Supporting Quality of Service in the Head-End of Hybrid Fiber-Coaxial Networks

by

Justin Liu

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1996

© Justin Liu, MCMXCVI. All rights reserved.

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

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
MAR 21 1997
LIBRARIES

Author . . . . . .
Department of Electrical Engineering and Computer Science
August 26, 1996

Certified by . . . . . .
John T. Wroclawski
Research Scientist
Thesis Supervisor

Accepted by . . . . . .
F. R. Morgenthaler
Chairman, Departmental Committee on Graduate Theses

Supporting Quality of Service in the Head-End of Hybrid Fiber-Coaxial Networks

by

Justin Liu

Submitted to the Department of Electrical Engineering and Computer Science
on August 26, 1996, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Many cable operators have begun the process of upgrading their cable systems in anticipation of delivering a wide range of interactive digital services to the home. These newer hybrid fiber-coaxial (HFC) networks can take advantage of unused bandwidth in broadcast cable television (CATV) and allow for an inexpensive and simple way for home users to have access to many broadband digital applications. The current trend toward more demanding network applications means that these HFC networks will need to support quality of service (QoS) control mechanisms which allocate network resources and can offer users certain network performance guarantees. The unique topology and characteristics of HFC networks present several interesting challenges for controlling and managing access to the system, and solutions developed for other kinds of networks may not be readily applicable. This paper explores techniques to understand these challenges and meet them with possible solutions. The primary goal will be to devise, study, and implement algorithms which demonstrate that support for QoS mechanisms, such as different classes (or priorities) of service, can be implemented with changes only to the head-end of these HFC systems. This solution offers flexibility in the design and distribution of cable modems by making this restriction.

Thesis Supervisor: John T. Wroclawski
Title: Research Scientist

Acknowledgments

Several people were instrumental in the development of this document and I would like to take this opportunity to share the blame.

First, I would also to acknowledge the people and the resources of the Communications Management team in the Broadband Information Systems Lab at Hewlett-Packard Laboratories in Palo Alto, CA. Specifically, I would like to recognize Srinivas Ramanathan, for his insight and guidance; Ed Perry, for his unwavering support; and Reuven Cohen, from whom I borrowed much of the code for the simulations used in this project.

I would also like to recognize my faculty advisor, John Wroclawski, who found time in his busy schedule to provide my project with supervision and to lend his critical eye to my work.

Finally, I would like to thank my friends and family, most importantly my parents and my brother, who survived endless complaining and gave me the emotional and spiritual support (not to mention considerable financial support) I needed to see this through to completion.

And now for something completely different.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

The unprecedented and explosive growth of the Internet and the World Wide Web has pushed wide area distributed computing beyond the confines of academic and government research and into the mainstream and the media spotlight. This expansion, coupled with the increasing pervasiveness of high-bandwidth, real-time multimedia network applications, has strained the performance and capacity of current networks to their limits, and there is clearly a need for fundamental changes in the network infrastructure. Applications are becoming increasingly complex and more demanding of all computing resources, especially those of the network, and the consumption of computing resources is likely to continue to outpace the ability to supply them. While plans to install bigger and faster network technologies, such as ATM and high-capacity optical links, may (or may not) eventually relieve this strain, other alternatives will have to be pursued at least for short and mid term solutions.

A necessary first step in bringing broadband interactive digital services to the home is to provide physical access points where users can connect to larger networks. Systems currently being used to deliver broadcast cable television (CATV) programming have excess bandwidth which can be reclaimed for this purpose, and since these cable plants are already widely installed and available at most homes, the costs associated with deploying network services can be kept to a minimum ([6, 18]). However,

the unique topological and physical characteristics of cable networks present new and different challenges from those encountered in more traditional types of networks. Techniques and solutions developed for other systems may not be directly applicable, although it is hoped that some of the lessons learned in designing and deploying other network technologies will be useful in the cable networks as well.

In addition to providing users with a physical access point, the next generation of networks may also have to provide network access control and management mechanisms in order to ensure users of certain guarantees of network performance or quality of service (QoS). There are many metrics which can be used to judge network performance, including concrete measures such as end-to-end delay, jitter (the variability of the delay), bandwidth (which will be used in this document, interchangeably with the word throughput, to refer to an amount of data transmitted per unit time, e.g., bytes per second), utilization, and buffer and memory usage, and also more subjective measures such as monetary value, ease of management and implementation, and end user satisfaction. Different applications and different users will have different requirements and bounds on these metrics and future networks should be able to provide appropriate resources to meet these different needs. Also important to consider when designing networks with QoS capabilities are the complexity and the flexibility of the design, since it is impossible to predict precisely both the future needs of users and applications and changes in the available technology.

This paper examines these two ideas in greater detail and presents solutions for integrating access control and management techniques into cable network technology for the purpose of introducing quality of service mechanisms. Specifically, this project studies a particular QoS feature and demonstrates the effectiveness of certain algorithms which could be implemented in the current generation of cable modem technology to support this enhancement, while maintaining the flexibility to allow for easy migration to other features and future generations of modems. The primary goal is to demonstrate the viability of a hypothesis which states that this control can be inserted with relatively minor changes to only the head-end of a hybrid fiber coaxial (HFC) network. I will also try to investigate the breadth of the range and capacity

of such controls, and then discuss some of the tradeoffs in designing such a system.

1.2 Outline

The general organization of this paper is as follows. Section 2 describes how a basic HFC network is set up and the characteristics of HFC networks and the access protocols which make them unique. Section 3 samples the variety of research in the general area of quality of service and then narrows the scope to specify the type of QoS which is studied in this project. Section 4 expands on the hypothesis stated in the previous paragraph and provides motivation as to why this hypothesis is both interesting and worthwhile from technical and practical standpoints. Section 5 outlines the details of the methodology in the experimentation and in the analysis of the results and some of the design choices I made. Then, sections 6, 7, and 8 each explain one of three different algorithms that were tested and describe results for each. Section 9 describes some attempts to refine and more fully understand aspects of the three algorithms tested earlier. And finally, section 10 contains overall analysis of the work, final concluding remarks, and a few lines about future directions for study.

Chapter 2

HFC Networks

This section gives more details as to what a hybrid fiber coaxial network consists of and explains the characteristics of HFC networks which are most important for this project.

2.1 HFC Overview

There is a growing desire from users to have access to a variety of broadband interactive digital applications from their homes, and while ideas such as fiber to the curb/home (FTTC/H) promise "infinite" bandwidth to the home, they are still potentially decades away from universal deployment. The general availability of cable systems make them a practical and economical first step in the delivery of broadband digital services to homes. Cable networks have long been used to deliver broadcast television to homes, but bandwidth which is not being used for television can be reclaimed and used to provide users with access to interactive digital services. Specifically, in addition to any unused channels in the frequency spectrum between 54 MHz and 450 MHz which is currently reserved for analog CATV broadcasts, the frequencies outside this range can also be made available for transmitting and receiving digital data. However, there are special features and characteristics of cable networks which will make this a challenging task.

While various cable protocols are still being proposed to standards committees

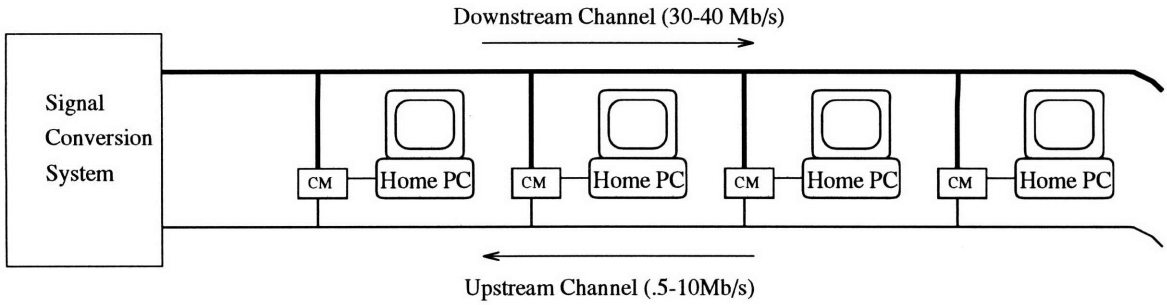


Figure 2-1: Cable Network Model

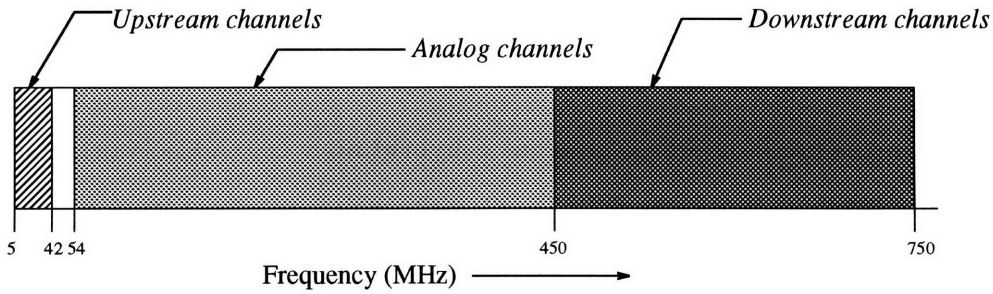


Figure 2-2: Spectrum allocation

(IEEE 802.14 [3]), they share important characteristics. First, communication over cable networks is directional so there are "downstream" (from the head-end to the home) and "upstream" (from the home to the head-end) channels which operate in different frequency spectra and can be thought of as separate and distinct (Figure 2-1), even though they coexist on the same physical cable. The spectrum between 450 MHz and 750 MHz will be reserved for downstream transmissions and the spectrum between 5 MHz and 42 MHz will be reserved for upstream transmissions (figure 2-2). These frequency bands will be further subdivided into smaller channels, 6 MHz wide for downstream and 0.5–5 MHz wide for upstream, in order to improve transmission efficiency. Each of these channels will be capable of supporting approximately 30–40 Mb/s in the downstream direction and about 0.5–10 Mb/s in the upstream direction, depending on the modulation and encoding techniques used. This approach to the division and channelization of the frequency spectrum is the most common one among the current generation of cable modems, however other schemes do exist (see [1, 7, 15]).

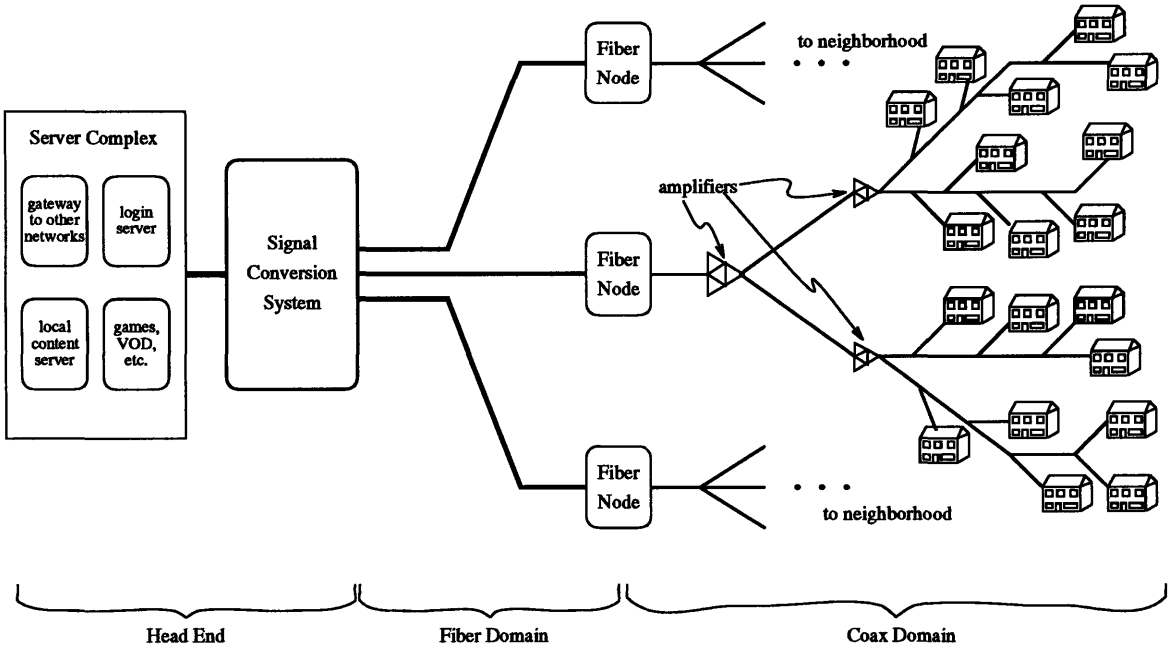


Figure 2-3: An HFC Network

2.2 A Basic HFC Network

Most cable operators have either already upgraded or will soon upgrade their systems to HFC networks, which allow older all-coax setups to span greater distances by the insertion of optical fiber, which is more reliable than its electrical counterpart. A typical setup is shown in Figure 2-3. At the root of the tree-like structure characteristic of an HFC network is the head-end, maintained by the cable service provider, where content distributed over the network originates. The functions of the head-end include the (re)broadcasting of regular cable television as well as possibly providing servers for user login and authentication, content of interest to the local community, video on demand or other interactive media, and gateways to other networks and the general Internet. Also at this root node is a Signal Conversion System (SCS) which manages the access to and from the HFC link and communicates between the HFC network and the servers. In this document, the term head-end will be used interchangeably with the term SCS to refer specifically to this root node of an HFC system which has control over access to the HFC link.

