A SIMULATION STUDY OF DOCSIS UPSTREAM CHANNEL BANDWIDTH ALLOCATION STRATEGIES FOR MINIMAL USER RESPONSE TIME

APPROVED BY SUPERVISING COMMITTEE:

Dr. Rajendra Boppana, Chair

Dr. Kay Robbins

Dr. Turgay Korkmaz

Accepted:

Dean of Graduate School

A SIMULATION STUDY OF DOCSIS UPSTREAM CHANNEL BANDWIDTH ALLOCATION STRATEGIES FOR MINIMAL USER RESPONSE TIME

by

RICHARD MURPHY, BA

THESIS
Presented to the Graduate Faculty of
The University of Texas at San Antonio
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT SAN ANTONIO
College of Sciences
Department of Computer Science
December 2004

ACKNOWLEDGEMENTS

I owe the successful completion of this graduate degree to the incredible patience and guidance I have received from my thesis chair, Dr. Rajendra Boppana. His encouragement kept me going when my inspiration grew weak. I would also like to thank Dr. Turgay Korkmaz and Dr. Kay Robbins for serving on my thesis committee and for providing some of the most outstanding courses I have taken in graduate school.

Two other individuals in the UTSA Computer Science Department have played vital roles in helping me through many technical and bureaucratic hurdles. Ms. Kirsten Corda kept me on the right course through the whole process and I am sure many other graduate students required her spirited assistance. Mr. Saman Desilva maintained the critical computing resources I needed and helped me fight problems with the C++ compiler, CSIM libraries, and NIS+ refusing me authentication.

Of course my wife Mary Bancroft has provided the most critical support during my extended career as a part-time graduate student. She has my deepest gratitude for standing by me through thick and thin. My son Spenser is an inspiration as well. It would have been fitting if we could have received our graduate degrees together. Also, I extend my thanks to Phil Jackson for his encouragement and support of this research by sharing his technical contacts in the broadband Internet industry.

Finally, I owe a great deal of gratitude to Southwest Research Institute, my employer of the past 18 years. I could not have finished this degree program without their continual tuition support. My supervisor, Mr. James Hokanson, and my good friend and network manager Mr. Edgardo Perez have been generous in their support. They, too, have my special thanks.

December 2004

A SIMULATION STUDY OF DOCSIS UPSTREAM CHANNEL BANDWIDTH ALLOCATION STRATEGIES FOR MINIMAL USER RESPONSE TIME

Richard Murphy, MS
The University of Texas at San Antonio, 2004

Supervising Professor: Rajendra Boppana

The DOCSIS 1.1 protocol for data distributed over CATV networks allocates bandwidth on the upstream channel on demand. A key component of a DOCSIS network is the software that produces the Mini-slot Allocation Packet (MAP) for use by cable modems to send their data to the Internet Service Provider (ISP). The DOCSIS standard leaves the issue of MAP creation strategies open. Also, certain optimization features, such as data fragmentation and piggybacking, are not evaluated thoroughly in the literature. In this thesis, I develop a detailed DOCSIS simulator and use it to evaluate various MAP creation strategies, alternative techniques to fragmentation, and the effectiveness of piggybacking. My results indicate that (a) for sufficiently large size MAPs, performance is not greatly influenced by the allocation strategies, (b) the efficiency of fragmentation without the overhead can be obtained by using a simple change to the cable modem bandwidth request, and (c) piggybacking is effective only for highly active users or in congested networks.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

Presently there are two prevalent technologies used to bring high-speed Internet access to the home: the cable television network and the Digital Subscriber Line (DSL). Both have widely displaced the older methods of dial access using modems and Basic Rate ISDN lines due to their cost/performance advantage [11]. DSL is provided by telephone companies and tends to offer individual bandwidth for each user but is much more constrained by distance than cable modem networks. Cable modem network implementation is based on the DOCSIS standard [1]. The DOCSIS standard describes channel bandwidth sharing among multiple users with the Cable Modem Termination System (CMTS) controls the use of scarce bandwidth resources. In North America the number of users of cable TV networks for Internet access is at least 16 million [5][7], but fierce competition from DSL and future wireless MANs necessitates understanding the behavior of the DOCSIS [1] protocol and its vendor-specific implementation aspects.

DOCSIS CATV networks usually run on existing cable plants by utilizing a pair of channels, one for the downstream and one for the upstream. Because most CATV plants were originally designed to deliver information in one direction only, the signal characteristics lead to a large amount of bandwidth being available in the downstream and a much smaller amount being available in the upstream direction. These signal characteristics are the need to minimize group delay through the upstream diplex filters and the difficulty of controlling the gain (and noise) between many transmitters (cable modems) and the headend [19]. The typical propagation distances in CATV networks make contention-based MAC protocols very inefficient. So DOCSIS uses a bandwidth reservation protocol in which contention plays a role mainly when the offered load is low. The Cable Modem Termination System (CMTS) orchestrates bandwidth allocation, which is simple in principle but complex when performance needs of cable modems are a factor.

The DOCSIS standard deliberately left undefined the precise way that bandwidth is allocated on the upstream channel. Ostensibly, the vendors of CMTS are to implement clever algorithms to produce optimal network performance from the user's perspective. In reality, CMTS vendors provide a small set of parameters that can be varied by the network administrator without a great deal of explanation on which parameters to vary to achieve a desired performance target or how the parameters interact. Most of these parameters involve tuning the RF characteristics of a channel, not the MAC layer performance. There is even a small side industry of third-party products that are intended to assist the network administrator in this opaque task [8]. One large vendor of CMTS, Cisco Systems, provides a complex document of guidelines using Erlangian analysis [3], but it is mainly useful on networks with persistent streams like Voice over IP (VoIP). Thus the performance dynamics of DOCSIS networks can only be understood through the rather sparse research literature.

It is also important to understand DOCSIS performance because one of the original protocol design assumptions is now commonly contradicted. The old assumption was that home users were mainly consumers of content found on the Internet and the asymmetry of DOCSIS networks fit well with this usage pattern. However, the popularity of peer-to-peer applications, such as Napster and Gnutella, suggests that some users may provide as much content as they consume, which elevates the importance of the upstream channel performance. DOCSIS 2.0 addresses the issue of constricted upstream bandwidth with higher density modulation choices and a form of wave division multiplexing (WDM), but it is not widely deployed and many physical CATV plants are not capable of supporting its features.

1.1 Motivation

There are several design issues that are not explored in literature. Before highlighting these issues, I must introduce a key protocol construct called the Mini-slot Allocation Packet (MAP), which defines how each future time slot on an upstream channel may be used. See Section 2.2.3.3 for a detailed explanation of the MAP.

- MAP creation is undefined, but it is unclear whether the DOCSIS specification is flexible enough to support performance enhancement through MAP creation strategies.
- DOCSIS 1.1 supports upstream fragmentation, but the high reassembly overhead inspires the search for a reasonable alternative to fragmentation.

In the realm of performance, DOCSIS again has two issues motivating this work:

- The design assumptions of the protocol are commonly violated through the use of peer-to-peer applications. Here the issue is whether DOCSIS is robust and fair when a few users attempt to monopolize the upstream bandwidth.
- The DOCSIS piggybacking mechanism is intended to improve performance under high offered loads. However, the effectiveness of piggybacking and the conditions for effectiveness are not clear.

The literature on DOCSIS research is described in detail in section 2.3 and a prominent gap is apparent. There is no research into the mechanics of MAP creation and how this affects performance, mainly because this topic is proprietary and therefore deliberately never described in product specifications. Additionally, the fact that DOCSIS is a mature protocol may discourage research on the topic of protocol enhancement.

1.2 Contributions of This Thesis

This effort is intended to provide a number of concrete benefits in the area of DOCSIS performance research:

- A DOCSIS simulator. With the exception of the CableLabs sponsored OPNET™ model, none of the other simulations appear to support the level of parameterization developed here. None of the existing models are described in any detail in the literature, so it is difficult to use them for other research efforts. The model developed for this work is intended to be immediately useful as a DOCSIS research tool and to be extensible so that DOCSIS 2.0 can eventually be modeled.
- Determination of possible efficiency improvements and overhead reduction in the protocol implementation.
- A MAP creation strategy that provides enhanced response time and utilization over competing strategies.
- A study of how peer-to-peer application users affect the mean response time and utilization of a DOCSIS network.
- Discovery of whether the piggyback feature is beneficial across a range of traffic loads..

