its tunable receiver to the data wavelength of the sending station and receives the data packet.

If a stations sees that more than one data packet is destined for itself in the next slot, it chooses one to be successful by a predetermined algorithm. Thus, one packet destined for a designated receiver is always successful. The senders in such a situation which are not chosen to be successful must start the transmission process over again.
by waiting until another frame and announcing the intended destination's address in a minislot.

The overlap of control and data slots is similar to Sudhakar et al.'s Case 2. Here because there is a separate transmitter/receiver set fixed upon the control wavelength, there is no problem with such an overlap. Stations can be sending (or receiving) a data packet in one slot while simultaneously preparing to send the next data packet and listening for the location of the next incoming data.

5.1.2 Analysis

If the size restriction on the number of users of the network is bearable, the DT-WDMA protocol does allow for high throughput.

In the analysis the transmission procedure is slightly altered so that newly generated packets are favored for transmission over old ones which have failed. If a new packet arrives at each station every slot, then the transmissions are uncorrelated. The throughput analysis is conducted by finding the average number of different receivers chosen by the set of $N$ transmitters in a slot.

Let

$$I_k = \begin{cases} 1 & \text{if receiver } k \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

Then

$$P\{I_k = 0\} = \left(1 - \frac{1}{N - 1}\right)^{N-1}, k = 1...N$$

and

$$P\{I_k = 1\} = 1 - \left(1 - \frac{1}{N - 1}\right)^{N-1}, k = 1...N.$$ 

The total throughput per slot can be calculated from

$$S_{tot} = E \left[ \sum_{k=1}^{N} I_k \right]$$

$$S_{tot} = \sum_{k=1}^{N} E[I_k]$$

92
\[ S_{tot} = \sum_{k=1}^{N} P\{I_k = 1\} \]

\[ S_{tot} = N \left[ 1 - \left( 1 - \frac{1}{N - 1} \right)^{N-1} \right] \]

The throughput per station per slot is

\[ S = 1 - \left( 1 - \frac{1}{N - 1} \right)^{N-1} \]

\[ S \approx 1 - e^{-1} \text{ as } N \to \infty. \]

Therefore, for a large \( N \), Chen et al. arrive at a maximum total network throughput per slot of

\[ S_{tot} = (1 - e^{-1}) N \approx 0.63 N. \]

As there is no contention for control minislots, the only limit on throughput is causes by destination conflicts. Chen et al. handle these as do Lu and Kleinrock [12]: among successful reservations, one transmitter is picked by the receiver in contention to be heard and the rest must retry later.

This protocol offers higher maximum throughput than do the others seen so far. The elimination of Aloha contention on the control channel combined with the assurance that one packet destined to each receiver will be successful each slot raises the throughput of the system. The throughput becomes only dependent upon the number of users (which corresponds to the number of data wavelengths) of the system.
5.2 Beyond Single Control Channels

While most protocols have included a control channel as a natural way to easily facilitate network management, there are a new generation of protocols that eliminate the use of a single control channel. There are several reasons for seeking to do so.

5.2.1 Drawbacks of a Single Control Channel

Firstly, the amount of information that must be disseminated over a single control channel is quite high. This is evident in the preceding protocols. Therefore, the processing requirements at each station needed to handle the information on the control wavelength are quite large. Each station would need to be equipped to process large amounts of information to constantly know the status of the network. Stations would be required to know from this the state of the entire network, and from this information would have to be able to further extrapolate the information relevant to their own transmissions very quickly.

Stations are not really required to know the status of every network user in order to transmit and receive their own data. Therefore, they could get by easier and with much less processing required if they were only to receive control information concerning their own operations.

Another reason for seeking to avoid one control channel systems is the problem of collisions on that control channel. The performances of many protocols, such as those of [13, 16], are limited by the probability of control packets colliding on the one channel as has been shown. Thus, it would seem that a logical way of alleviating this bottleneck would be to get rid of the single wavelength design.