The SCS then sends this data out from the head-end over multiple high speed

fiber links to fiber nodes located within each neighborhood. This is the optical fiber domain of the HFC network. At each of these fiber nodes the optical signal is converted into an electrical signal which is then sent out over the coaxial cable links. This coax domain covers approximately the last twenty percent of the distance to the homes. The electrical signals are boosted and branched by amplifiers placed between the fiber nodes and the end users' homes. Traditionally these amplifiers have been unidirectional for CATV, but in order to support bidirectional traffic, they are replaced by bidirectional amplifiers. The fiber nodes also convert the electric signals being transmitted back to the head-end from the end user into optical signals which are usually multiplexed in time to allow the fiber nodes to share the optical link. Each fiber node could support services for up to 2,000 homes, assuming that about half (1,000) of those homes subscribe to broadcast cable television and then 20% of those subscribers (200) also subscribe to data services. A higher take rate would mean that cable operators would need to deploy more fiber nodes into the neighborhoods. Overall, an HFC network could span up to 35 miles and support about 2,000 homes per fiber node over about 50 fiber nodes for a total of up to 100,000 homes per head-end.

Inside each home (Figure 2-4), the coaxial cable is split and connected to televisions in the home, for normal broadcast TV viewing, and also to a cable modem (CM). The cable modem is then connected to the user's personal computer(s) much as other input/output devices, such as traditional data modems or printers, are connected (i.e., over Ethernet or a serial/parallel cable, directly on the internal bus, etc.). The cable modem acts as the interface between the coaxial cable network and the user's computer, and is capable of transmitting and receiving at several different frequencies or channels on the coaxial cable, allowing for the efficient use of the total available bandwidth by accessing different upstream and downstream channels. The behavior of the cable modem is specified by a medium access control (MAC) protocol which describes the communications protocol between the cable modem and the head-end over the HFC network. In this study, I will be using a generic cable MAC protocol, described below, so the techniques and results presented in the rest of this paper should be general enough to be applicable to a broad range of similar HFC

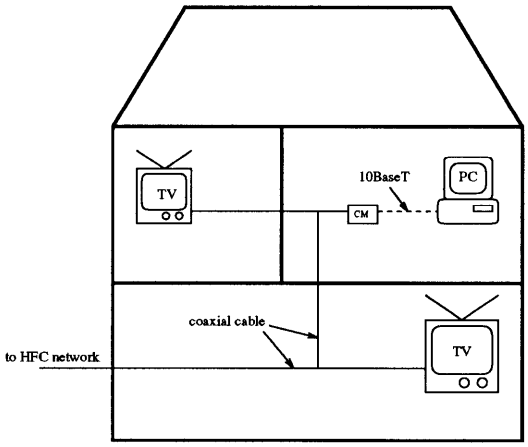


Figure 2-4: In the home

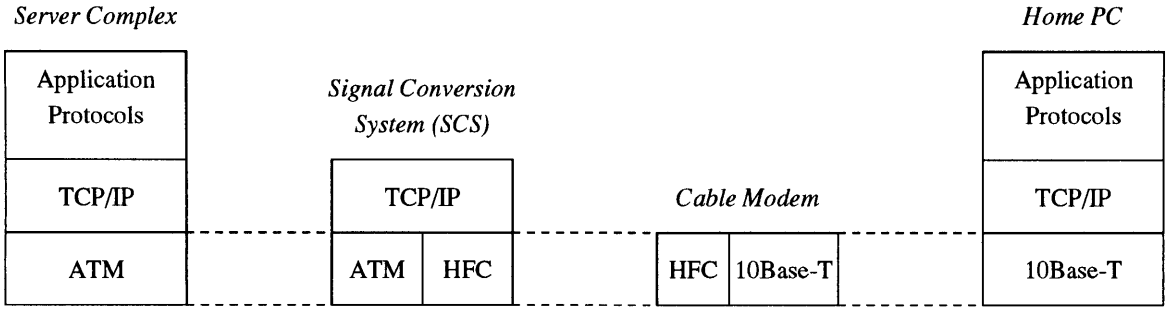


Figure 2-5: Protocol layering

MAC protocols.

Figure 2-5 shows a simplified picture of an HFC network from a protocol layering perspective. At the highest level, there are distributed applications which run the client component on the home PC and the server component in the server complex. The standard Transport Control Protocol/Internet Protocol (TCP/IP) is used to communicate between the home PC, the server complex and the Internet and any online service providers. The server complex communicates with the SCS over an ATM link (or possibly some other high speed interface) which then communicates with the cable modem using an HFC link, and which finally communicates with the home PC over a 10 Base-T Ethernet link. This paper focuses specifically on the HFC link between the SCS and the cable modem, but it is important to understand the context and the higher layers in the protocol stack which surround that link.

2.3 The MAC Protocol

This section describes the important features of the MAC protocol which is used in this study. This protocol is based in part on the work in [19] and shares certain relevant characteristics with a broad class of other HFC MAC protocols. The assumptions I make below serve to define this class.

First, I assume that the upstream and downstream channels are asymmetric in bandwidth (for example, typical values might be about 30 Mb/s available downstream and about 3 Mb/s available upstream). Since I intend to concentrate only on the more challenging upstream problem (see section 2.4), the only assumption I make about the downstream channel is that there will be sufficient bandwidth available in the downstream channel to distribute control messages for managing the upstream channel.

I also assume that the upstream channel is slotted (a typical slot size might be 64 bytes). The head-end marks upstream channel slots as either *reserved* for a particular modem, or *free* for any modem to contend in. When a cable modem that has a packet of data to send detects a free "contention" slot, it may try to send a request for a reservation in that slot. This reservation request contains the number of consecutive slots the modem requires to send its packet. Within some predetermined timeout period, the head-end must return an acknowledgment which informs the modem when it will be able to transmit that packet. The modem then waits for that scheduled time and sends the packet.

If the head-end does not acknowledge the reservation request within the timeout period, the modem assumes that the request transmission suffered a collision and runs a backoff algorithm. Many different backoff algorithms exist which could be employed, such as slotted Aloha, but the differences in the performance of the various backoff algorithms has only a small effect on the overall maximum throughput, especially as the average reservation length increases.[1] The only necessary assumption for this

[1] Recall that the maximum throughput S is related to the average reservation length σ and the maximum throughput each reservation slot S_r by: $S = \sigma/(\sigma + (1/(S_r - 1)))$

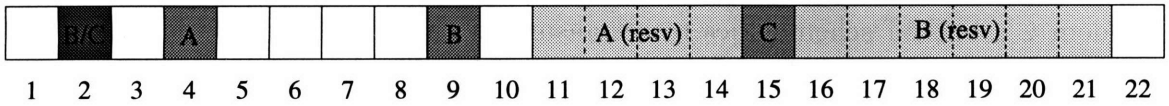


Figure 2-6: An example of upstream transmission

project is that there is some process in place which dictates how the modems resolve collisions and recontend.

Finally, in this system I assume that the MAC protocol is designed so that differences in the physical distances between the cable modems and the head-end do not affect access to the link. For example, in the MLAP (MAC Level Access Protocol [3]) implementation, this is accomplished by a ranging process that is performed when each cable modem powers on. During this ranging process, the head-end determines the round trip delay to/from the modem and uses this information, and the maximum round trip delay (which is a known network parameter) to calculate a round trip correction value. The cable modem then uses this value to adjust its transmission of each slot in such a way that the slots from all modems are exactly aligned. This means that modems which are closer than others do not get better (or worse) access to the network. As a result of this assumption, QoS mechanisms can be based on average performance and I can conclude that each modem will have a fair share of the average.

To demonstrate with a simple example, suppose there are three cable modems on a link, modems A, B, and C. Figure 2-6 shows a segment of the upstream channel, where each block is a slot and the blank slots are empty ones (i.e., available for contention). I have also assumed a round trip time equal to 6 slots in this particular example. At time 2, B and C both try to send in a contention slot and collide. The head-end detects the collision and does not acknowledge the requests and both modems time out at time 8 and execute their backoff algorithm. Meanwhile, A sends a reservation request for 4 slots at time 4. The head-end acknowledges the request and informs A that it can transmit starting from slot 11 (the first available slot, taking into account the downstream delay). B retransmits its request at time 9 and asks for 6 slots. This request is granted and the head-end informs B that it can transmit at time 16.

Modem C's backoff scheme tries to retransmit at time 11, but that slot is marked reserved and C must wait for an empty slot. A transmits its 4 slot packet from time 11 to time 14. C transmits in slot 15, which is empty. Since C only requires 1 slot it does not make any reservation. And finally, B transmits its packet from time 16 to time 21.

Note that in this particular example, the head-end scheduled each reservation in the first available slot. However, the protocol does not state that this behavior is required and as we will see later in this report, we will make use of this fact to enforce some control over the traffic. Remember that a modem cannot transmit while it is waiting either for an acknowledgment from the head-end or for the appropriate slot to transmit a previously granted reservation. Also, although some MAC protocols allow reservation requests to be "piggy-backed" onto other reservations, the protocol used in this paper does not allow this behavior. This means that each separate reservation request must successfully navigate the contention process, although multiple packets may be accumulated and share a single reservation request (up to a maximum reservation length). This is done to prevent having one user dominate the link. Finally, I assumed that there is no support for the fragmentation of packets, so if the head-end cannot allocate enough consecutive free slots to accommodate the entire request length, it must deny (send no acknowledgment in response to) that request.

2.4 HFC Challenges

As mentioned above, cable networks have distinct and separate upstream and downstream channels and as a consequence hosts connected along the same coaxial tree cannot directly hear the broadcasts of their neighbors and thus are unable to detect collisions on the upstream channel. This means that an external agent is required to coordinate the network traffic and detect the collisions. Fortunately, the head-end in the tree-and-branch topology is in an ideal position for such an agent and it will play the primary role in arbitrating and controlling access to the upstream channel. Also, since downstream communications are an example of a one-to-many broadcast, the

head-end can control and coordinate traffic along the downstream channel as well.

HFC networks also have a relatively high round trip delay times, where a round trip delay refers to the time it takes for a modem to send a packet up to the head-end and then have reply sent back in the downstream channel (including all packet processing time). This magnifies the problem of having to rely on the head-end to perform collision detection. This delay time is further increased by the addition of forward error correction (FEC) data in both the upstream and the downstream channels. FEC is necessary since the coaxial segment of an HFC network are susceptible to noise and relatively high error rates. The main problem encountered with this long round trip delay is that the head-end must schedule a packet for some time in the future without knowing what other requests it may receive in the interim period. That is, during the delay between the time when the head-end sends an acknowledgment to a cable modem and the time when that cable modem receives the acknowledgment, new reservation requests can arrive at the head-end.

It is important to keep in mind that the upstream and downstream channels are not symmetric; in fact, each downstream channel, due to a wider and less noisy frequency spectrum, may be able to carry an order of magnitude more data than each upstream channel, and there may also be many more downstream than upstream channel. Furthermore, in the downstream case there is only one transmitter on the link with multiple receivers ("one-to-many"). This case is relatively well understood and conventional priority scheduling disciplines designed for switching nodes (e.g. Weighted Fair Queuing, Class-Based Queuing, Delay Earliest Due Date) can be adopted in a relatively straightforward manner. A more challenging problem exists in controlling access, handling collisions, and scheduling traffic in the upstream channel. This thesis will focus on this problem. I will only assume that a fixed bandwidth channel in the downstream direction is available to distribute all the information necessary to fully control access to the upstream channel. This is a reasonable assumption given the greater capacity in the downstream direction.