1.2 Organization of Thesis

In this thesis I present a review of the relevant research literature in Chapter 2, along with a detailed introduction to the DOCSIS protocol. The simulator is discussed in Chapter 3, with a complete listing of the control parameters appearing in Appendix B. Chapter 4 covers the design and operation issues in DOCSIS that warrant examination for potential performance improvements. Chapter 5 contains the results of my simulation experiments and a discussion of the factors underlying the results. Chapter 6 is a summary of my research rational and results, some suggested directions for future work with the simulator, and additional future research topics.

Chapter 2: Background and Related Work

Media Access Control (MAC) protocols are part of the link layer in the classic seven layer International Standards Organization (ISO) network model [52]. MAC protocols may be subdivided into three classes: a) Fixed Assignment Multiple Access (FAMA), b) Random Access, and c) Demand Assignment Multiple Access (DAMA). A common FAMA protocol is Time Division Multiple Access (TDMA). For TDMA protocols with fixed bandwidth assignments for each station, the critical fault is one of poor channel utilization. Because of this, static bandwidth assignments are rarely used in computer networks. Typical random access MAC protocols such as Carrier Sense Multiple Access with Collision Detection (CSMA/CD) do not work well when propagation delays are large. For CSMA/CD and ALOHA type protocols, large packet propagation times introduce high latency into every media access. When using fixed contention intervals to send data, stations closer to the base station have an advantage over those further away, which is another reason CSMA/CD or CSMA/CA type protocols are inappropriate for networks with large propagation delays. Therefore, DAMA protocols are used for networks with large propagation delay.

2.1 The DAMA Family

DAMA protocols attempt to avoid the problems of FAMA and random access protocols by requiring stations to explicitly request bandwidth and by limiting potential collisions to the requests themselves. DAMA protocols tend to be used in networks that use satellite links and wireless transmission between a base station and user stations. Typically, the base station can transmit to many user stations by using a broadcast channel, whereas user stations must share a more bandwidth-limited channel when sending data the other direction. This is the exact nature of Hybrid Fiber Coaxial (HFC) CATV networks. A key assumption made in DAMA protocols is that the network consists of clients and servers. This implies that individual stations (clients) do not need to communicate directly with each other but only with some set of servers located outside the DAMA network. Data sent from the central server to the clients is said to flow in the *downstream* direction and from the clients to the server is the *upstream* direction. A general characteristic of DAMA networks is that of asymmetric bandwidth, with the downstream usually having more bandwidth than the upstream. These characteristics require the MAC protocol mechanisms to be very different depending on the direction, but both mechanisms are usually described under a single standard.

DOCSIS 1.0, the original MAC protocol standard adopted by cable TV companies, and the newer DOCSIS 1.1 belong to the DAMA class of protocols. IEEE standards such as 802.14 for CATV networks and 802.16 for wireless broadband networks are other examples of DAMA protocols. I will describe DOCSIS 1.1 first and compare it with IEEE 802.14 and 802.16 protocols.

2.2 DOCSIS 1.1 Overview

DOCSIS was designed to transport data over a physical infrastructure that was implemented for delivering broadcast television. These CATV plants provide a large amount of bandwidth to households (downstream) and much less bandwidth in the reverse direction (upstream). Some older networks use modems and dial-up links for the upstream transmission because of the poor signal characteristics on the upstream CATV plant. The physical topology is branch-and node, which can be modeled as a tree with one to six branches. An example topology with four branches is shown in Figure 2-1. The headend is called the Cable Modem Termination

System (CMTS) and the user nodes cable modems. More complex trees are built using layer 3 forwarding between the root nodes of these basic trees.

Modern CATV networks have replaced the branches closest to the root with fiber-optic media. Hence the name “hybrid fiber-coaxial” is applied to most CATV networks. Coaxial cable is only used in the last segment, which serves 100 to 200 houses.

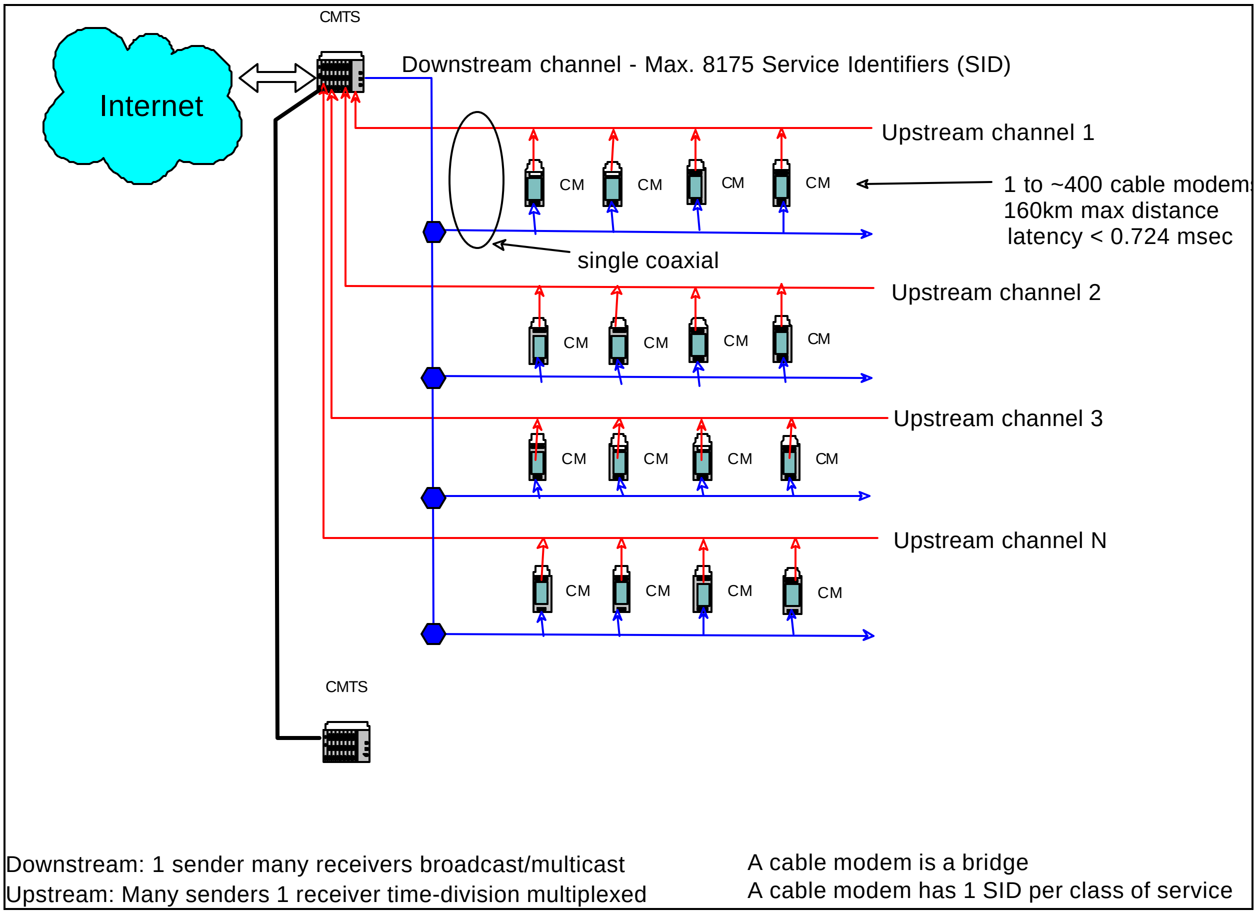


Figure 2-1. DOCSIS 1.1 Network Model

2.2.1 Physical layer

Because of the use of legacy CATV technology and the fact that most CATV networks were originally intended to be one-way networks, the physical layer is complex in terms of modulation, signal-to-noise limits, transmission guard times, and spectrum use. There are different modulation schemes for upstream and downstream transmission, these being 16QAM or 256QAM downstream and QPSK or 16QAM upstream. Unlike many of the IEEE 802 protocols, DOCSIS supports several different data rates (see Tables A-1 and A-2 in Appendix A). The intent was to allow DOCSIS to run on both antiquated CATV plants and newer plants designed for two-way traffic without forcing "least common denominator" performance on those willing to invest in modern technology. Specific details of the physical layer can be found in the DOCSIS specification [1]. For this research, the main physical layer attribute of interest is that of asymmetric bandwidth on each simplex link. Also, due to signal-to-noise ratio restrictions, Forward Error Correction (FEC) using Reed-Solomon encoding is optionally used on the upstream. The downstream uses interleaving to minimize the damage caused by burst errors. The interleaving level is a CMTS parameter. The use of FEC decreases the bandwidth available for user data and interleaving increases the latency of each transferred protocol data unit (PDU).