Two papers propose eliminating the single control channel. They do so in the two most obvious next steps. The first seeks to eliminate the control channel altogether. The second takes the opposite approach and adds multiple control channels to the network. Both limit the size of the network by using preassigned wavelengths per station, much as Chen et al. did in their TDM-based scheme [3].

94
5.3 Semaan and Humblet: Elimination of Control Channel

The first paper seeking to avoid the single control channel architecture proposes the elimination of a control channel altogether [15]. The idea motivating this work is to embed the control channel within the data channels. The architecture of the network is as follows.

Each station is assigned to a fixed, independent wavelength. Thus, the number of users of the network is limited by the number of wavelengths available. Each station is equipped with a single transmitter and receiver, both tunable to all wavelengths. The system is slotted, with slots multiplexed into frames. Each frame consists of \(N + 5\) slots, where the first five are control slots – three of which are of no concern here – and the following \(N\) are for data transmission. Data slots are allowed to be of two types: reserved or free. Reserved slots are only used for isochronous traffic, while free slots may be accessed by any station for datagram traffic.

The system operates on cycles, as do all systems which require the sender to receive acknowledgment of a reserved transmission time and channel. Because of propagation delays in the network, stations must allow for control information to reach all nodes before transmitting the corresponding data. Thus, the system operates on a cycle of \(R\) frames, where \(R\) represents the smallest integer larger than \(1 + (\text{maximum propagation delay})/(\text{frame duration})\). Frames are numbered modulo \(R\). Control slots then carry information concerning the given frame number in the next cycle.

Each station transmits its control slots on its dedicated wavelength. Data, how-
ever, is transmitted on the wavelength of the intended receiver. In each frame, a station first tunes to the wavelength of a station which it intends to send data to in order to hear which data slots that receiving station has available in the next cycle. This is an option only if the system is used exclusively for datagram traffic or if some slots are held for datagrams only. Then all stations tune back to their assigned wavelength with their receivers to gather incoming traffic while using their transmitters to tune to intended receivers’ wavelengths to send them data.

5.3.1 Types of Traffic

This protocol allows for three kinds of data to be sent: isochronous, datagram, and bulky data. Whenever data or a connection request is received, the receiving node will announce the fact in its ACK slot, which is one of the control slots. Because there is no control channel, there is no other way to deliver the reception information the the network. With a control channel, all users know who is transmitting when and where every cycle. Now, concerned users who wish to know whether their data has been received must listen to the destination’s ACK slot.

It should be noted that this protocol has been designed to implement isochronous data primarily. This protocol is not really suited to handle only datagram traffic. However, as the analysis in this thesis is based upon datagram traffic, that is all that will be considered here.

5.3.2 Datagram Traffic

In order to send datagram traffic to another station, the sender first listens to the control slots of the intended receiver to find out which slots are available for use in the upcoming cycle. The sender then matches these with the slots in which it has not scheduled any other transmissions. Slots which are open for both source and destination are called the useful slots. After determining the useful slots, the source will randomly choose one in which to send the data packet. Thus, the source is using slotted Aloha on the useful slots.
It should be noted that this is only an issue if the network is handling isochronous traffic in addition to datagrams. If there is only datagram traffic, then the sender does not have to listen to the status of anything. Data packets are simply sent in any slot at any time. No control information is processed prior to transmission. Thus, the network has been reduced to the simplest Aloha network.

As usual, if more than one station sends a packet in that slot there is a collision and all are lost. Stations must read the ACK slot of the receiver to determine the fate of their data. ACK slots reach the sender two cycles after the data has been sent. Therefore, the minimum retransmission delay in case of collision is two cycles. Because only one Status slot and one ACK slot may be read by a station per frame, only one datagram can be sent per frame.

The throughput for datagrams is thus the standard for slotted Aloha,

\[ S_{u.s.} = Ge^{-G} \]

per useful slot. Thus, the maximum throughput is the familiar \( \frac{1}{e} \) per useful slot per frame. However, because there can be at most one datagram sent per user per frame, the throughput per user is less than one slot per frame. Because the number of users equals the number of channels, throughput is less than \( \frac{1}{N} \).