Chapter 3

Quality of Service

Quality of service (QoS) has fallen into that class of terms which have become overused and overloaded and which encompass a wide variety of vastly different concepts. Any paper which proposes to discuss quality of service should include a clear definition of what QoS means in the particular context. This section starts with a brief overview of the breadth of QoS research in general, and then narrows the focus for this paper by defining the specific QoS delivery capabilities studied in this project.

3.1 QoS Survey

Quality of service today has come to mean different things to different people, but at its root it tries to describe the relationship between what kind of performance a particular user wants from a system and the performance that user is getting. In some sense, QoS represents an attempt to quantify "customer satisfaction," in concrete terms which could be used to form the basis of a business agreement. It applies to a number of different areas, including the ongoing research in the area of QoS in I/O subsystems and QoS in operating systems, but here the focus is on QoS in the network.

The Internet, as it was originally conceived, offered a simple point-to-point best-effort service. A broad spectrum of individuals and companies from all industries are joining the global network every day and there is a need to be able to distribute and

ration out limited network resources among the various users. The current framework, while robust and effective for applications such as ftp and email, has proved woefully inadequate for real-time applications such as networked video playback or teleconferencing. These network applications are becoming increasingly pervasive and more demanding on resources and the next stage in the evolution of the Internet clearly requires some changes to meet the changing needs. The challenge is to address this problem with a solution that is both flexible and cost-effective. QoS control is one such possible solution which attempts to deliver certain guarantees about network performance to different applications.

Applications such as ftp and email are *elastic*, which means that they always wait for the data to arrive and are relatively insensitive to and tolerant of delays and retransmitted packets. Hard real-time or *rigid* applications, such as remote robotics and control, on the other hand, require data to arrive reliably within certain timing bounds; data which arrives outside a given window in time are essentially useless. These applications also often have high bandwidth requirements which must be met by networks. There is also a large class of real-time applications that fall somewhere in the middle of the spectrum and can adapt and change their requirements in response to fluctuations in network performance parameters such as available bandwidth, delay, or jitter. These are referred to as *adaptive* applications and some examples are teleconferencing or networked video playback applications. A network should be able to provide service for all three of these types of applications which meet their needs appropriately.

There are several techniques which can be used to control and manage the flow of network traffic and accommodate different network resource needs. First, an admission control algorithm and a preemption service must be employed to prevent and respond to overload conditions where the network traffic exceeds the network's capacity. The former would permit a network to give a "busy" signal and deny service to certain users if it cannot support any new traffic. This is necessary in order to make any absolute guarantees of bandwidth or delay bounds. The latter would allow certain data packets to be dropped in the event that the network receives more

traffic than it can handle. It is likely that a solution would include elements of both techniques. For example, a network could conditionally admit certain new flows with the provision that they may be preempted later in whole or in part if the network becomes very congested.

A traffic scheduling algorithm is also necessary to meet timing guarantees. The goal of a scheduling algorithm is to allow multiple streams to share access to a single link while striving to meet delay bounds and bandwidth guarantees and to minimize the jitter seen by the applications. Several techniques, such as Delay Earliest Due Date (Delay EDD), Jitter Earliest Due Date (Jitter EDD), or Weighted Fair Queuing (WFQ), have been suggested for switch nodes and one-to-many links, however, it remains to be seen whether all or part of these methods can be easily adapted to a shared medium network such as a hybrid coaxial cable (HFC) network.

Another consideration is the difference between *guaranteed* ("hard") and *predictive* ("soft") service. A guaranteed service uses worst case estimates to make admission control and scheduling decisions and gives users absolute guarantees on performance. A predictive service uses statistical estimates to make decisions, thereby allowing networks to provision for some average data rate instead of peak data rates, albeit with the possibility that the guarantees may not always be met. Initial studies have shown that by allowing applications to request a soft guarantee, overall network utilization rises dramatically with little or no effect on delays or packet loss ([14]). Networks can provide a combination of the two services by giving guaranteed service to certain mission critical applications which require absolute bounds on performance and predictive service with statistical bounds for other less critical traffic.

And finally, since the Internet is composed of many heterogeneous networks, each with different capabilities and technologies, which are overseen and controlled by several separate entities, a full QoS implementation requires some method for ensuring that end-to-end performance guarantees can be made for applications which span the Internet. To this end, a signaling protocol such as the ReSerVation Protocol (RSVP, [4]) can be used to communicate resource needs of applications to all points along the network. RSVP is currently pending approval before standards committees, but even

after it is approved there will still be many details to work out for a real deployment of such a system.

It should be noted that an implicit assumption in implementing any of these techniques is that there will be a cost structure imposed whereby a user will pay more money for guaranteed service or lower delay bounds. As part of this pricing scheme, it may also be desirable to divide users into two or more classes (or priorities) where users in a higher class would pay more for better overall access to the network. The challenge here will be to differentiate quantitatively the service characteristics for the different classes while still maintaining a high level of network utilization and fair access for all users.

3.2 QoS in HFC

The study of QoS control in HFC networks is the subset of the end-to-end network QoS control problem dealing with that particular type of physical link. Cable modems are still a relatively new networking technology and there is much ongoing research investigating QoS control in HFC networks. The IEEE 802.14 working group is currently investigating several proposals for a MAC standard for HFC networks. While very little detailed work has been made public by this working group, none of the proposed protocols has complete support for multimedia applications and QoS specifications, although some may be extendible for such support.

PDQRAP (Prioritized Distributed Queuing Random Access Protocol [17, 24]) and its variants are one proposal for a distributed HFC protocol where each modem knows which slots are not reserved and can be used to send data in an immediate access mode. This protocol also has the addition of a prioritized contention system that allows higher priority traffic to have better access to the contention channel.

Other suggestions are more centralized with all state information stored at the head-end. For example, in the CPR (Centralized Priority Reservation [16, 19]) protocol, the head-end grants reservation requests to modems just prior to when the modem can start transmitting data, which allows lower priority requests to be de-

layed if a high priority request arrives at the head-end. CPR also supports constant bit rate traffic, such as voice, by allowing a periodic reservation. Another centralized protocol is ADAPt (Adaptive Digital Access Protocol [11]) which has a variable size frame that is divided into two parts, one for synchronous traffic and the other for asynchronous traffic. This system is designed to make it easy for users who already have reservations to continue to send data, especially in bursts, and its frame structure is designed to work well with ATM (Asynchronous Transfer Mode).

For this project, I will further restrict the scope to studying one particular form of QoS delivery capability. This specific case is one in which there are two distinct priority classes of users, which I will label *basic* and *premium*, that require different levels of service. Although in general multiple users within a household can share one single cable modem, here the class or priority label applies to each cable modem; i.e., all traffic from a single modem is classified as either basic or premium and there is no attempt in this study to further differentiate among traffic streams from the same modem.

As an example of how this might be applicable in the real world, we could imagine that there are casual users who mainly use the network to browse web pages or send and receive email. These users would subscribe to the basic service and might pay on the order of \$20–\$40 per month. On the other hand, a typical premium subscriber might be a work-from-home or telecommuting user and might be charged \$50–\$100 per month to have better access to the network. This is very similar to the business model used in broadcast cable television today, where some users pay extra to have access to "premium" channels.

Additionally, the two classes are still traditional "best effort" systems within each class and there is no guarantee made on a per modem basis. Instead, I rely on fairness in the link MAC and at the transport (TCP) layer to ensure that each modem will receive equitable communication access relative to the other modems in its class. What the premium user is expecting is a "better" best effort than the basic user in terms of various network performance measures which will be detailed in section 5. When the network load is low, more basic user traffic can be accomodated and

can consume the excess bandwidth. However, when the network load becomes high, priority will be given to premium traffic, although some bandwidth must still be made available for basic users.

This specific instance of a QoS mechanism is chosen because it is simple enough so that the number variables and parameters can be controlled. However, this relatively simple case can still yield significant results and reveal insights into the potential of the techniques which are used in this project. In the future, this two class service could be extended to provide support for guaranteed and predictive service with additional control mechanisms such as an admission control algorithm or a more sophisticated high level signaling protocol (i.e., RSVP).

Chapter 4

Hypothesis

4.1 What

The basic question which guided this project is: Is it possible to add and implement effective QoS measures in the upstream channel of an HFC network with only changes in the head end and with no changes to the cable modem behavior? The short answer to this question, as will be demonstrated in the rest of the paper, is *yes, it is indeed possible*, and ways in which this could be accomplished will also be suggested. A fair question to ask at this point might be: Why is the hypothesis either interesting or important to study? The next section addresses this question.

4.2 Why

The inherent characteristics of HFC networks make the head-end an obvious target for introducing QoS mechanisms. As noted in section 2, the head-end is located at the root node of the tree-and-branch topology of the HFC network and is thus in the ideal position to make decisions about link allocation and management. Also, the head-end can be easily and completely controlled by the operator of the cable system. Thirdly, the head-end will have access to specific information about all the cable modems that it services, such as billing information and MAC addresses, through communication with the servers in the server complex.

The primary advantage of having changes only in the head-end is that there is a minimal cost associated with upgrading the link to support QoS. Cable modems are already becoming available in increasing numbers and a head-end only solution means that the cable modems can be distributed to homes and users now, with QoS control features added later, as they are wanted or needed. The cable operator would not have to make an additional investment in the future in distributing newer modems to subscribers or in developing and downloading firmware to already deployed cable modems. An important distinction to keep in mind here is that not only are the cable modems unchanged, but also every cable modem has identical behavior. Stated anther way, premium users and basic users would be use exactly the same equipment and the only differentiation between the two classes would be in some soft state at the head-end. This means that a user's service class could be changed dynamically.

Also, since changes are only instituted in the head-end, the system is very flexible. As applications and users' behavior continue to change and evolve, a QoS scheme may also need to adapt to these changes. New algorithms can be implemented at the head-end which perform better under the new conditions, and again, there is no reinvestment in new modems to take advantage of this improved service. These changes could be implemented with just an update of the software in the head-end.

As stated earlier, in this project I have limited myself to studying schemes which implement dual service classes of premium and basic users because it is a relatively simple case and its scope can be carefully controlled. However there is reason to believe even this simple scheme, or a similar one with more than two classes, could be a valuable feature in a real system in and of itself. Certain users may want or need better access network performance and be willing to pay for it. Cable network operators would be able to charge premium users more than basic users for better access to the link and thereby enhance the value of their network without a major investment.

Chapter 5

Methods

This section describes the techniques used in this project and details the design choices made, the experiment itself, and the methods used for analyzing and evaluating the results, as well as the reasons these choices were made.

5.1 The Experiment

For the experimentation and testing of algorithms in this project, I used a computer simulation to collect data. The particular simulation used was written in C and is based heavily on code originally written by Reuven Cohen which simulates the upstream channel in an HFC system. The most important aspects of this simulation are described below and further details about the simulation can be found in Appendix A. The advantages of simulation over testing on a real system are that certain problems which are orthogonal to this investigation (RF noise, cracked cables, etc.) and inevitable hardware delays and bugs can be avoided. Also the simulation simplifies the situation by eliminating the concerns of the downstream channel. As stated earlier, the only assumptions I need to make about the downstream channel are that there is adequate bandwidth for a small amount of control information to be distributed, and that there is some fixed transmission delay from the head-end to the modems.

There are still a large number of variables, even after limiting the study to simulation of just the upstream channel. In an effort to control the scope of the project

further, I made certain decisions. The first important decision was to settle on what kind of traffic to generate, and in this simulation I chose to use a Poisson burst model for aggregate traffic in each class. This means that packets are generated at Poisson interarrival times and each packet is then assigned to a random cable modem and enters that modem's queue. The length of each packet is determined by a static distribution and packets are generated independently for basic and premium users, so that packet arrivals are Poisson within each class as well.

The particular static distribution used in this study is derived from a study in which data was collected in a typical local area network ([13]). Since the particular packet size distribution chosen might have severely affected the results, I tested various other distributions, including ones which consisted only of maximum and minimum size IP packets. While there were some minor differences, all of the distributions which were tested had nearly identical performance under the simulation. As long as the maximum and minimum size and the average size were kept the same, the same basic behavior was shown.

This rather simple traffic model has some limitations when compared with more sophisticated models. Specifically, in [23], the authors claim that actual traffic is self-similar and exhibits long range dependencies and cannot be modeled accurately by traditional traffic models. However, due to the timescale and scope of this particular project, it was impractical to add this complexity into the traffic generator. This does not inherently invalidate the work presented here, since the particular performance factors studied in this project will likely be largely unaffected by a change to a more complex model, but the conclusions drawn should be verified in future work with more realistic traffic models.