DOCSIS defines a transmission convergence layer for the downstream that may be considered part of the physical layer. This layer consists of an endless sequence of 188 byte frames using ITU-T H.222.0 MPEG encapsulation. The purpose of this convergence layer is to allow digital TV frames to be interleaved with frames carrying data on the same downstream channel. MAC layer frames are not synchronized with these MPEG frames. A MAC frame may start anywhere within an MPEG frame and may span several if needed.

Since the upstream and downstream channels are simplex, each is incapable of supporting certain layer 2 protocols, such as IEEE spanning tree or distributed queue protocols like DQDB. The CMTS implements an abstract entity called the MAC forwarder that allows packets to be sent between cable modems without traversing the entire CMTS protocol stack.

2.2.2 Data Link Layer: Media Access Control (MAC) and Logical Link Control (LLC) Sublayers

The data link layer is usually responsible for link-to-link addressing, media sharing, flow control, data framing, and error detection and correction. This layer provides mechanisms for most of the interesting characteristics of DOCSIS. The functionality of the data link layer is divided into two sublayers: Media Access Control (MAC) and Logical Link Control (LLC). In this section I will discuss these sublayers concurrently because their functions are closely interwoven. Specifically, with DOCSIS these mechanisms provide for bandwidth requests and allocations, service differentiation, and bandwidth efficiency through the use of variable length packets. Each cable modem is identified by one or more Service Identifiers (SIDs), with each SID identifying a service class between the cable modem and the CMTS. A SID serves as an LLC identifier and, because they are unique on a given CMTS, they function like a MAC address.

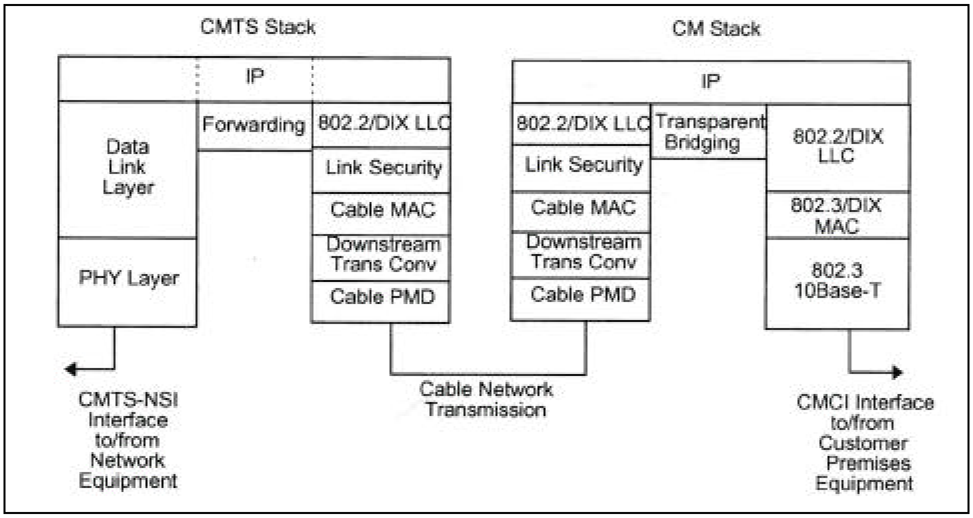


Figure 2-2. DOCSIS Protocol Stack

Figure 2-2 shows the relationships between the different sublayers within both the CMTS and cable modem protocol stacks. The LLC sublayer defines the data PDUs carried by DOCSIS are to be complete Ethernet packets or, optionally, ATM cells. In practice, only Ethernet packets are used. There is also a link security layer within the LLC sublayer, but its use is optional and it is not considered in the simulator described in Chapter 4.

2.2.2.1 Downstream Transmission

DOCSIS defines a set of MAC frame headers that are used to classify frames containing data and those containing various kinds of management messages. Only the data headers are common to both downstream and upstream transmissions. The downstream direction is broadcast media in the sense that all cable modems see all traffic. SIDs are used to address specific cable modem service points and a special SID represents a multicast address.

Downstream transmission is much simpler than upstream transmission, both conceptually and in implementation. The downstream direction supports higher date rates than the upstream direction, but because the CMTS is the sole sender to many cable modems, queuing latency may occur in the CMTS. Because downstream data follows the same physical paths as upstream data, the propagation delay is essentially the same in both directions.

2.2.2.2 Upstream Transmission

The upstream bandwidth is shared among cable modems with the CMTS controlling the bandwidth allocation. Upstream transmission is specifically a form of statistical time division multiple access (TDMA), where time is divided into “mini-slots” of 6.25µs x 2^n (n=1 to 7). Because of the different upstream symbol rates and modulation methods, a mini-slot can carry 1 to 1024 bytes. However, large sizes are wasteful of bandwidth because a mini-slot is the smallest bandwidth allocation quantity. In practice, mini-slots carry 8 or 16 bytes. All mini-slots in the upstream channel are assigned to one of six uses by the CMTS: initial maintenance, station maintenance, bandwidth request, immediate data, short data, and long data. The first four of these are subject to contention, while the latter two are reserved based on requests for bandwidth. The CMTS communicates these assignments to each cable modem via the downstream channel with a special MAC layer packet called a Mini-slot Allocation Packet (MAP). This process, along with the MAP physical and logical structure, is illustrated in Figure 2-3. The portion of Figure 2-3 labeled "upstream" shows the division of time on an upstream channel across two sequential MAPs. Because all cable modem clocks are synchronized to the CMTS clock, each advances its mini-slot usage by its propagation delay to the CMTS so that data sent upstream will arrive at the CMTS during the expected mini-slots.