5.4 Humblet et al.: Multiple Control Channels

The other approach to eliminating the single control wavelength is to have multiple control channels [7]. In the basic form of this protocol, each station is equipped with one tunable transmitter and receiver pair and one fixed transmitter and receiver pair. Additionally, each station is given two dedicated wavelengths: one for control information and one for data. Thus, the maximum size of the network is limited to one half the number of available wavelengths. This is the most severe size bottleneck seen in all the protocols.

In order to directly compare the performance of this protocol with its companion
by Semaan, the modified network architecture in which each station only has one tunable transmitter and receiver is considered [7]. Obviously, cutting the number of transmitter/receiver pairs has the advantage of saving each node the cost of excess hardware. However, the performance of the protocol suffers. Now, like so many other protocols, this has the problem of a station's not being able to transmit data while processing control information.

Once the hardware is reduced, the architecture is similar the that of [15]. Again, the system is synchronized and slotted, with slots compiled into frames. Frames consist of \( m + n + 1 \) slots. The \( m \) slots are of length one a while the \( n \) slots are of length \( L, L > 1 \), so as not to cut too severely into the amount of time the stations can transmit data. The control portion of the frame wouldn't take up too much time then.

### 5.4.1 Types of Traffic

As with the Semaan paper, the protocol here is designed to be of use mainly for isochronous data transfer. Here there will be a control packet sent prior to each data transfer. However, there will be no need to look first for slots which the receiver has available.

In the first \( m \) slots of a frame, the receivers of each station are fixed upon their control wavelengths. When stations are involved in isochronous data transfer, every station in the network that has a connection set up with a particular user is assigned one of these \( m \) slots. Thus, as many as \( m \) stations may have connections set up with a user at once. During the next \( n + 1 \) slots, the transmitter of each station is fixed upon their assigned data wavelength. The \( n \) slots are used to send data packets, while the 1 extra slot is used to transmit the status of the station. The status slot tells the idle slots a user has among its \( m \) short, fixed receiver slots. So, up to \( m \) stations may have connections set up with a user, \( n \) of which may transmit each frame. Usually \( n \leq m \).
<table>
<thead>
<tr>
<th>Hardware</th>
<th>Wavelength</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT</td>
<td>data</td>
<td>n+1 slots. n = data, 1 = status</td>
</tr>
<tr>
<td>TT</td>
<td>–</td>
<td>Send con. req.'s and what FT slot data's in.</td>
</tr>
<tr>
<td>FR</td>
<td>control</td>
<td>m slots. Receive TT announcements.</td>
</tr>
<tr>
<td>TR</td>
<td>–</td>
<td>Tune to FT's to read data, status</td>
</tr>
</tbody>
</table>

Table 5.1: Hardware and Wavelength Uses for Control and Data Transmission

5.4.2 Datagram Traffic

In order for one station to send a datagram to another, two steps are involved. First, the sender picks a slot at random from the available ones out of the $n$ total in its own transmitter. The sender announces this slot number in one of the idle slots in the destination's $m$ control slots (determined from the status slot of the destination)$^1$. Then the sender transmits the datagram in the slot it chose out of the $n$ available during the data phase of the frame.

The data may not get to the destination if another station uses the same idle slot out of the $m$ in the destination's "control" portion of the frame or if another packet is sent the the receiver out of the $n$ in the "data" portion of the frame. When isochronous traffic is also being sent, if there are no open slots out of the $m$ in the destination's "control" frame, there can be no datagrams sent. As Semaan and Humblet did in [15], this protocol could allow some slots to be held open for asynchronous traffic as "free" slots.

The performance of datagram traffic for this protocol shares a feature with that of Chen et al.'s DT-WDMA protocol in [3]. The similarity is that, following successful control packets, if more than one data packet is destined for a particular receiver in a slot, one of them will always be chosen to succeed and the rest will have to retransmit later. The protocol is basically slotted Aloha on the control channel followed by transmission on a fixed wavelength and slot for the data. A success on the control channel does not guarantee success on the data channel. Failure on the control channel does insure that the data will be lost.