In this project I chose to make the assumption that while premium users will demand to have better network performance, the workload characteristics of premium and basic users will be the same. That is, premium and basic users will generate very similar traffic with identical characteristics (i.e., same packet distribution, same average interarrival time) and the challenge is to differentiate the performance that packets from different classes receive. For this reason, in the simulation I assume

that there are an equal number of premium and basic users, set to 10, and that both sets of users are creating and trying to send the same amount of average load to the network. The number 10 was chosen after some simulation showed that with smaller numbers of modems there are artifacts due to the round trip delay time and longer queuing in the modems. Also, if each neighborhood consists of approximately 2,000 homes, and that about half of those homes subscribe to broadcast cable television (1,000). Of these homes, perhaps twenty percent will subscribe to cable data services (200), and out of that group we can assume that about ten percent will be active at any given time, which leaves about 20 modems.

5.2 Data Plots

The following sections contain plots of the data collected from the simulation which may need a few words of explanation. In the graphs representing the average delay (figures 5-1 (a), 6-1, 7-1, 8-1, and 9-1 (a)), the x-axis shows the total load which has been created and which the modems are trying to transmit to the network. This load is shown as a fraction of the total capacity of the link, so that a load of 1 corresponds to a data rate equal to the maximum link capacity. Remember that premium and basic users here have the same traffic profile, so half the load is coming from each type of user. The y-axis shows the average delay measured in numbers of slots.[1] This delay corresponds to the number of slots which pass between the time the packet is created and enters the input queue of a cable modem until the time that the last bit of the packet has been received at the head-end. The dotted curves in the graphs represent what behavior a packet would receive in the base case (see section 5.3 below) and is shown for comparison. The solid and dashed lines then correspond to the behavior that packets from premium and basic users, respectively, would receive under the algorithm which is being tested.

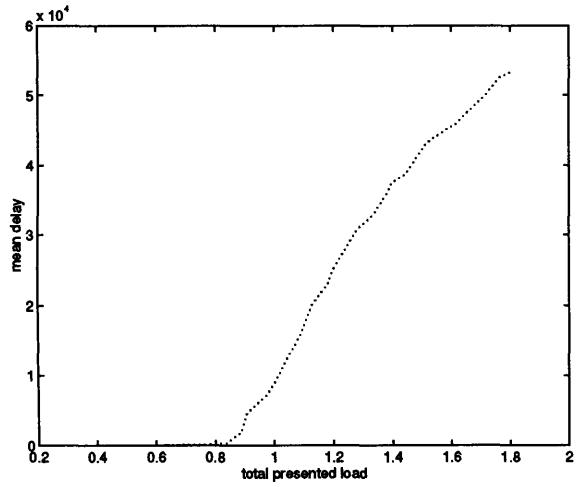
The average throughput curves (figures 5-1 (b), 6-2, 7-2, 8-2, and 9-1 (b)) have

[1]For a typical upstream channel that has a total bandwidth of 3 Mb/s and with 64-byte slots, each slot is equal to approximately 0.17 ms.

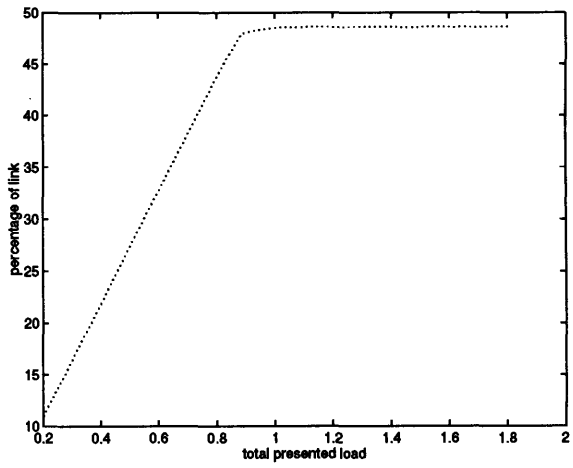
the same x-axis of presented load as before and the dotted line again represents a base case for comparison (section 5.3). The solid and dashed curves correspond, respectively, to the aggregate throughput that the premium and basic classes each receive under the algorithm which is being simulated. The vertical axis measures the average raw throughput, which is given as a percentage of the total raw link capacity. This throughput statistic is collected by tabulating the total number of slots of data that are sent upstream by each class, and then dividing that amount by the total number of slots of the entire simulation run. Finally, figures 5-1 (c), 6-3, 7-3, 8-3, and 9-1 (c) show the percentage of slots which are "free" (i.e., unused and available for contention) compared again to the total presented load to the network. Again, the dotted line corresponds to the base case and here the solid line corresponds to the algorithm being implemented.

5.3 The Base Case

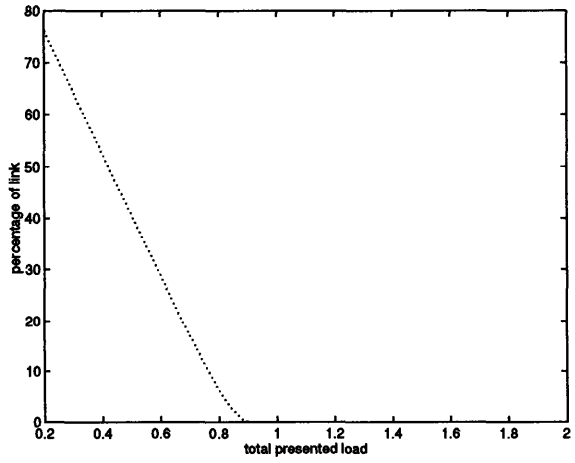
A base case is used in all of the graphs described above in order to provide a constant frame of reference which can be used for comparison across all the tested algorithms. The scenario which is used for the base case is one where there are no priority distinctions among the modems, and all twenty modems on the link receive the same level of service. Figure 5-1 shows the results gathered from this scenario. These curves are repeated in all future graphs as dotted lines. Note that the throughput plot (figure 5-1 (b)), actually shows the aggregate throughput that modems in this scenario would receive reduced by a factor of two and the actual total throughput (and utilization of the link) is twice what is shown on this plot. For example, a point on the graph showing 25% of the link actually corresponds to a link utilization of 50%. This factor is included in the base case plots as a normalization factor in order to provide a more meaningful comparison with later experiments. Remember that in the cases where premium and basic users are distinct, there are only ten modems of each class, but in the base case there are a total of twenty ("classless") modems. To account for this difference, it is necessary to include a factor of two.



(a) Mean delay



(b) Throughput



(c) Percentage of free slots

Figure 5-1: The base case

5.4 Analysis

One of the major difficulties in studying networks is determining which data to collect and then how to interpret the data; the HFC network is no different. Ultimately what cable operators will be concerned with is keeping their customers satisfied with the cost-performance of the network, but the challenge is to correlate this "satisfaction" metric with values which can be actually measured and controlled in a network. It has been suggested ([8]) that what end users really care about is an overall "transaction time," the time between the initiation of a transaction to its completion, where an example of a transaction could be up/downloading a file.

However, this metric is still difficult to quantify since different users have different expectations for different applications. As technology, applications, and user behavior changes, this metric also changes. In order to restrict this effect, I assume that this "transaction time" from a link layer perspective can be characterized by studying the mean packet delay and the average throughput. The delay is the dominant factor in determining the setup time for an action and the throughput controls the time after the setup until completion.

A possible point of confusion is in the choice of the horizontal axis, and specifically in what is meant when when the presented load is greater than one. This axis refers to the amount of data that is actually being created by the traffic generator model. Of course there can never be a load of more than one on the link, so most of the data which is created in this region ends up being queued for long delays in the cable modems. I contend, however, that this high load region of the graphs is the important area for study. I have assumed that most applications will use a congestion avoidance protocol such as TCP above the MAC layer. TCP has the properties that it tries to acquire as much of the available bandwidth as it can, but when it detects a reduction in bandwidth, it assumes this is caused by congestion in the network, and will back off and reduce the bandwidth it consumes. The first property implies that the link will normally be in very high load since TCP will use all the available bandwidth. And the second property implies that if we can apply some bandwidth limits in the high load

region, TCP will cause the traffic to slow down and conform to that bandwidth limit. Future work might involve integrating a TCP simulation into the HFC simulation, but that lies beyond the scope of this project

The other network performance parameters that were collected from the simulation were the number of contention slots which had a collision, the number of free or empty contention slots, the standard deviation of the packet delay, and the average reservation length. These were important for revealing further insights about the behavior of the system and what the effect of different implementation tradeoffs might be.

With this framework in mind, the following three sections explain three different algorithms which could be used to implement a dual priority class service. Section 6 explains the delayed reservation algorithm, which imposes a minimum delay on all basic user requests. In section 7, the rejection of reservation scheme, which arbitrarily rejects certain some basic user requests, is described. And finally, section 8 describes the frame algorithm, which uses fixed length frames to restrict basic users to a maximum bandwidth.

Chapter 6

Delayed Reservation

This section explains the first of three algorithms, the *delayed reservation* algorithm, that were simulated in this project. The explanation is followed by simulation results and some brief interpretation and analysis of those results.

6.1 How it works

The basic idea of a delayed reservation is fairly simple. The algorithm in its simplest form processes reservation requests in the following manner. When the head-end receives a reservation request from a premium user, it schedules that request by searching for the first available sequence of slots that is long enough to accommodate that request, starting from the next slot (accounting for the downstream transmission delay in informing a modem of a reservation). However, when it receives a reservation request from a basic user, the head-end enforces a minimum delay in scheduling the request. That is, instead of starting the search for empty slots from the next slot, the head-end starts searching some fixed number of slots beyond that point.

One obvious effect this algorithm will have is to increase the average delay a basic packet (i.e., a packet from a basic user) will experience, since basic packets which could be scheduled immediately are now forced to be delayed a minimum time. Also, since that modem cannot transmit in the intermediary time period, this algorithm effectively reduces the maximum bandwidth that basic modems can consume. By

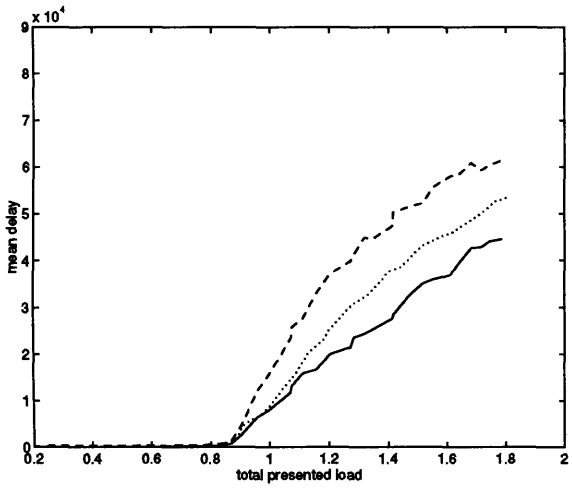
forcing the basic modem to wait, the channel is cleared to potentially allow premium modems that have data to gain access to the link. One tradeoff in scheduling requests further into the future, however, is that there is an increase in both the processing time, in order to scan a longer allocation vector, and the memory requirements, to store the longer vector, at the SCS.

6.2 Results

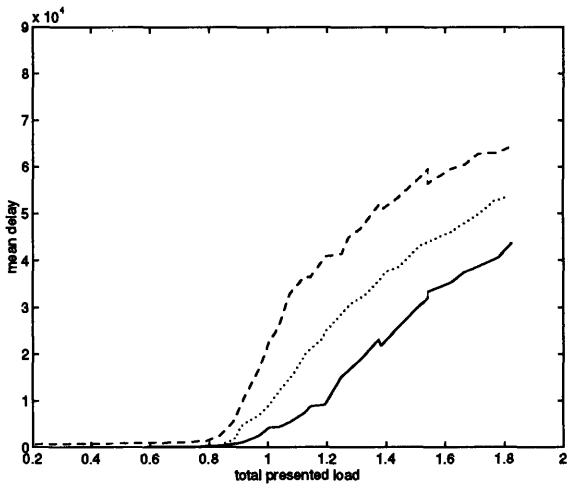
Figures 6-1, 6-2, and 6-3 show some results from simulation and indicate the general trends as the imposed delay is increased from 200 slots to 500 and then 2000 slots. Remember that the dotted curves represent the base case that is used for comparison (as explained in section 5.2), the solid curves represent premium user behavior, and the dashed lines represent basic user behavior. Figure 6-1 shows the expected increase in the delay of basic packets and the corresponding improvement in the delay as seen by premium packets.