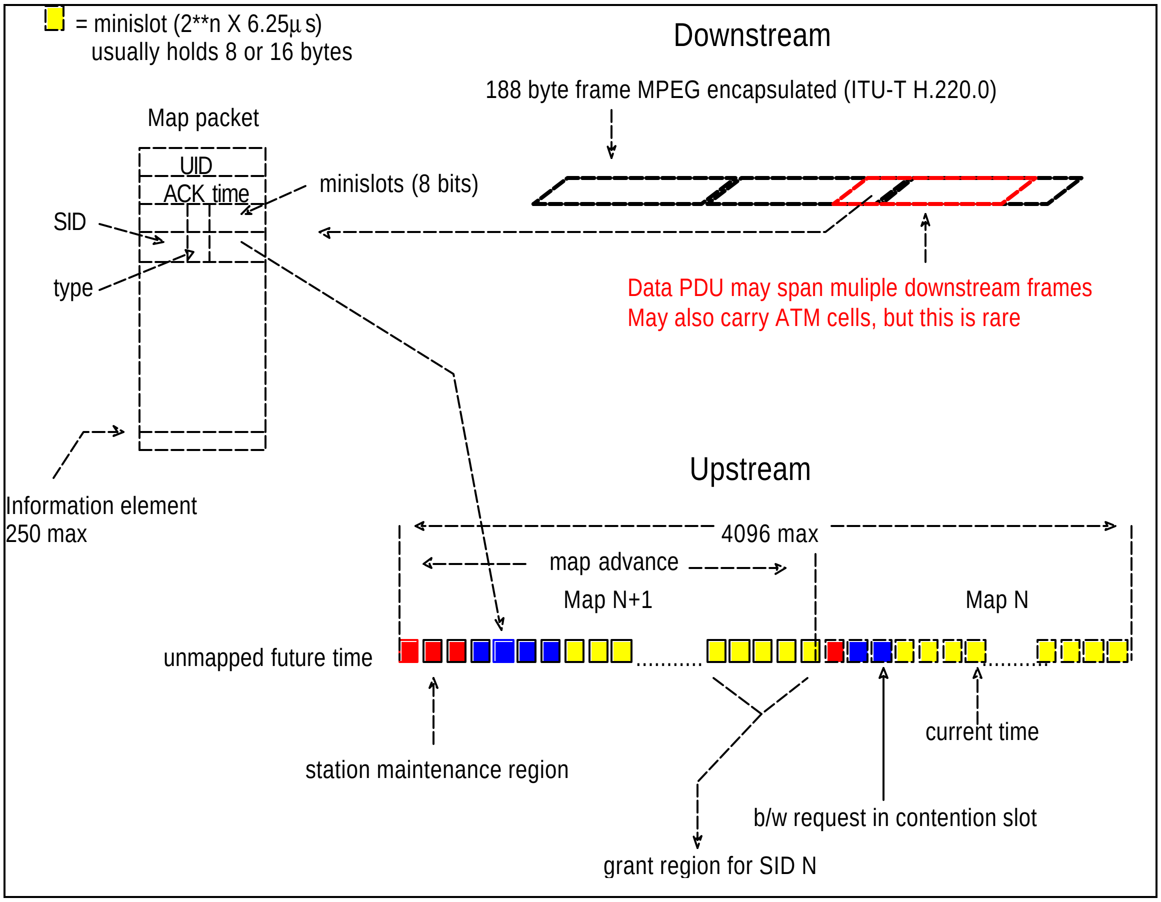


Figure 2-3. Logical View of Steady-State Upstream and Downstream

There are two additional features that make the use of upstream data mini-slots somewhat more complicated: concatentation and fragmentation. Concatentation, available in DOCSIS 1.1 and earlier versions, allows a cable modem to put more than one PDU into a data grant region. Here, a PDU is the entire Ethernet frame containing the packet to be sent upstream. Fragmentation, available in DOCSIS 1.1 and later versions, refers to the reverse situation. In this case a cable modem may place only a portion of a PDU (along with a special header) into a grant region. Concatenation tends to be the default behavior of cable modems, since it appears in all DOCSIS versions. The CMTS must enable fragmentation on cable modems that support it because a given upstream channel may have a mixture of DOCSIS versions within the cable modem population.

2.2.2.3 Creation of MAPs

One of the most complex tasks performed by the CMTS is the construction of the bandwidth allocation MAP for each upstream channel. Obviously, there are numerous scheduling algorithms that can be applied to the bandwidth requests, such as FIFO, smallest first, and round-robin. There are also numerous strategies that can be applied towards creating the various types of contention slots. Some, such as unsolicited grant service and real-time polling service [1], are intended to enforce quality of service (QoS) guarantees, which are available in DOCSIS 1.1. The protocol specification intentionally avoids making any sort of recommendations toward MAP generation algorithms because this area is assumed to be available for different vendors to distinguish their CMTS products. Clearly, even a simple FIFO scheduling discipline can be enhanced by various policy parameters in order to limit latency and other factors. Section 2.3.1 covers research related to MAP creation.

DOCSIS 1.1 does specify some limitations that affect the creation of MAPs. No MAP may contain more than 240 Information Elements (IE), where an IE is a set of data fields that describe a particular transmission region, its purpose, and which cable modems can use the region. The sum of all active[1] MAPs cannot describe more than 4096 mini-slots, thus limiting the amount of storage a cable modem must allocate to hold MAP information. Additionally, since the bandwidth request header provides only 8 bits for the bandwidth, no more than 255 mini-slots may be requested at one time.

2.2.3 Comparison to Ethernet

DOCSIS 1.1 is a much more complex protocol than Ethernet primarily because each cable modem must maintain bandwidth request and allocation information and the allocation of bandwidth involves communication with a central authority, the CMTS. Another complicating factor is that DOCSIS essentially uses two different MAC schemes, one in the downstream direction and one in the upstream direction. In addition, the implementation of DOCSIS is complicated by the open-ended nature of MAP generation. Table 2-1 condenses the comparison of DOCSIS 1.1 and Ethernet. The next section compares DOCSIS to an equivalent DAMA protocol, IEEE 802.14.

---

[1] Active MAPs describe future time slots and they have been sent by the CMTS downstream.

Table 2-1 Comparison of DOCSIS 1.1 to Ethernet

Feature	Ethernet (10/100 Base X)	DOCSIS 1.1
Maximum prop delay	22.5/2.25µs	724 µs
Management messages	None (except for jam)	Many
Bandwidth symmetry	Symmetric	Asymmetric
Bandwidth allocation	Probabilistic	1st probabilistic, subsequent queued data deterministic
Bandwidth allocation	Decentralized	Centralized
Maximum PDU	1500 bytes	1500 bytes
Error detection	CRC	Interleaving
Station-to-station communication	Peer-to-peer	Always indirect via CMTS
Time	Asynchronous, relative to each station	Synchronous, relative to CMTS
Addressing	Explicit, broadcast, multicast	Implicit in upstream Explicit, broadcast, multicast in downstream
Address length	48 bits	12 bits
Number of stations sending within one RTT	one	many

2.2.4 Comparison to IEEE 802.14

The IEEE 802.14 protocol has never replaced DOCSIS due to its reliance on ATM and the ensuing complexity. DOCSIS acquired too much market penetration and the ever-delayed 802.14 standard offered no compelling features to overcome the dominance of DOCSIS. While the IEEE committee wanted to use ATM to more naturally support QoS and traffic classes, the designers of DOCSIS deferred these features to DOCSIS 1.1. As a practical matter, there are no 802.14 compliant products on the market. This discussion is included here in order to better understand the research literature, as the purpose of 802.14 has devolved into a simulated protocol to compare with DOCSIS. Most of this prior work has involved comparisons of the relative efficiency of each collision avoidance algorithm. Table 2-2 gives four key differences between DOCSIS 1.1 and IEEE 802.14.

Table 2-2 Comparison of DOCSIS 1.1 to IEEE 802.14

Feature	**IEEE 802.14**	**DOCSIS 1.1**
Encapsulation	ATM required (non-standard cell format)	ATM optional
Collision avoidance	Ternary tree	Binary exponential backoff
Connection orientation	Both	Connectionless
Maximum round-trip delay	0.4ms	0.8ms

However, the IEEE did specify a DAMA protocol that has been implemented, although for a radically different type of physical network. The IEEE has formalized the definition of a DAMA protocol for wireless broadband, which is designated the *802.16-2001* standard. This protocol is partially derived from MCNS and DOCSIS 1.x. It represents a super-set of features, so in this sense DOCSIS 1.x is a special case of 802.16-2001.

Before examining the body of research involving the DOCSIS protocol, it is helpful to briefly look at some interesting features of the upstream MAC protocols that were considered by the IEEE 802.14 committee. With only one or two exceptions, these proposed protocols have a large overlap with both DOCSIS and IEEE 802.14 in the upstream MAC layer. Most were intended to carry more than one class of traffic, going beyond the best-effort only approach of DOCSIS 1.0. The next table lists key characteristics of the upstream channel and their disadvantages. The reader is referred to [14] for an exhaustive review of each competing protocol.

Table 2-3 Pros and Cons of MAC layer features

Upstream MAC feature	**Disadvantage**
Bandwidth request includes # of slots and # of future maps with the allocation	Larger bandwidth request headers, more complex scheduling of grants
Pipeline polling & plain polling	Wasted downstream bandwidth, loss of cable modem asynchronicity.
Priority increment on collision	Partially adopted in 802.14, more CMTS complexity
ATM cells	Good for fragmentation, but high overhead
Explicit NAKs	Uses downstream bandwidth and makes cable modem more complex
Spatial domain resolution	Complex registration and ranging process
Virtual upstream channels	Wasted bandwidth if channels can not be adapted to traffic load or classes

These rejected protocol features provide guidance by allowing us to avoid mini-slot allocation strategies that have already been found poor.