Letting $P$ be the probability that someone tries to transmit in a control slot, $P^n$ is

---

$^1$Slots may not be idle if isochronous traffic is also being permitted

99
the probability that someone will try for the \( i \)th data slot. The probability of success per slot per channel is

\[
1 - \left( 1 - n \cdot \frac{P}{n} \left( 1 - \frac{P}{n} \right)^{n-1} \right)^m.
\]

If all users are busy, \( P = 1 \) and this can be approximated by

\[
P(\text{success/slot/channel}) \approx 1 - \left( 1 - e^{-1} \right)^m.
\]

Taking into account that the \( m \) control slots are of length 1 and the \( n \) data slots of length \( L \), the network throughput becomes

\[
S = \frac{nL}{nL + m} \left( 1 - \left( 1 - e^{-1} \right)^m \right).
\]

This can be very close to one. Figure 5-2 illustrates this.

Solving this equation for \( m \), one discovers that the optimal \( m \) is a function of \( nL \). For a pure datagram system, this makes sense. If there are few data slots, there is no need to have a large number of control slots to make reservations for them. Conversely, if there are many data slots, it would make sense to have a lot of control slots so that there could be more reservations made. Of course, there may be problems filling the data slots in any case as stations are restricted to one datagram per cycle.

5.5 Comparison of Two Protocols: Datagram Traffic

In the data transfer phase of each frame in the two protocols, there is a fundamental difference. In Semaan [15], the receiver is fixed on a given wavelength and the senders must tune their transmitters to this. To get datagrams through, then, it is only necessary that they not collide in the slot they are sent in on the receiver's wavelength. Therefore, performance is limited by the Slotted Aloha operating on the channel.

In Humblet et al. [7] the receivers are not fixed on a wavelength during data
Figure 5-2: Throughput per Channel of Humblet et al. versus $m$ and $nL$
exchange. The receivers are the ones that are fixed. Thus, the destination of a packet must know the specifics of when and where it is being sent from in order to tune in to receive it, which must be scheduled within one round trip delay before the data transmission must take place. This protocol is the more complicated of the two. However, the throughput possible with is far higher than that of Semaan. Where Semaan is limited to a maximum throughput of $\frac{1}{2}$, Humblet et al. can achieve a throughput very close to one per data channel. It must be noted, however, that both of these protocols were designed to handle isochronous data transfers. Datagrams are not an emphasis.

5.5.1 The Use of Acknowledgments

Both of these protocols rely on the use of acknowledgments. In protocols where all control information is broadcast to all users via a single control channel, ACKs are not necessary. Each user can see for itself the outcome of any data or control packet by simply consulting the control channel and looking for the expected outcome. Once that single broadcast channel is eliminated, however, ACKs become necessary to the operation of the protocol.

5.5.2 Semaan and Humblet’s Acknowledgments

In [15], where no control channel is present, the ACK slot is of major importance in each frame. Because frames are numbered modulo $R$ in cycles, where $R$ is the propagation delay of the network, each ACK slot serves to acknowledge transmissions occurring in the previous cycle. There is a two cycle delay incurred in waiting for an ACK after sending a packet. This makes the delay of this protocol higher than that of other Aloha-based protocols where there is a broadcast control channel present. A sender must retain even a successful data packet for two whole cycles while waiting to see if the transmission was successful or not.
5.5.3 Humblet et al.'s Acknowledgments

Where there are multiple control channels – each station having its own – ACKs are also necessary [7]. Here a station only receives control information that is pertinent to its own transmissions. Moreover, for datagram traffic, each user only receives on its control wavelength the notifications of datagrams that are heading towards it in the next frame. To find out whether a datagram which it has sent has arrived successfully at the intended destination, a user must consult the status slot of the intended receiver in the next frame following transmission.

Because the status slots are sent while the stations are operating with their transmitters fixed upon their assigned data wavelengths, this means that the sender of the datagram must tune its receiver to the receiving station and listen to the status slot. While it is doing this, it cannot be listening to the status slots of any other stations. Thus, the sender is tied up a further cycle listening to the ACK of its last datagram before it can consider sending another.