Looking at the throughput curves in figure 6-2, the basic user does get much less average bandwidth with the delayed reservation algorithm and most of that bandwidth can be reclaimed by the premium user. However, as the imposed delay gets very large, as in figure 6-2 (c), the basic user's average bandwidth is restricted even when there is capacity in excess of the actual presented load. This is confirmed in 6-3 (c) by the large increase in the number of unused slots, which corresponds to a drop in the utilization of the network.

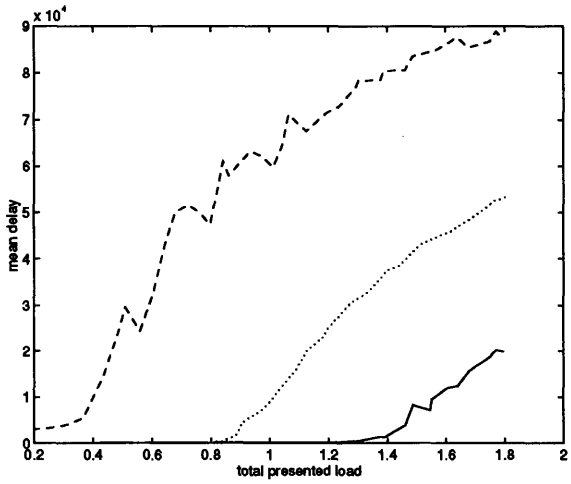
This delayed reservation algorithm in this form is probably not the ideal one for implementing QoS, but it does demonstrate the potential for head-end control. The behavior of this system when the network becomes overloaded (i.e., when the modems are trying to send more data than link capacity) is desirable in that the delay for premium users stays low at the expense of delay for basic users, but the basic users still maintain a minimum bandwidth and are able to send some data. However, the problem is that many slots are being wasted, and we would like to be able to allocate those extra slots to basic users when the premium users are not using



(a) delay=200 slots

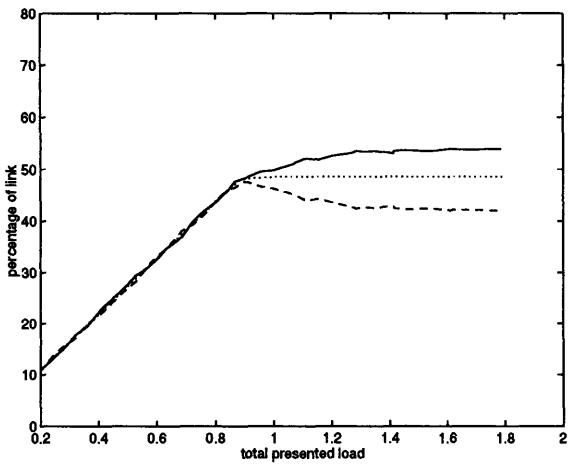


(b) delay=500 slots

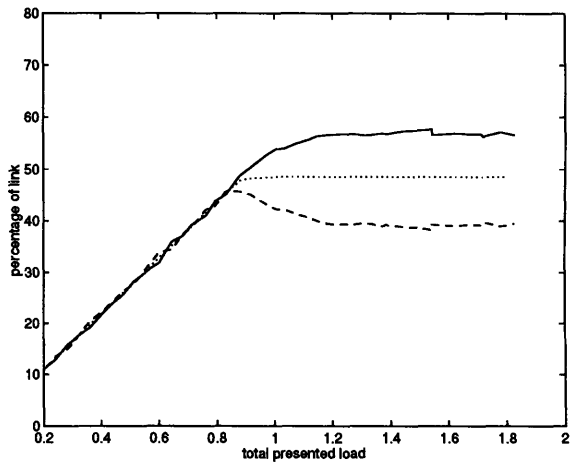


(c) delay=2000 slots

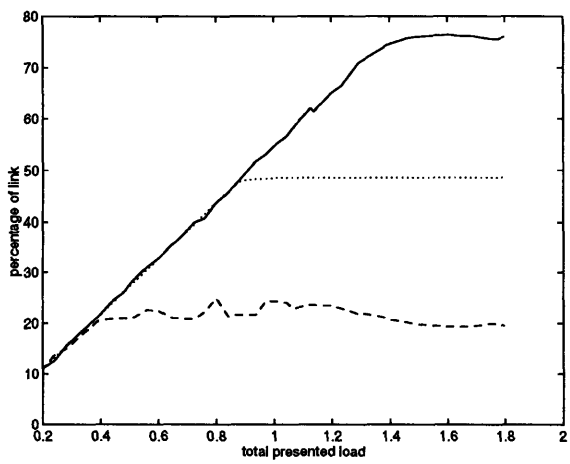
Figure 6-1: Mean delay in Delayed Reservation



(a) delay=200 slots

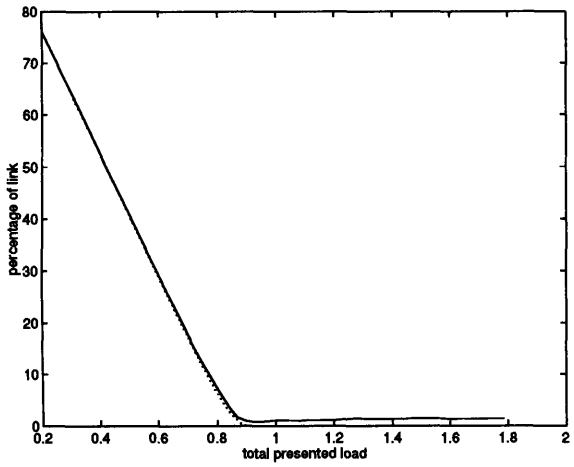


(b) delay=500 slots

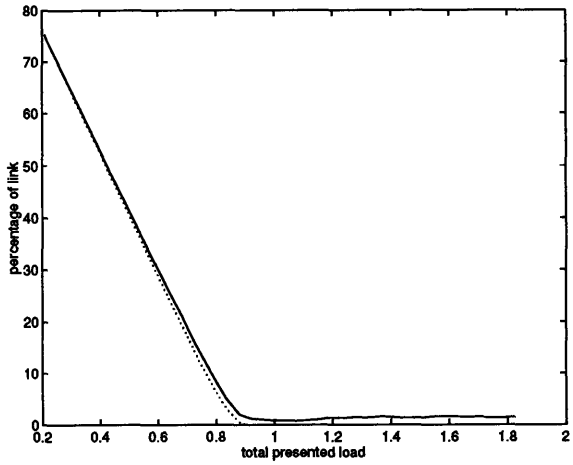


(c) delay=2000 slots

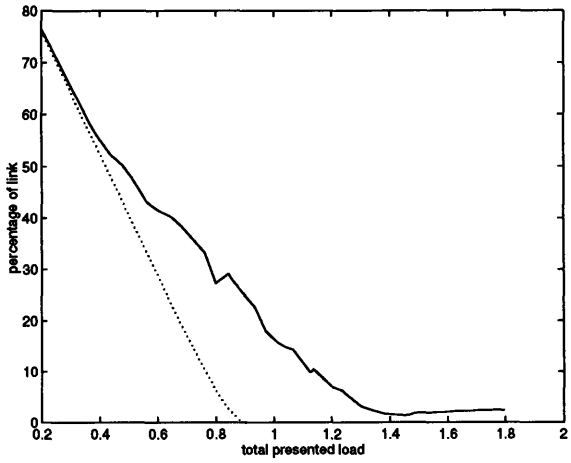
Figure 6-2: Throughput in Delayed Reservation



(a) delay=200 slots



(b) delay=500 slots



(c) delay=2000 slots

Figure 6-3: Percentage of free slots in Delayed Reservation

them. This suggests that the head-end should be able to monitor the load on the network and dynamically adjust the amount of the fixed delay depending on that load. Some methods for doing this monitoring will be explored later.

Chapter 7

Rejection of Reservation

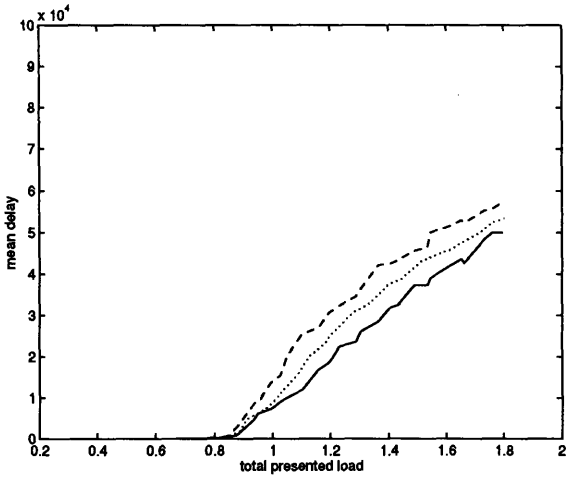
This section explains another algorithm which I call *rejection of reservation* and is then followed by some results from simulation.

7.1 How it works

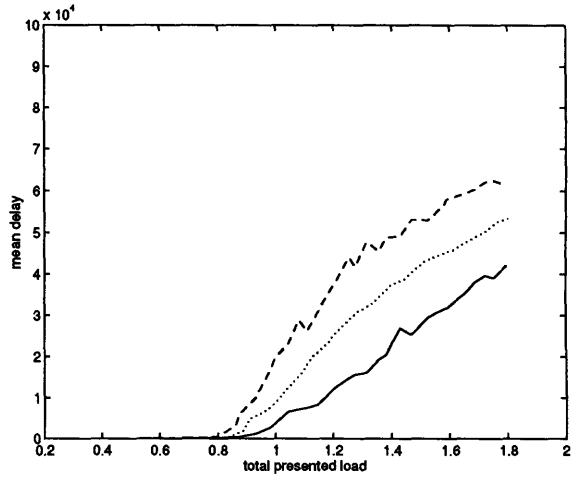
In this algorithm, the head-end will randomly reject reservation requests from basic users at some predetermined rate. This means that even though there was no collision and no corruption of the request, the head-end will not send an acknowledgment back to that modem. This forces the modem to execute its backoff algorithm and to contend at a later time in order to try to send the packet again.

As before with delayed reservation, while the modem is waiting a timeout period for an acknowledgment and then while it is executing the backoff algorithm, that modem is unable to transmit, which frees the channel for traffic from other modems. Again, there is an expected increase in the average delay of basic packets, since some reservations which would have been accepted immediately are instead turned away and forced to go through the reservation process again. This also leads to an overall decrease in bandwidth for basic modems, as in the previous delayed reservation case.

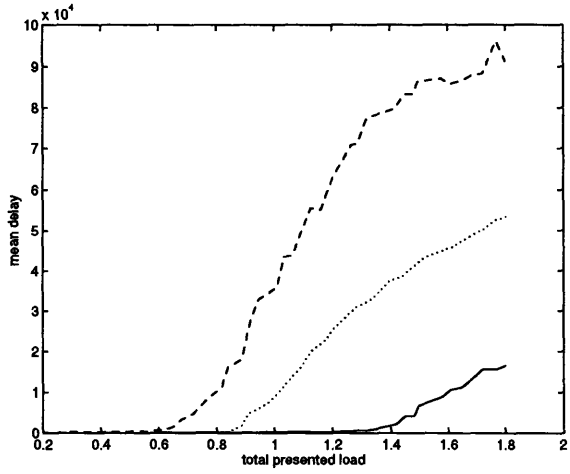
7.2 Results



(a) rate=25%

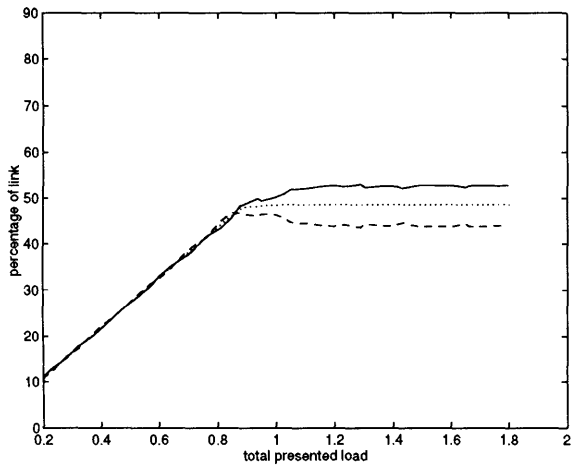


(b) rate=50%

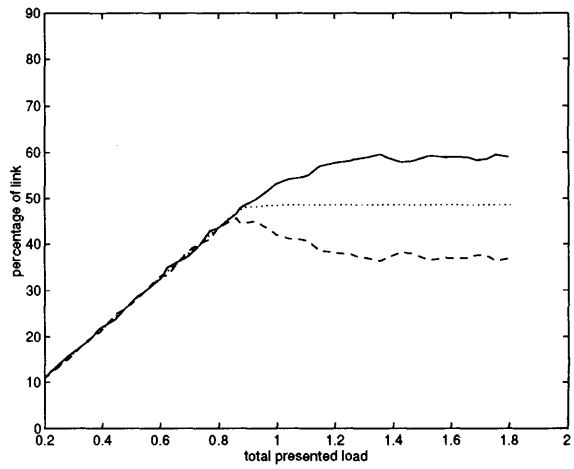


(c) rate=90%

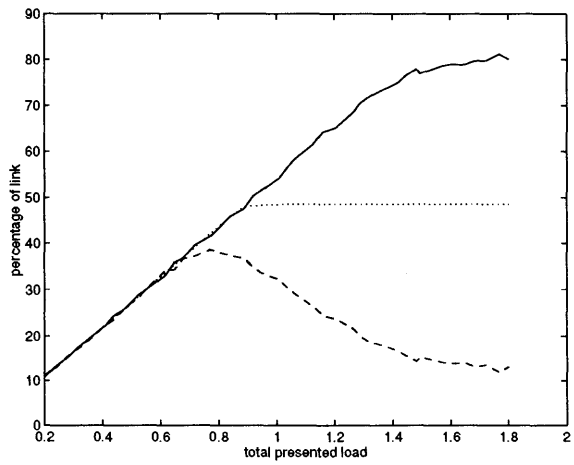
Figure 7-1: Mean delay in Rejection of Reservation