2.3 Related Work

DOCSIS is probably the most widely used protocol with the smallest body of published research because of the difficulty in simulating this complex protocol, assumptions that must be made about MAP generation, and the lack of available data on the performance of actual networks. Since all DOCSIS networks are owned by private companies locked in fierce competition with DSL providers, performance data is likely considered to be highly-sensitive. In contrast to Ethernet, there are no protocol analyzers available for recording traffic traces from a DOCSIS network.

In this section, I have classified the research literature into six areas: CMTS design, TCP behavior over CATV networks, MAC protocol design, contention algorithm analysis, DOCSIS performance with special traffic classes, and fault recovery. The work described in this thesis falls in the CMTS design area, although one of my proposed allocation strategies involves a minor modification to the MAC protocol.

2.3.1 Algorithms for the CMTS

Most of the relevant research in this area falls under two subtopics: request scheduling disciplines and mini-slot allocation schemes. Abi-Nassif et al [12] describe an offered load estimator based on a weighted average of collision outcomes over a sliding window of MAP intervals. Such an algorithm allows the CMTS to base future contention slot allocations on the estimated future offered load on each upstream channel. Such a scheme avoids the waste of bandwidth due to the allocation of contention slots that are never used. However, it is useful only under moderate loads because, at higher loads, the DOCSIS piggyback feature lowers the need for contention slots. Detailed simulations comparing 802.14 and DOCSIS request scheduling and collision management are done by Limb and Sala [38]. The three scheduling algorithms studied indicate a divergence in performance only when the offered load is greater than 0.55. This research suggests the use of partial grants to increase the chance of piggybacking subsequent bandwidth requests. Another approach to mini-slot allocation is found in a paper by Sala [45], where the concept of a “sea of mini-slots,” as opposed to MAPs, is introduced.

Much of the research into MAC scheduling algorithms involves supporting QoS-specific traffic. Perkins and Gatherer [43] use a flow-based approach to schedule grants and demonstrate the advantages of piggybacking in latency reduction. Sdralia [49] proposes that prioritized first-come first-serve (p-FCFS) scheduling be the performance base against which all CMTS request scheduling algorithms are measured. This scheme is modeled using on-off sources sending short (100 byte) and long (1500 byte) packets, but no case is made that this sort of traffic pattern approximates any actual traffic. Sdralia, along with Limb [38] and Cisco Systems [6], provides a good analysis of the performance ranges one can expect on DOCSIS upstream channels. DOCSIS behavior during isochronous traffic loads is studied in [53]. This work covers the special case of low-bit rate isochronous traffic. Xiao [56] covers the mixing of isochronous and bursty data traffic on Multimedia Cable Network System (MCNS[2]) networks. A Markovian analysis of upstream traffic results in an elaborate algorithm for computing feasible regions that minimize the loss probability of a voice packet while keeping the average delay of data packets less than some value t. Xiao assumes an upstream data rate of 10.24Mb/s, which is the maximum possible rate under 16QAM modulation. This unrealistic assumption, plus the complexity of the proposed algorithm, makes actual implementation of such a scheme very unlikely. Yin [57] points out that the Abi-Nassif [12] mini-slot estimation scheme assumes that collisions are negligible, which may not always be true. However, this different estimation method only stands out when the offered load is greater than 0.8 and when looking at the mean number of contention cycles and request/grant latency.

Clearly, there are many request-queue scheduling disciplines that could be studied, but the results would not be unique to DOCSIS networks. On the other hand, the allocation of contention slots and even the arrangement of the different transmission regions within MAPs offers practical performance improvement for minimal CMTS overhead. This issue will be addressed in this thesis.

2.3.2 TCP Behavior Over CATV Networks

After DOCSIS networks became common, research in the area of TCP behavior over such networks arose. Some estimates of Internet traffic classify ~80% of it as HTTP, which of course depends on TCP. An early paper by Chatterjee [17] explores the effects of asymmetric

---

[2] DOCSIS precursor.

bandwidth and the MAC protocol on both interactive and bulk transfer TCP sessions. The main effect is that the TCP window size tends to be too small due to asymmetric bandwidth skewing the bandwidth-delay product. He also notes that interactive applications suffer more from latency since they are using contention slots instead of piggybacking bandwidth requests.

TCP performance is more likely to suffer due to oversubscription of the upstream channel and traffic limiting done by the cable modem itself. Most cable modems download a profile (operational parameters) from the CMTS when they complete their initial ranging and registration. This profile can set traffic shaping and bandwidth limiting parameters to be enforced by the cable modem. Cohen [20] models an asymmetric network that loses packets due to transmission errors instead of congestion. By modeling several different scheduling algorithms, Elloumi [26] attempts to find one that is optimal for minimizing the TCP "slow ACK" problem caused by an asymmetric network. The model is actually for 802.14 rather than DOCSIS. One of the few works to address downstream issues is the Master's thesis by Kaza [35], where he analyzes TCP performance over a lossy downstream channel. His model is particularly thorough, covering unsolicited grants (USG), USG-AD, real-time polling service, and committed information rate (CIR) service. Kaza's results indicate that the DOCSIS MAC protocol has a minor effect on the TCP round-trip time (RTT) estimation, but a lossy downstream channel will produce serious effects. This result is of interest to the operators of older CATV networks where the transmission characteristics may be poor.

Since the bandwidth delay product is important when setting TCP window size, it is useful to look at how this quantity varies in DOCSIS networks. Assuming a maximum network diameter, approximately 116 mini-slots can be in transit. For the highest bandwidth upstream channel with the largest (and very impractical) mini-slot size, the bandwidth-delay product is about 118K bytes. For a more typical network it is around 928 bytes. Clearly the window and buffer size requirements are not extreme.

2.3.3 MAC Protocol Comparisons

A broad area of the literature involves comparing DOCSIS to 802.14 or some variation of 802.14. Included in this literature are survey articles intended to explain the salient features of the protocol, most of which were published before DOCSIS was widely adopted. This body of work is useful in understanding why DOCSIS was adopted and 802.14 was not.

Laubach [37] gives an excellent historical review of the development of 802.14 and DOCSIS, along with a functional description of the protocols. Fellows [27] covers DOCSIS in detail, including link encryption, QoS features, and interaction with Internet Group Management Protocol (IGMP) for multicast support. A survey of the protocol standards activity for HFC networks circa 1998 is covered in [54]. Considering that the protocol specification itself is large and full of physical layer and transmission engineering details, papers like [37] are invaluable to an understanding of the protocol behavior, if not the minutiae.

A DOCSIS-like MAC protocol is described in a very early paper by Limb [38]. Here one can see the prototype of MCNS being conceptualized and studied through a simulation. This protocol has no MAPs; grant regions and acknowledgement slots are interleaved with downstream data. There is no convergence layer in this protocol. Unlike DOCSIS, all requests receive a specific acknowledgement ahead of any grant and the collision resolution method is p-persistent with p = 0.2. DOCSIS clearly addressed the short-comings in this proto-protocol. One of the best side-by-side comparisons of DOCSIS and 802.14 is found in [40]. This paper is exhaustive in comparing each feature of the protocols by analyzing the protocols rather than simulating them. The author does identify research topics, namely performance, optimal allocation of request region mini-slots, and the scheduling of grant region mini-slots. Another comparison of 802.14 with DOCSIS focuses on the physical layer differences and the different collision resolution algorithms [43]. Sriram [51] shows analytically how ATM and variable length PDUs compare on HFC and wireless networks. The results give a clear justification for implementing DOCSIS over 802.14, which uses ATM cells as PDUs. Variable length PDUs waste much less bandwidth.

2.3.4 Analysis and Comparison of Contention Algorithms

Research in this area involves either comparisons of 802.14 and DOCSIS or simulations of novel contention resolution approaches. Golmie [29] covers the contention algorithm for 802.14 and describes the implementation issues around it and several alternatives. Sala [45] does a complete analysis of p-persistent and tree-based contention algorithms. While this paper is not specifically about DOCSIS, it does show that, at most, the throughput varies ~4% across all the methods simulated. Thus ease of implementation becomes the discriminator and p-persistent methods are to be preferred. Perhaps the ternary-tree method used in 802.14 is another reason this protocol was never implemented.