This is not the case with the no control channel protocol. There, the “status” and “ACK” are two different slots. In each frame duration, assuming the lasers are tunable fast enough, a station can read the Status slot of one user and the ACK slot of another. Therefore, a sender can be preparing to transmit the next datagram in a cycle while checking to see if the last one was received. This cuts the delay and degree of time “wasted” for a transmitter to wait on and ACK.
Chapter 6

Expanding Protocols which Limit the Number of Users to N

As the past several chapters have illustrated, there are many proposed protocols which require each station to be assigned a unique wavelength to transmit data on. Humblet et al. take this a step further in [7] by requiring each station to be assigned two unique wavelengths.

Many of these protocols give good throughput and delay performance. However, they are limited in their possible applications by the size restrictions on the user population which they impose. Protocols which allow the user population to be greater than the number of WDM channels available are far more flexible. It is much easier to expand or shrink the number of users of the network. In situations where more and more users may need access to the network as it evolves, these are the protocols to use.

However, many of these protocols are limited by the Aloha contention they use to allow stations to gain access to the data channels. Aloha-based systems will be restricted in throughput by the $\frac{1}{e}$ probability of success in each Aloha slot. Additionally, all such protocols must use a control channel in the network on which all stations must contend for data transmission times, make reservations, announce new entries in transmission queues, etc. The option of eliminating the control channel is not available in these networks.
Of the protocols which limit the size of the user population, only Humblet et al. mention the possibility of allowing more users onto the network than there are dedicated channels available for. This would require the introduction of some element of contention on the network. Stations would no longer have reserved access to any channels, but would have to contend for them. However, rather than every user contending for every wavelength, within these protocols, modifications would allow for a small number of users to contend for one data wavelength.

This chapter investigates what alterations must be made to the protocols that use dedicated wavelengths to enable them to support larger user populations. Each protocol will experience a degradation in performance from this action. Some will adapt more readily than others to the increased number of users.

6.1 The DT-WDMA Protocol

The protocol of Chen et al. in [3] was shown earlier to be capable of achieving a maximum network throughput per slot of $0.63N$ as $N \to \infty$. Access was gained by each station announcing the destination of its data in a preassigned control channel minislot and then transmitting the data on its data wavelength. However, the use of assigned, unique data channels limited the size of the network to $N$ stations, where $N$ is the number of data wavelengths.

If there were $M$ user stations, $M > N$, changes would have to be made to the protocol. Let the users be grouped into $N$ "clusters" of $\frac{M}{N} = X$ users each. If the $N$ control minislots are still used, then each of the $X$ stations in a cluster would share both the minislot and the cluster's assigned data wavelength. One way to allow users to send data would be to have the stations which share a cluster compete for the control minislot via Aloha contention. Whichever user had a successful control minipacket would gain control of the data channel for the next slot. However, each station contending for the minislot would have to wait one round trip propagation delay before finding out whether its control minipacket was successful or not. This delay, coupled with the delay incurred from having to retransmit unsuccessful control
minipackets, would be significant. Additionally, the throughput of the network would come to be limited to that of Slotted Aloha.

An alternative idea requires that each of the $X$ stations in a cluster take turns using the control minislot and data channel. Here, an arriving packet waiting for transmission may have to wait up to $X - 1$ slots before the other users in the cluster have had their turns. Once it is a station's turn to transmit, though, the DT-WDMA protocol operates as it did originally. The maximum throughput possible would not be lowered by this access strategy. Situations can be conceived where it would even increase as collisions at clusters allowed for less collisions between actual nodes.

### 6.2 The Chen and Yum Scheduling Protocol

The Chen and Yum protocol can also be easily expanded to handle more than $N$ users. The maximum total network throughput of $N$ packets per slot can still be achieved. This makes this protocol remain highly desirable.