(a) rate=25%

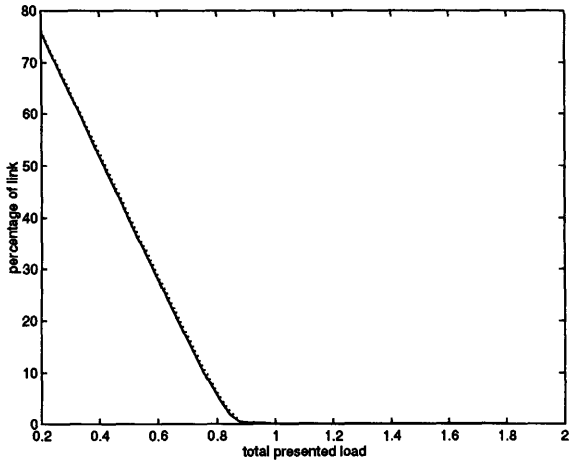


(b) rate=50%

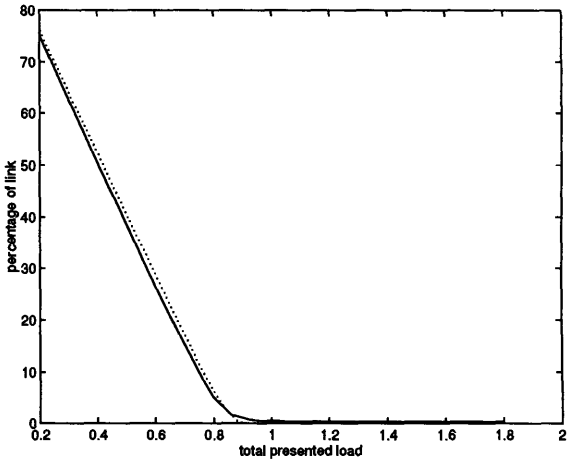


(c) rate=90%

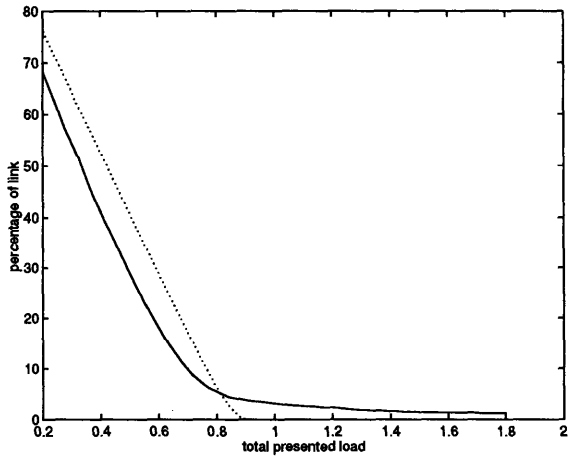
Figure 7-2: Throughput in Rejection of Reservation



(a) rate=25%



(b) rate=50%



(c) rate=90%

Figure 7-3: Percentage of free slots in Rejection of Reservation

As shown in figure 7-1, there is the expected increase in the delay of basic packets and the decrease in delay of premium packets. This effect becomes more pronounced as the rate of rejection is increased from 25% to 50% to 90%. However, comparing these plots to the ones from section 6, there is a much smaller impact on the delay of basic users at low loads for a similar improvement in the delay of premium users. The throughput curves in figure 7-2 demonstrate the bandwidth reduction for basic users and the corresponding bandwidth increase for the premium users. Note that in figure 7-2 (c), instead of sharply restricting the bandwidth for basic users, their bandwidth is gradually decreased as the load from the premium users increases which gives better utilization as compared to delayed reservations. Figure 7-3 shows that indeed there are very few wasted slots. The reason that in figure 7-3 (c) the number of free slots is actually lower than in the base case for low loads is that by rejecting some basic user requests, more free slots are used when those requests recontend.

Overall, the general trends which we would expect a priority system to exhibit are observable. There is the improvement in both delay and bandwidth for premium users while still maintaining acceptable performance for basic users. The gradual decrease of the basic users' bandwidth and the high utilization is highly desirable, since it means that the link is being used efficiently and that basic users are granted access to the link when premium users do not need it. In section 9, some modifications to this scheme are proposed to increase efficiency further. Note that under this algorithm, unlike the delayed reservation algorithm (and also the frame algorithm, as we will see later), basic users do not maintain a minimum bandwidth. When premium users generate a very large amount of traffic, the basic user may get essentially zero bandwidth. This may or may not be desirable behavior, depending on the system and the business agreements which are in place.

Chapter 8

Frames

The *frame* algorithm is explained in this section as a third possible alternative for implementing a two priority system and is also followed by a brief discussion of the results. This algorithm is similar to a number of scheduling algorithms which use a frame to allocate bandwidth ([2]).

8.1 How it works

In this algorithm, the upstream channel is divided into fixed length *frames* and each frame consists of a predetermined number of slots. Within each frame, basic users are allowed to reserve a maximum number of slots. If scheduling a basic packet would result in this maximum being exceeded, the head-end pushes that reservation into the next frame. Otherwise, the head-end will schedule all reservations in the first available space, as in the base case.

The net effect of this frame algorithm is that basic users are restricted to a maximum average throughput, leaving the rest for premium users and for contention slots. The actual throughput seen by basic users will be slightly less since there is no fragmentation of packets in the system, and some basic slots within a frame may not be allocated to basic users if a reservation will not fit within the remaining basic slots in that frame.

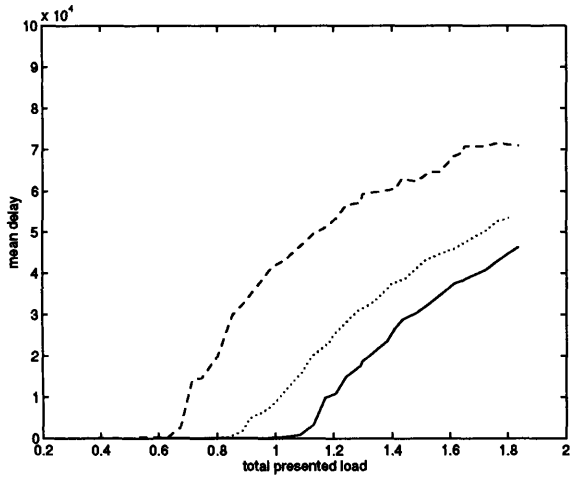
8.2 Results

Figures 8-1, 8-2, and 8-3 show the results from three different simulations of the frames algorithm. The first step was choosing the frame size. If the frame size is too large, both the delay and the variability of the delay that basic users experience increases since when the portion of one frame allocated to basic users is consumed, the next basic request must wait until the next frame. However, if the frame is too short, efficiency decreases due to the fact that no fragmentation is done and this will affect both premium and basic users. Each time there is a transition from basic to premium packets from at the frame boundaries and at the basic boundary within each frame, there is potential for some slots to be wasted. Also, both the frame and the segment allocated to basic users should both be either greater than or equal to the maximum allowed reservation size, otherwise those maximum length reservations would never be accepted by the head-end.

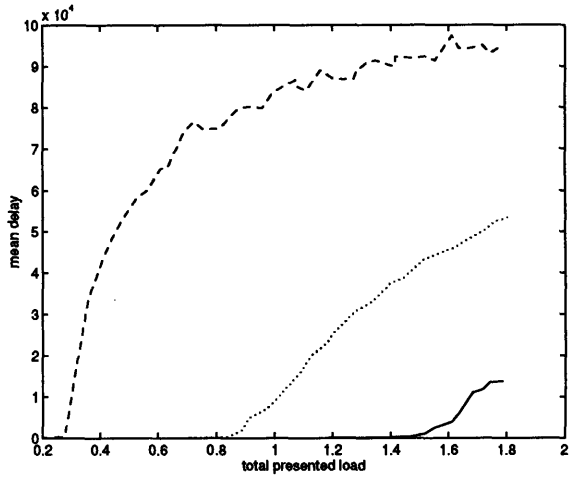
A simple example of this effect is shown in figure 8-4, where the frame size is 10, of which a basic user may reserve a maximum of 4. At the beginning of frame 1, a premium user (P_1) has reserved the first 4 slots in the frame. A basic user (B_1) then makes a request for 3 slots and since that is still allowed for this frame, the head-end schedules it immediately. Then another basic user (B_2) requests 4 slots, and since this would exceed the basic user limit for frame 1, the head-end schedules it at the beginning of frame 2. Still in frame 1, the head-end receives a request from a premium user (P_2) for 5 slots, but because it has already scheduled into frame 2, the premium request in this case cannot be scheduled immediately and must wait until after the basic reservation at the beginning of frame 2, clearly an undesirable situation.[1] After some initial simulation, I chose a frame size of 500 slots. Larger frame sizes exhibited a high degree of instability and smaller frame sizes greatly reduced the efficiency and utilization of the link.

The delay curves in figure 8-1 show that the basic user delay increases dramatically

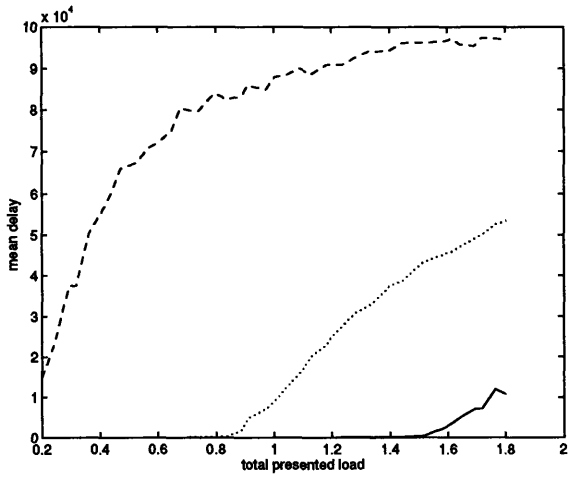
[1]Note that if P_2 had made a request for 3 or less slots, that request could have been granted in Frame 1.