2.3.5 DOCSIS Performance Under Special Traffic Classes

DOCSIS 1.0 has no support for different classes of traffic. Streaming video, voice-over-IP, interactive multi-user games, and other specialized applications became popular and DOCSIS 1.1 has provided extensions to better support such traffic. Hence, for inelastic delay-intolerant traffic, the CMTS can provide a fixed unsolicited grant to a unicast SID at regular intervals. For inelastic delay-tolerant traffic, a unicast request region is provided to a specific SID at regular intervals. Elastic traffic is handled through the default request/grant mechanism. Some of the prior research involves modeling the behavior of DOCSIS 1.1 under various classes of traffic, mostly to explore fairness issues and whether DOCSIS can indeed provide QoS transport. The literature generally shows that DOCSIS 1.1 can deliver different traffic classes without performance degradation [56]. Chu [18] and Rabbat [44] describe schemes for providing service differentiation.

The research described in this thesis involves only a single class of “best effort’ elastic traffic. The simulator has the ability to generate CBR traffic concurrently with random traffic, but unsolicited grants are not implemented. The simulator can produce unicast request regions, but this is only used in verification mode. This functionality could be easily added, however.

2.3.6 Fault Recovery, Overviews, and Management

The last body of research surveyed here covers both very general and very specific protocol topics. Sdralia et al [50] simulated the behavior of DOCSIS under fault recovery conditions. This work examined what happens on a DOCSIS network after a power failure when several hundred cable modems all attempt to re-register and obtain CMTS time synchronization. The best overviews of DOCSIS and how the protocol is expected to work are given by Fellows [27] and Laubach [37]. Laubach touches on the issue of fragmentation being left out of DOCSIS 1.0, which is related to a research area of this thesis. The topic of management of DOCSIS networks via SNMP is covered in a thesis by Schnitzer [47]. His work is most relevant to performance measurements in DOCSIS networks.

2.4 Comments on Research Simulations

In the entire body of research examined for this thesis, 17 papers included simulations of networks. Within these simulations, seven were done by OPNET™, five were specially written for the research, two used the NIST ATM simulator, one used NS, one used Parsec, and one was unspecified. Almost no information was provided on the simulation input parameters, model verification, confidence intervals, assumptions, and many other factors that would allow duplication of the simulation results. This was one reason a simulator was developed in CSIM [48] for this work. None of the published research describes a simulation tool that could be used for further independent research on DOCSIS networks.

Chapter 3: The DOCSIS Simulator

One of the goals of this work is to produce a public domain simulator that faithfully models the DOCSIS MAC layer and facilitates implementation and evaluation of new features. While CableLabs has sponsored the development of a large and complex model using the OPNET package, it is available only to members of CableLabs. To address this deficiency, I developed a DOCSIS simulator in C++ with the CSIM simulation package. CSIM is a simulation library that handles the scheduling of events and the collection of statistics. CSIM is an ideal modeling framework for complex protocols. The use of object-oriented programming and C++ facilitates the implementation of network entities that maintain state and exhibit complicated interactions with other entities. Using CSIM has the additional advantage of allowing fine control over the simulation overhead since there are no predefined, unchangeable protocol modules. As one example of how this is important, this model uses preallocated data structures to represent data packets, which leads to a fixed number of calls to the C++ *new* operator regardless of how long the simulation is run.

3.1 Simulator Layout and Algorithms

The simulation program is organized as a tree of process containers in which the top-level CSIM process performs the parsing of the network configuration and control parameters and then creates the CMTS process. This process then creates an upstream process, time-division multiplexer process, and data sink process for each upstream channel. It also creates and distributes the specified number of cable modems on each channel and links each one with the upstream and downstream data queues. These queues are implemented as CSIM mailbox objects.

The cable modem process consists of three subprocesses: (1) the customer premise equipment (CPE), which provides data (PDUs); (2) the upstream channel MAC process; (3) and the downstream channel cable modem (CM) process. This division is required because of the asynchronous interactions between the upstream, downstream, and CPE data flows. The MAC process is the most complex because it implements the DOCSIS upstream state-machine. This involves tracking transmission opportunity deferrals, grant request status, and the size of the upstream transmission queue.

3.1.1 Physical to Abstract Mapping

The mapping of physical network entities to simulator abstractions is most easily understood by referring to Figure 3-1. All of the physical entities are implemented by at least one abstract entity. The upstream process represents what the DOCSIS specification calls the "MAC forwarder." This process, along with the MAP builder and CMTS processes, emulates a real-world CMTS. The CM and MAC processes emulate a cable modem and the CPE process plays the role of a PC attached to the cable modem. The TDM and data sink processes have no real-world analog and are constructs that simplify the simulator implementation. There are additional logical processes not shown that control the simulator, collect data, and manage internal data objects.

3.2 Assumptions and Limitations

As in most simulations, various assumptions are necessary in order to make the model implementation tractable. These assumptions, along with resource limitations like CPU cycles, in turn impose limitations on the range of possible physical networks that can be modeled.

Assumptions:

- MAC addresses are already known (no ARP requests).

- Ranging and modem registration is already completed when the simulation starts.

- Cable modems may act as “on-off” sources of traffic, but they do not leave the network

- MAPs are never lost on the downstream.

- The time to create a MAP is negligible.

- Packets sent downstream in response to upstream traffic do not incur delays beyond the downstream propagation and transmission delay.

- Concatenation is always enabled on the upstream.

Limitations:

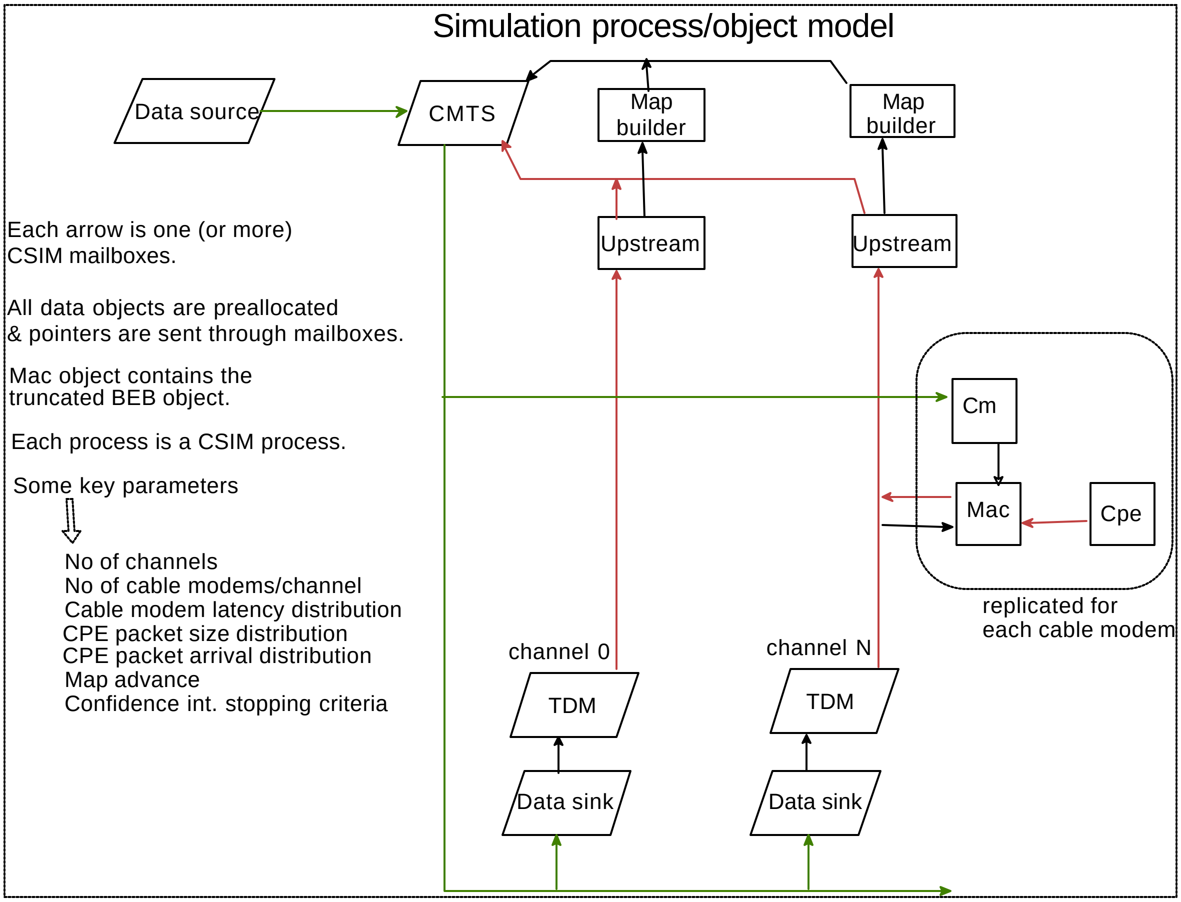
- Cable modems may not change upstream channels.
- The maximum number of packets in transit is fixed (50000), but can be changed by recompiling the simulator.
- SIDs are assigned in ascending order based on distance from the CMTS.
- Interleaving and FEC overhead are not explicitly modeled.
- A late MAP is considered an error.
- Fragmentation is not modeled.
- Cable modems can located randomly or at fixed intervals, but not at arbitrary specified distances from the CMTS
- Immediate data slots are not modeled.