Again the cluster idea is used. To each TDM time slot on the control channel there is assigned $X = \frac{M}{N}$ users. Rather than one station announcing its new packet arrivals in the TDM minislot each frame, the $X$ stations in a cluster share the minislot, by Aloha contention. This adds slightly to the time between a packet arriving at a station for transmission and its presence being announced to the network.

The $NxN$ matrix structure of the algorithm proposed by Chen and Yum can still be used, provided that control minipackets contain the intended packet destination address, the sender's address, and the clusters to which the stations belongs. Transmissions will be scheduled to maximize the number of clusters which transmit per slot. From each cluster at most one station can use the cluster's dedicated wavelength at a time. If the restriction is imposed that each scheduled receiver must be in a separate cluster, then the $NxN$ matrix algorithm can be used to schedule transmission from cluster to cluster, rather than user to user as before. The throughput would remain the same as in the original protocol.

The scheduling algorithm could be altered to remove the restriction that all re-
receivers must be in separate clusters. This would not greatly improve matters though. Chen and Yum's protocol eliminates both transmitter and receiver collisions, neither of which is a great problem if the user population is large. Chen and Yum achieve a throughput very close to one. Clusters may lower this a little, but not very much.

Another variation of the protocol would be to expand the matrix structure to an $M \times M$ matrix. This would allow more flexibility in choosing which users are allowed to transmit. There would not be a "one user per cluster" restriction any longer. This may prove to be of great use if the amount of traffic transmitted is not uniform for all users.

Altering the protocol in either of these two manners will increase the delay of packets. There will be a bigger delay before new arrivals are announced to the network. Increasing the number of TDM control minislots could improve this. There will be a further increase in delay for packets waiting to be scheduled for transmission. It stands to reason that if there are more packets waiting to be sent, there will be a longer wait until a particular packet is scheduled to be sent. However, there will still be a number of arrivals announced each minislot and still be up to $N$ data packets sent each slot. The overall performance of the network does not greatly degrade with the increase in user population.

Another obvious way to alter this protocol would be to implement a $M \times M$ matrix structure from which a $N \times N$ transmission matrix would be found. This would have to be done so that it ensured that all stations have a fair chance at being selected to transmit.

6.3 Problems With Having No Control Channel

One protocol that does not adapt readily to having more than $N$ users is the no control channel protocol of Semaan and Humblet in [15]. With this protocol, each station is required to transmit its control slots on its dedicated wavelength at the start of each frame and follow this by sending data on the wavelengths of the intended receivers. A station cannot send data to an intended destination without first reading the control
slots of that destination to see which slots in a frame are open and available for datagrams to be sent in.

This leads to problems when there are $M > N$ users of the network while only $N$ channels for data. Now each user no longer has a dedicated wavelength available to announce its control slots and receive data packets on. It is not clear how to upgrade the number of users in this case. The cluster solution does not produce obvious results as it has with the previous protocols. There is a great problem in determining which users have the use of the dedicated wavelength to receive data and announce control information on at what time. Perhaps this could be handled by having one station be in charge at each cluster.

If, for example, there were a longer control period in which all $M$ users were allowed to broadcast their control slots, how would the wavelengths for data reception be allocated? If, on the other hand, stations took turns using the dedicated wavelength to send control information and receive data, stations wishing to send data to a user would have to wait many frames until that user was allowed to broadcast its control slots and then send data in the same frame one round-trip propagation delay later.

The protocol grows cumbersome rather quickly with the addition of more users. The increase in delay and in the amount of processing each station would have to do would have to be investigated very closely before modifying this protocol would pay dividends.

### 6.4 Reducing the Number of Required Channels in the Multi-Control Channel Approach

Humblet et al. realize that their protocol of [7] has a problem in that it limits the number of users to be $\frac{N}{2}$ by requiring each user to have use of two dedicated channels. They offer two suggestions to mitigate this problem.

First it is suggested that the number of data channels required could be lowered by using a static TDM assignment of control channel time slots. However, TDM
requires that the number of users of the network remain still fixed.