(a) premium=60%

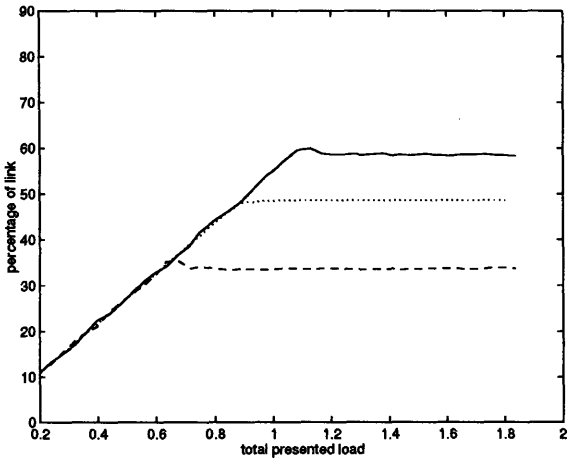


(b) premium=80%

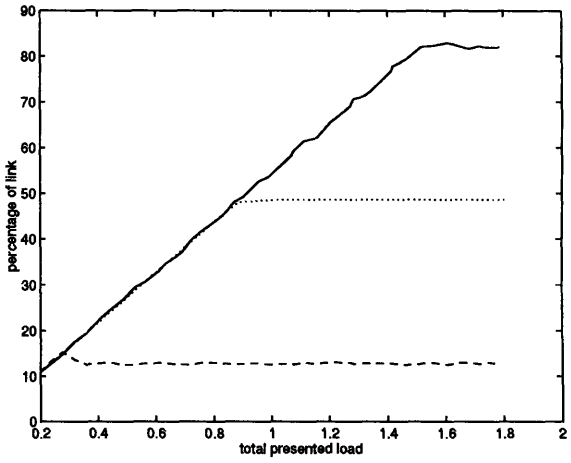


(c) premium=90%

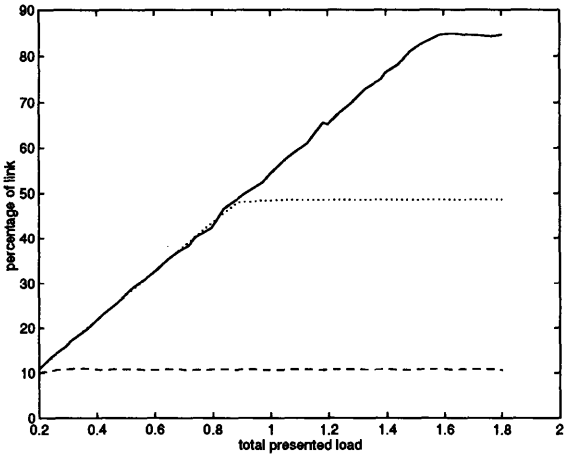
Figure 8-1: Mean delay in Frames



(a) premium=60%

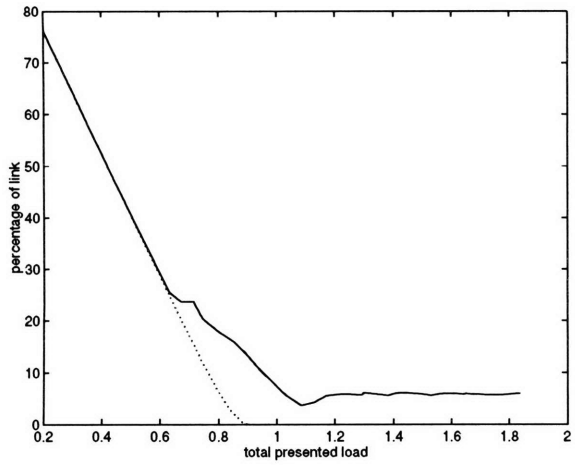


(b) premium=80%

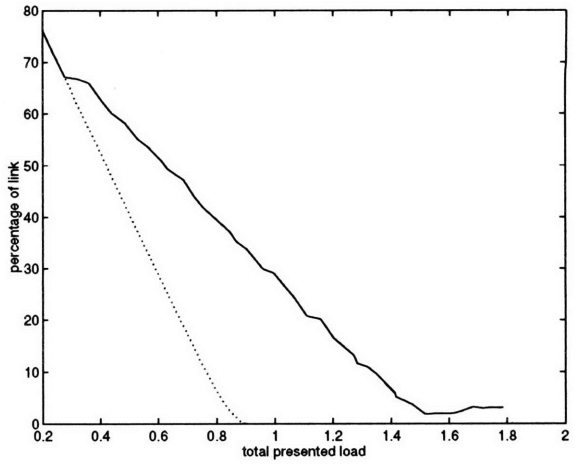


(c) premium=90%

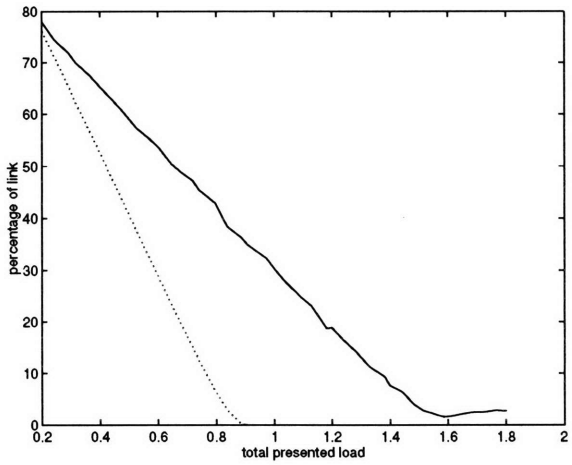
Figure 8-2: Throughput in Frames



(a) premium=60%



(b) premium=80%



(c) premium=90%

Figure 8-3: Percentage of free slots in Frames

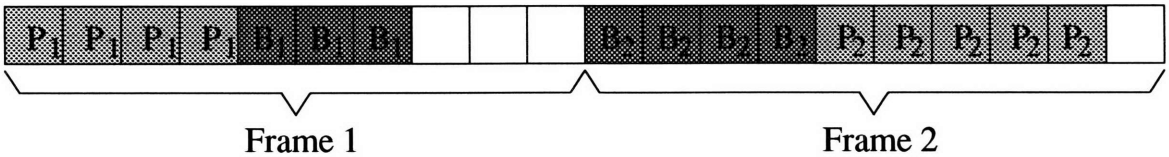


Figure 8-4: Frame example

as the portion of the link reserved for premium users increases from 60% to 80% and 90% (or 300, 400, and 450, slots out of 500). The throughput curves shown in figure 8-2 display the expected division of throughput, with the basic modems claiming slightly less than the maximum bandwidth they are allowed, due to the inefficiencies mentioned above. This inefficiency is more clearly demonstrated in figure 8-3, where the large number of wasted slots is plainly observable. On the basis of this output alone, this solution does not appear to be very useful, but there are also ways in which this system could be altered to improve performance.

Chapter 9

Results and Refinement

This section refers back to the results of the previous three sections and draws further conclusions about the relative performance of each. It also suggests refinements to the algorithms which may improve performance.

As noted previously, all three algorithms were successful to varying degrees in providing clearly differentiable levels of service to premium and basic users. In all three cases delay for premium users was decreased and bandwidth was increased. However, one area where the three diverged was efficiency, or overall link utilization. Since the goal is to try to statistically multiplex two classes of service onto the same physical channel (as opposed to having completely separate channels, for example with time-based or frequency-based division), the link utilization is an important consideration and a better system will have higher utilization.

It was observed that the rejection of reservation scheme worked best in this regard, since as long as premium users did not have traffic to send, basic users could acquire more bandwidth. But, as the load presented by premium users increased, the bandwidth that basic users were able to reserve was gradually decreased. The system was able to compensate for the increased load without further intervention by the head-end. This is the biggest advantage of the reservation rejection scheme, that there is very little processing overhead at the head-end and scheduling decisions can be made in a relatively short time.

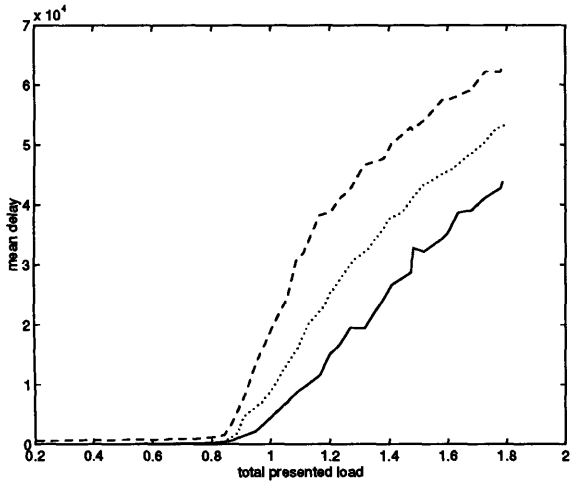
However this method still has some shortfalls, the biggest being the increased

use of the contention channel. By arbitrarily rejecting some basic user requests, we artificially increase the collision rate, when in actuality there are no collisions. Since the modem whose reservation was rejected will eventually still want to send that packet, we would like to be able to devise a system where it would not have to contend again. The head-end should be able to predict how the backoff scheme would work and automatically incorporate that extra delay into its scheduling algorithm. This evolves into an algorithm which combines elements of the rejection scheme with the delayed reservation scheme. A sample of the output from a simple implementation of this idea is shown in figure 9-1, where 90% of basic user requests are delayed 500 slots, instead of rejected. Note that there is still the gradual reduction in throughput, but there is improvement in the graph of the number of free slots.

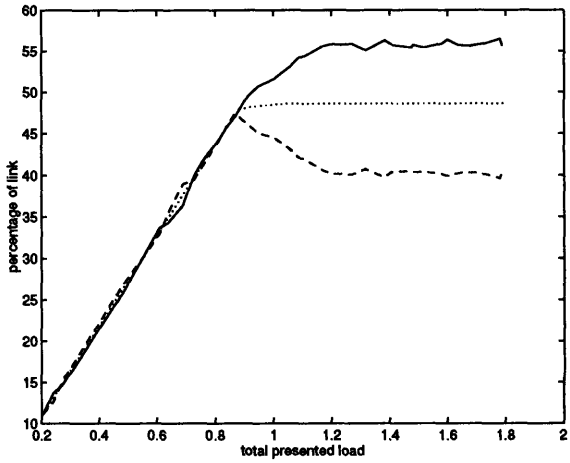
The biggest problem with both the delayed reservation and the frame algorithm is the large processing overhead. In the simulation I assumed that the head-end had a very large vector for scheduling packets and that it could look very far into the future to schedule a packet. But in reality, since the searches are linear, this becomes very time-consuming as the network load increases and the head-end must schedule packets further and further into the future. It may not be practical to expect that the head-end will be capable of making these calculations within a reasonable timeframe. This disadvantage is not as pronounced in the reservation rejection algorithm.

Another problem with the delayed reservation and the frame algorithm is that basic user performance is degraded even at low network loads, when in reality there is excess capacity available for basic user requests. Better implementations would constantly monitor the load on the network and adapt the algorithm accordingly. In the case of the delayed reservation algorithm, this may mean having a dynamically self-adjusting delay which is variable and not fixed. This delay would slowly increase as the network load increased. In the case of the frame algorithm, this might mean restricting the basic users to fewer and fewer slots per frame as the load increased.

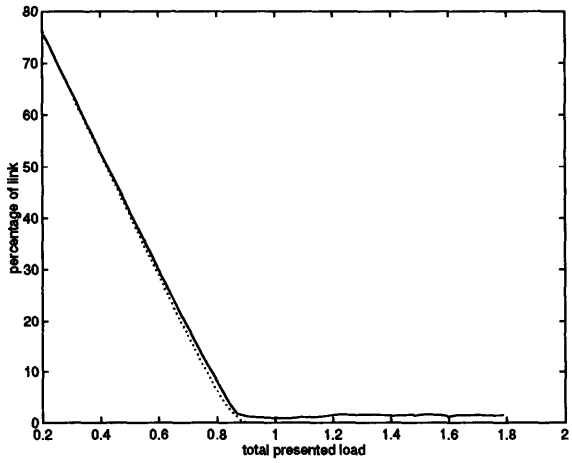
In order to dynamically adapt as the network load changes, a robust method for monitoring the network load is needed. The simplest method, implemented at the head-end, is to observe the number of empty or free slots, sampled at some regular



(a) Mean delay



(b) Throughput



(c) Percentage of free slots

Figure 9-1: Combination algorithm

rate. When there are a large number of free slots, network load is generally low, and when free slots are scarce, the opposite is true. By constantly sampling the free slots, the head-end would be able to determine the load on the network, and depending on what performance is desired or dictated by business agreements, it can adjust the algorithm parameters accordingly.

As mentioned earlier, one aspect which was ignored in this study, but which could affect the performance of any algorithm implemented at the link level, is the effect that higher layers in the protocol stack could have. Specifically, a protocol such as the transport control protocol (TCP) could change the behavior of the sources. In the simulations, there is a constant load to the network which does not change with changing network performance. However, TCP can adapt to the amount of bandwidth available by altering its window size. Its congestion avoidance algorithm will cause applications to slow their rate of transmission as bandwidth on the link is decreased. As premium users impose greater and greater load on the network and crowd out basic users, TCP in the basic user data streams will sense this increased load and shrink their window sizes, relieving the strain on the network. Again, this is why it is important to study the behavior near a constant load of one, since TCP will dynamically scale its flows to try to meet that load. An additional complexity which this introduces is that in trying to match the available bandwidth, the bandwidth TCP actually consumes tends to oscillate and further study would be required to fully understand those effects.