- Short and long data grants are combined into one type of grant.
- Layer 3 and higher protocols are not modeled.
- QoS is not available.
- Each cable modem is assigned a single SID.

3.3 Model Capabilities

The simulator is highly parameterized so that many different network configurations can be modeled without code modifications. Appendix B covers the available parameters in detail. The simulator may be run as a daemon process so that many instances can be started in parallel using scripts. It contains an option to generate output file names based on the time of execution, which allows multiple instances to run in the same directory without overwriting output files. The user may specify from one to six upstream channels each with a different number of cable modems. The model run time is proportional to the number of cable modems, but response time convergence within a specified confidence interval can be used to stop the simulation in place of using a fixed number of batches. This model uses the process-oriented world view, as opposed to the explicit event scheduling approach.

Other features of the simulation include the ability to provide two different traffic classes simultaneously, control of the station maintenance overhead, and the ability to choose packet size and interarrival distributions.



Each arrow is one (or more) CSIM mailboxes.

All data objects are preallocated & pointers are sent through mailboxes.

Mac object contains the truncated BEB object.

Each process is a CSIM process.

Some key parameters

No of channels
No of cable modems/channel
Cable modem latency distribution
CPE packet size distribution
CPE packet arrival distribution
Map advance
Confidence int. stopping criteria

Figure 3-1 Simulation Process Model

3.4 Statistical Parameters and Instrumentation

The easiest way to understand the gathering of statistics in this simulation is to visualize global variables accumulating information at each point in Figure 3-1 indicated by an arrowhead. There are two types of counters: process counters and monitor points. Process counters track all the data entities through the model and are used to assure internal consistency. Thus, the number of data messages sent downstream will equal the number seen at each data sink process. Likewise, the number of timeslots sent upstream will match the type and number received by the upstream process. Some statistical measures are derived from these counters, like average MAP size. Other performance parameters tracked in this group include the number of contention slots skipped doing the deferral algorithm, the number of requests sent by piggybacking, and the number of grant slots passing each cable modem.

The monitor points collect data on the number, size, and times data packets are seen at various points in the simulation. These are used to compute channel utilization and response time. Response time is computed by having upstream packets "trigger" the sending of downstream packets from the CMTS. The original upstream packet has its creation time transferred to the downstream packet so that the total time to send a packet upstream and receive a response is measured at each cable modem.

3.5 Verification and Validation of the Simulator

The simulator has been verified through the use of the extensive trace capability and hand-checking the results. This verification shows that all internal data structures maintain integrity, no packets disappear or become corrupt, and no run-time errors prevent run completion. The scenarios that were verified range from a simple single upstream channel with the cable modems at fixed intervals from the CMTS to networks with three upstream channels and random cable modem distributions.

Validation was done in two steps: (1) checking the computed metrics (response time and utilization) against the expected values and (2) comparing the plotted curves to the expected curves. For example, a result derived from M/M/1 queuing analysis shows that response time is roughly $\frac{1}{1-r}$, where ? is the channel utilization. This function has a distinctive shape and the response curves produced by this simulator produce a good approximation of this shape.

Chapter 4: Improving DOCSIS - Design and Operational Issues

In this chapter I examine several aspects of the DOCSIS protocol, such as fragmentation and MAP creation strategies, and evaluate their effect on overall performance. In addition, I also compare the performance differences between a small group of heavy users and a large group of light users. My performance comparisons extend to considering a group of users containing a few rogues who place an extremely heavy load on the upstream channel. This traffic mix is intended to represent typical Web browsing users sharing the upstream bandwidth with users of peer-to-peer applications.

I start by reviewing the process by which cable modems request bandwidth on the upstream channel and how a CMTS allocates the same (see Chapter 2 for a more elaborate description). The bandwidth request process is well defined within the protocol. Cable modems must request the number of mini-slots needed to send the packets queued for upstream transmission. This request is not always precise because the request field has only 8 bits, which limits the request to 255 mini-slots. Also, since a mini-slot usually holds 8 or 16 bytes, there is a small round up error for most packets. If an older cable modem is used, it may not support concatenation. If this is the case, the bandwidth request is only for the first queued packet.

The bandwidth allocation process is not specified within DOCSIS and it is unclear how actual CMTSs implement this action. For example, Cisco Systems provides only a single configuration option for bandwidth allocation in the operating systems code for their CMTS. Except for the DOCSIS restriction on the number of mini-slots that can be assigned at any given moment (4096), there are no specific rules for either the size of MAPs or how the various types of mini-slots may be ordered within a MAP. In practice, MAPs do have size limits, typically varying from 80 to 400 mini-slots, which is somewhat dependent on the mini-slot size. When the network load is high and MAP sizes are too small, the CMTS may not allocate the requested bandwidth completely to one or more cable modems. In such cases, the CMTS may allocate only a portion of the requested bandwidth.

The rest of this chapter is organized as follows. Sections 4.1 and 4.2 are concerned with design improvements, the first concerning fragmentation and the second concerning MAP creation strategies. Sections 4.3 and 4.4 consider operational performance. Here I first discuss the issues related to the effectiveness of piggybacking. Section 4.4 covers the general issues of robustness and fairness when rogue users are present.

4.1 Fragmentation

Whenever the allocated bandwidth is less than the requested bandwidth, a DOCSIS 1.1 compliant cable modem can use fragmentation to avoid wasting bandwidth. When used, a cable modem may send partial packets, which are reassembled by the CMTS. The ability to send partial packets ensures that the amount of allocated bandwidth wasted is minimized.

While fragmentation is simple from the sender's point of view, it is a complex mechanism for the receiver (here the CMTS). Fragmentation is considered harmful by many researchers [36]. There are two major disadvantages to fragmentation that are relevant here: (a) performance suffers when fragments are lost because only the higher protocol layers (TCP) detect packet loss, but all fragments must be resent, and (b) reassembly of the packets is resource intensive and time consuming. Fragments must be buffered and a timer set so they are not buffered forever. A timer that is too short will cause the appearance of a fragment loss.

Reassembly efficiency is a reason the designers of IPv6 disallow fragmentation at intermediate routers [31]. Additionally, there is a DOCSIS specific reason why fragmentation is undesirable.

In practice fragmentation may not yield the expected benefits. DOCSIS 1.0 cable modems do not fragment, while DOCSIS 1.1 cable modems may fragment if the CMTS does not disable it. As a practical matter, both versions of cable modems can be intermixed on the same upstream channel. When this is the case, the CMTS must either disable it for all cable modems or keep track of the DOCSIS 1.1 cable modems. In the first case, bandwidth is wasted. The second case increases the bookkeeping, which again distracts the CMTS from the primary purpose of transferring user data. It is noteworthy that a CMTS may be servicing up to 6 upstream channels simultaneously. Simply issuing partial grants to cable modems that cannot fragment would be a serious waste of bandwidth. But fragmentation increases CMTS overhead in three ways: reassembly overhead, slightly larger packet headers, and extra bookkeeping when DOCSIS 1.0 and 1.1 cable modems are both in use.