Next, to cut the number of control channels required, the cluster approach is again employed. Using the protocols original two transmitter/two receiver configuration, each fixed transmitter would remain on a fixed wavelength while $\frac{N}{M}$ fixed receivers shared a wavelength. This would make the total number of wavelengths needed be $N + M$. Using the cluster approach, up to $M$ users at a time could have connection access to a whole cluster. However, this approach taken alone would still limit the number of users to be less than $N$.

This protocol is very difficult to expand to a large number of users. If one clusters the users during the control portion of the frame \(^1\) the probability of there being open slots out of the $m$ allocated to the cluster lessens as the cluster size increases. Thus, it becomes harder for stations wishing to connect with the cluster to find open slots in which to transmit datagrams. This could lower throughput significantly.

Throughput would also be affected by the adoption of a static assignment of control channel time slots designed to reduce the number of data channels required.

This protocol functions at its best when being used with its full complement of two dedicated channels per station.

\(^1\) assuming here that the single transmitter/receiver architecture is used
Chapter 7

Conclusion

There are many different ways to access an optical star network. Proposed protocols vary in their complexity and their performance. It is possible to see by tracing their evolution that there are ideas which are clearly more effective than others.

The first set of protocols investigated in this thesis are based upon Aloha contention to access a control channel by which a sender may alert a potential receiver that there is traffic on the way. These protocols are all limited in their control channel throughput by the maximum throughput of Slotted Aloha, $\frac{1}{4}$. It can be seen that slotting the data channel improves performance. The best of the Aloha contention protocols is Jeon and Un's BRP which allows a per channel throughput bounded by $\min \left[ \frac{L}{N_e}, 1 \right]$. The control channel performance is only a limiting factor until the number of control minislots is so large that the number of successful reservations exceeds the number of data channels.

The next improvement which stems from the BRP is the introduction of scheduling algorithms. These allow all uncollided control packets to be buffered rather than thrown away. By placing all of the uncollided control packets into a queue, a throughput of exactly $\min \left[ \frac{L}{N_e}, 1 \right]$ per channel can be achieved.

Using a scheduling algorithm designed to maximize network throughput as Chen and Yum have done is a way to achieve a throughput very close to one packet on every data channel. Although this protocols has been designed for the number of users to equal the number of data channels, it can be modified to handle a larger user
population.

Other scheduling algorithms are not so successful. Jia and Mukherjee's RCA protocol invokes a great deal of complexity to solve a very slight problem. The probabilistic approach of Chipalkatti et al. requires so much control information to be processed and sent that it is infeasible.

A third class of protocols requires that each user have dedicated control or data wavelengths on which to send data following an announcement that it is coming. Chen et al.'s expansion of TDM provides a throughput per channel of \((1 - e^{-1})\). The key feature here is that if more than one successful control packet announces data in the next frame destined for the same receiver, the receiver will choose one of the data packets to be successful. They are not all lost due to a receiver collision as in other protocols.

The final innovation has been a search to get away from the single broadcast control channel architecture. While neither of the two protocols investigated here were designed for datagram traffic specifically, the multiple control channel protocol of Humblet et al. provides a per data channel throughput which can be made very close to one. However, the number of users of the network must always be restricted by the use of both assigned unique data and control wavelengths by each user.

The protocol of Semaan which uses no control channel breaks down when only datagrams are transmitted and becomes a simple Slotted Aloha system.

Therefore it is possible to achieve a per channel throughput very close to one by several methods. Queuing and probabilistic scheduling require a more complex network, with issues such as how to initialize new users and how to handle errors on the control channel being raised. The multiple control channel approach requires a limitation in the number of users by requiring multiple control channels.

The least complex way to achieve a per channel throughput very close to one is to use the BRP proposed by Jeon and Un. The throughput here is almost the same as that of the more complex queued protocol. If the underlying Aloha guiding the control channel is stabilized by reducing the offered load as the number of backlogged nodes increases, then the delay of the BRP will not blow up to infinity. Thus its delay
performance is better for large $\frac{L}{N}$ than that of the queued system.
Bibliography