The frame algorithm may see like an unattractive choice since it shares many of the faults of the delayed reservation and none of the advantages of the rejection scheme. However, there is one advantage that the frame algorithm has over the other two. Eventually networks will probably need to be able to provide finer granularity in QoS mechanisms than just a dual priority class system. Networks may have to be able to separate certain data streams as requiring better service and may need to make better guarantees for those streams. In the future, networks may also have to support constant (or variable) bit rate traffic which requires a certain number of slots at regular (or semi-regular) intervals. The frames algorithm provides a framework, shared by

many techniques proposed for real-time switches ([2]), in which these features could be integrated more easily. This inherent structure may make this system more attractive in the future.

The other advantage of the frame algorithm is that it allows for much more control over the actual bandwidth each class receives. In the delayed reservation and reservation rejection schemes, the bandwidth which a basic user receives may depend on a number of different variables and the relationship between that bandwidth and the algorithm parameters is not obvious. However, in the frame algorithm this relationship is much clearer and it is easier to restrict basic users to a specific bandwidth.

Chapter 10

Conclusion

The following section contains some final conclusions and also suggests other related areas which might be of interest in future studies.

Returning to the original hypothesis proposed in section 4, namely whether effective QoS measures can be added to an HFC network with changes only in the head-end of the system, the primary conclusion in this project is that this hypothesis is most certainly valid. As demonstrated by the results presented in previous sections, QoS mechanisms can be successfully integrated into the HFC head-end. While this statement may seem very simple, it raises some interesting implications. Both cable operators and users can make investments into the system and not worry about immediate obsoleteness. Being able to implement new algorithms by changing the software at the head-end renders the system very flexible and allows the network to extend its useful lifetime.

The algorithms detailed above served mainly to support the hypothesis. However, as shown in the results, they are significant in themselves as possible real implementations. Some combination of one or more of the algorithms, coupled with refinements from the discussion in section 9 and from further testing, could eventually find its way into an actual deployed head-end on an HFC network. The general trends and overall behavior observed above could be expected to propagate through in future development.

10.1 Future Work

This project could be accurately described as a part of a work in progress. Although some significant results and conclusions were drawn in this paper, clearly there is still more work to be done. Even within the relatively narrow scope of this project there are still unexplored avenues such as other algorithms or enhancing the capabilities of the simulation which may yield even more interesting results.

Outside the scope of this project there are also still many areas which will be important for providing interactive services to a broad base of users and for implementing quality of service mechanisms into future networks that lie beyond the scope of this thesis. One such area is mapping and defining QoS demands and metrics from a more human perspective. This involves determining factors such as how much video or audio detail a person can actually resolve, and which kinds of lapses in QoS are tolerable for different users and applications and which kinds are not. Further study in this area would probably consist of experiments involving human test subjects and other more qualitative forms of analysis. Applications will also have to change in the ways they interact with people and with networks and there is a great deal of work to be done in developing more advanced applications.

In this project I made the assumption that appropriate developments will take place in the evolution of RSVP and that an interface will exist both for applications to communicate with RSVP agents and also for the RSVP protocols to be mapped into cable protocols. There are some issues here which need to be defined and resolved as to the best way to implement these interfaces and what features they should have. A complete end-to-end solution will require the end points to extend all the way to the user and the top application level and a resource request must pass from the user, through the application, the operating system on the host machine, and the link between the computer and the cable modem before it enters the domain of the HFC protocols which were studied here and the relationships among all these layers will require careful study.

As discussed previously, layers higher on the protocol stack such as TCP will likely

influence the behavior of the system and these influences should be accounted for in future studies. Also mentioned above is the simplistic traffic model which was used in this project. Future work should include using a more sophisticated and more realistic model for the traffic on the network in order to verify the results presented here.

As cable modems become more readily available and as cable data networks are deployed, there are opportunities to test and refine algorithms in a real HFC network environment. Simulations are effective for initial exploration of ideas, but there are inevitable factors which are unaccounted for in simulations which reveal themselves in an actual system. Real users may not behave in entirely predictable ways and other unforeseen problems may arise. There are also many details which were ignored in this project, but which would become important for a real deployment. For example, problems in the downstream channel will also affect the upstream allocations. Also, the head-end may need to implement some admission control, which would not allow any new modems to join the network if it is highly congested. Again, by requiring that changes take place only at the head-end, the process of adjusting the system after deployment becomes much less costly and much less work intensive.

Another aspect which will require further study is the merging social and economic considerations with the technical ones which this thesis will explore. I have assumed that a form of a payment or cost system will be in place to provide a tradeoff for reserving a higher or lower form of service. Otherwise, it would be in the best interest of each application to ask for the best service it can get. In this example, if premium and basic users pay the same amount, everyone would opt to become a premium user. A payment structure would offer incentive for users with less rigid and less critical applications to accept a lower QoS in exchange for a lower cost. Also, since we unfortunately live in an imperfect world, there should be provisions against abuses of the reservation system, such as one host using the reserved (and paid for) resources of another host. The structure of each local cable network will most likely be controlled by a single entity, the cable operator who controls the head-end, and each host that is given access to the network agrees to be "well-behaved" and adhere to the established

protocol. However, some policing is still necessary to ensure that there are no abuses, either intentional or unintentional.

The evolution of networks is an ongoing process and taking careful steps now, in the earlier stages of development, will hopefully yield an architecture which will carry us forward for many years. Any system which is deployed must have at least one eye to the inevitable changes of the future. The Internet, in one form or another, is here to stay, and as we approach the next century, there is an enormous and growing opportunity for far reaching social and economic change as a result of newly developing technologies.

Appendix A

Simulation Details

This section contains a detailed description about the simulation and how each experiment was conducted.

The program used to simulate the upstream channel was based on a simulation written in C by Reuven Cohen and was compiled and run on H-P workstations. Since the upstream channel is slotted, it was natural to update state information after each slot. Each slot was 64 bytes long and the maximum length of a single reservation was 63 slots (4032 bytes). The buffers in both the cable modems and in the head-end were set to be very large so that they would never overflow.

For each experiment there were 20 active cable modems sharing a single upstream channel (and in a real system they would also share a single fiber node). The round trip delay between the head-end and the modems was set to 10 slots (approximately 2 ms for a 3 Mb/s channel). Ten modems were designated premium modems and the remaining ten were designated basic modems. This distinction was only used to schedule reservations at the head-end and the actual operation and behavior of the modems are identical (see section 2.3). The various algorithms described in sections 6, 7, and 8 were used to perform the scheduling.

At the start of each iteration new packets are generated according to the Poisson burst model described in section 5.1 and the static packet length distribution was taken from the paper by R. Gusella [13] on Ethernet traffic. The basic packet generator would generate packets which arrived during that slot and then assign it to

a random basic modem, and then similarly for the premium packet generator and the premium modems. Each modem would contend in the next available contention slot if it had data waiting in its queue to be sent. In that contention slot, a modem would make a reservation for as many of the packets waiting in its queue, up to the 63 slot maximum. For this project, I ignored the problem of loss and assumed that the upstream channel was error free and so the modem would get a reply from the head-end in one round trip delay, unless there was a collision. In the case where there was a collision, each modem involved in the collision would execute a standard random backoff algorithm.

Each run of the simulation was started with a warmup period of 20,000 slots in order avoid start-up effects and then data collection was performed for a simulated time of 200,000 slots per data point. The values for the warmup period and the running time were determined empirically to ensure that the system had reached a stable point. For each plotted curve, the load presented to the network was swept from 0.2 to 1.8 in increments of approximately 0.04. The results were filtered to extract data which could then be imported into MatLab for graphing.

Bibliography

[1] M. Adams. "Compression trial results: A full service network update," *Proceedings of the SCTE 1995 Conference on Emerging Technologies, Orlando, Florida*, January 1995, pp. 65–76.

[2] D. S. Batlle. "MAC Protocols for Multimedia Data over HFC Architecture," *Georgia Tech Technical Report GIT-CC-95/18*, October 27, 1995.

[3] C. Bisdikian, B. McNeil, R. Norman, and R. Zeisz. "MLAP: A MAC Level Access Protocol for the HFC 802.14 Network," *IEEE Communications Magazine*, March 1996, pp. 114–121.

[4] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and J. Wroclawski. *Resource ReSerVation Protocol (RSVP) — Version 1 Functional Specification*, Internet Draft draft-ietf-rsvp-spec-13.ps, July 19, 1996, ftp://ftp.isi.edu/rsvp/docs/rsvpspec.ID13.ps.

[5] R. Braden, D. D. Clark, and S. Shenker. "Integrated Services in the Internet Architecture: an Overview," RFC 1633, June 1994.

[6] J. Brightman. "Hybrid fiber/coax: Front runner in the broadband transmission race," *Telephony*, November 1994, pp. 42–50.

[7] C. Carroll. "Development of integrated cable/telephony in the United Kingdom," *IEEE Communications*, Vol. 33 No. 8, August 1995, pp. 48–60.

[8] D. D. Clark. "Adding Service Discrimination to the Internet," Version 2.0, September 1995, http://ana-www.lcs.mit.edu/anaweb/ps-papers/TPRC2-1.ps.

[9] D. D. Clark, S. Shenker, and L. Zhang. "Supporting Real-Time Applications in an Integratred Services Packet Network: Architecture and Mechanism," *Proceedings of the ACM SIGCOMM '92*, August 1992, pp. 14–26.

[10] D. D. Clark. "The Design Philosophy of the DARPA Internet Protocols," *Proceedings of the ACM SIGCOMM '88*, August 1988.

[11] J. E. Dail, M. Dajer, C. Li, P. Magill, C. Siller, Jr., K. Sriram, and N. A. Whitaker. "Adaptive Digital Access Protocol: A MAC Protocol for Multiservice Broadband Access Networks," *IEEE Communications Magazine*, March 1996, pp. 104–112.

[12] S. Floyd and V. Jacobson. "Link-sharing and Resource Management Models for Packet Networks," *IEEE/ACM Transactions on Networking*, Vol. 3 No. 4, August 1995.

[13] R. Gusella. "A measurement study of diskless workstation traffic on an ethernet," *IEEE Journal on Selected Areas in Communications*, Vol. 38 No. 9, 1990.

[14] S. Jamin, P. B. Danzig, S. Shenker, and L. Zhang. "A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks," To appear in *IEEE/ACM Transactions on Networking*, 1996.

[15] D. Large. "Creating a network for interactivity," *IEEE Spectrum*, April 1995, pp. 58–63.

[16] J. O. Limb and D. Sala. "An Access Protocol to Support Multimedia Traffic over Hybrid Fiber/Coax Systems," *Second International Workshop in Community Networking*, July 1995, pp. 35–40.

[17] H-J. Lin and G. Campbell. "PDQRAP-Prioritized Distributed Queueing Random Access Protocol," *Processdings of the 19th Conference on Local Computer Networks*, October 1994, pp. 82–91.

[18] J. McConnell and J. Lehar. "HFC or SDV architecture? Economics drives the choice," *Communications Technology*, April 1995, pp. 34–40.

[19] D. Sala and J. O. Limb. "A Protocol for Efficient Transfer of Data over Fiber/Cable Systems," *INFOCOM '96*, March 1996.

[20] S. Shenker. "Fundamental Design Issues for the Future Internet," Xerox PARC, 1995.

[21] S. Shenker and J. Wroclawski. *Network Element Service Specification Template*, Internet Draft draft-ieft-intserv-svc-template-02.txt, Integrated Services Working Group, November 1995.

[22] C. Venkatramani and T. Chiueh. "Design, Implementation, and Evaluation of Software-based Real-Time Ethernet Protocol," *Proceedings of the ACM SIGCOMM '95*, August 1995.

[23] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. "Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level," *Proceedings of the ACM SIGCOMM '95*, August 1995.

[24] C-T. Wu and G. Campbell. "Extended DQRAP (XDQRAP). A Cable TV Protocol Functioning as a Distributed Switch," *Proceedings of the 1st International Workshop on Community Networking: Integrated Multimedia Services to the Home*, July 1994, pp. 191–8.

[25] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. "RSVP: A New Resource ReSerVation Protocol," *IEEE Network*, September 1993, pp. 8–17.