A CMTS is both a router and a bridge; the bridged (DOCSIS) interfaces have strict timing requirements and the CMTS must create MAPs for each upstream channel. The router side must handle multiple high speed interfaces, routing protocols, SNMP management, and other processes that consume clock cycles and memory[3]. DOCSIS is a complex protocol and features that increase either the overhead in the CMTS or make the implementation more complicated must be avoided. Relieving the CMTS of maintaining packet reassembly buffers and timers is a worthwhile goal. Because of this fragmentation overhead and complexity, it is desirable to find other mechanisms to minimize wasted allocated bandwidth.

---

[3] Other tasks may include service flows for QoS and data encryption.

Efficient Partial Bandwidth Allocation

When the CMTS allocates only a fraction of the bandwidth requested by a cable modem, it does so somewhat arbitrarily, without knowing how much bandwidth will minimize waste. If the CMTS can allocate partial bandwidth by knowing the size of packets at the head of cable modem queues, then allocated bandwidth can be used optimally without the need for fragmentation. This requires cable modems to indicate the size of packets at the head of the queues. To facilitate this, I propose to modify the request format slightly so that a cable modem can send two numbers to CMTS: (1) the aggregate request size, as in the DOCSIS specification and (2) the size of the first packet in its queue. The latter is called minimum usable (*mu*) bandwidth. If the CMTS cannot allocate the requested bandwidth, it can instead allocate the amount given by *mu*, which is sufficient to fit one packet without any wasted mini-slots. This is analogous to the maximum transmission unit (MTU) concept used on the Internet. Using extensive simulations, I show in Chapter 5 that *mu* keeps mini-slot waste to a small percentage of allocated bandwidth.

4.2 Improving DOCSIS Performance Through MAP Creation

It is natural to ask whether the manner in which MAPs are created can affect the overall performance. In this section, I will explore several MAP creation strategies and their effects on performance. The performance metrics used in this work are response time and utilization. Response time is defined as the time between the arrival of a packet at a cable modem for upstream transmission and the time at which the packet returns to the cable modem on the downstream channel. Response time is measured from the point-of-view of a user of a DOCSIS network. Utilization of the upstream channel has the usual definition of throughput divided by capacity. It is measured from the point-of-view of the CMTS for each upstream channel.

In order to better present the concepts in this section, I introduce the concepts of *target MAP size* and *MAP completion*. The term *target MAP size* refers to the upper limit, which may or may not be strictly defined, on the number of mini-slots a single MAP may cover. The *target MAP size* is an implementation data entity and is not found in the DOCSIS specification. The notion of target MAP size is similar to that of target token rotation time in FDI and token ring networks. At the moment when no more future mini-slots may be allocated, the CMTS mini-slot allocation list is transferred to a physical data packet, formatted into Information Elements, and broadcast on the downstream channel. This action is called *MAP completion.* There may be different triggers of this action, such as reaching the *target MAP size* or the possibility of a gap occurring between MAPs.

4.2.1 Parameters of MAP Creation

Within the guidelines of DOCSIS, there are two general aspects of MAPs that vary: the target MAP size in mini-slots and the proportion of overhead mini-slots to grant mini-slots. MAPs can be smaller than a target MAP size, especially under low offered loads. Clearly, an absolutely fixed MAP size would require padding with either overhead mini-slots or unrequested grant mini-slots. This reduces utilization, so it is preferable to treat the target MAP size as a reference size and let individual MAP sizes vary within reasonable bounds. I examine various methods to set individual MAP sizes given a target MAP size, to determine grant sizes, and to minimize overhead slots.

4.2.2 MAP Size Control

MAP size is important because it affects response time. MAP size indirectly specifies the interarrival time of bandwidth request mini-slots and of grant mini-slots. A good analogy to illustrate the effects of bounding MAPs is that of the target token rotation time (TTRT) used in FDDI [55]. If the token does not arrive within bounded intervals, the overall response time suffers if one user retains the token much longer than other users. This situation also allows one user to gain unfair use of bandwidth. My early simulations using naïve MAP strategies provide empirical evidence of this same effect. DOCSIS response time became unusable under heavy loads when MAPs were unbounded.

The optimal or ideal target MAP size is influenced by a variety of network parameters, such as the offered load, the number of cable modems, and the traffic distribution. Exploring these relationships is a topic for future research and the target MAP sizes used here should not be construed to be optimal. Rather, given that MAPs must be bounded and that there are several reasonable choices for MAP sizes, I compare performance results using the MAP strategies explained in this section and listed in Figure 4-1.

It is intuitive to surmise that using small MAP sizes can improve response time. However, sending many small MAPs is usually considered inefficient use of the downstream channel. This is a minor drawback. A more serious drawback is that overhead slots are a cost incurred on a per MAP basis, thereby decreasing utilization. Small MAPs make allocating large grants difficult as well. This could delay granting a request by several MAPs when the MAP size is too small. This in turn increases the response time from the point of view of a cable modem since grants arrive at larger intervals. Consideration of this parameter suggests many possible ways the MAP boundary could be specified. I have selected three methods that are simple to implement and do not require elaborate state maintenance [57]. I call these methods *strictly bound (sb)*, *loosely bound (lb),* and *average bound (ab)*:

- When the *strictly bound* MAP creation method is applied, the target MAP size is never exceeded. If a bandwidth request cannot be satisfied without exceeding the target MAP size, the MAP is completed and the unsatisfied bandwidth request must wait for the next MAP to be built. Using this method, MAPs can be smaller but they never exceed the target MAP size.

- In the *loosely bound* MAP creation method, the target size is checked after each bandwidth allocation is made. If the target MAP size is exceeded, the MAP is completed. The MAP boundary is almost always exceeded, but by only a single grant. Under high offered loads, this approach may increase the MAP size by the largest request size.

- For the *average bound* method, the target MAP size will vary using a credit scheme. Any MAP that does not use the quota of mini-slots defined by the target size accumulates the difference as a future credit. This credit can be used to extend subsequent MAPs, thus the MAP size can go up and down, but the average size is always near the target value. In this simulation, the accumulated credit can not exceed 50% of the target size in order to prevent the credit pool growing very large under low offered loads. MAPs are created such that they are smaller than target MAP size plus available credit.

4.2.3 Grant Allocation Methods

Grant allocation offers a way to avoid fragmentation. I have explored three different approaches to this issue. Each approach has an allocation type, called (1) *any*, (2) *minimum usable (mu)*, and (3) *last minimum usable (lmu)*.

The *Any* approach provides the grant requested under the bounding methods previously described. The *mu* and *lmu* approaches require a small but significant protocol modification.

Recall that bandwidth requests are for the total amount of mini-slots needed to send upstream the packets queued at the cable modem. If the CMTS needs to provide a smaller grant than that requested, it has no information to help choose a grant that can be utilized completely by the cable modem. Simplistic approaches such as giving grants large enough to hold an Ethernet frame or an average sized packet are likely to be wasteful. My preliminary simulations verified this. Adding the *mu* field allows the cable modem to state the minimum amount of bandwidth it needs, along with the total amount. Since *mu* indicates the size of the first packet in the cable modem's queue, allocating bandwidth based on *mu* will ensure that the requesting cable modem can completely use the grant.

To facilitate the description of the three methods of bandwidth allocation, I define three variables: (1) $\mathbf{m}$, which specifies the target MAP size, (2) $\mathbf{g}$, which indicates the total of the bandwidth requests found in the CMTS queue at the moment a MAP is constructed, and (3) $\mathbf{g_m}$, which is the total of the minimum usable allocation requests in the CMTS queue. The three methods work as described below:

- The *any* method is really a special case because grants are allocated using only the MAP bounding rules and standard DOCSIS grant request field. No other information is considered.

- With the *mu* method the allocation amount depends on $\mathbf{g}$, $\mathbf{m}$, and $\mathbf{g_m}$. If $\mathbf{g} = \mathbf{m}$, then each queued request can be completely satisfied and the allocations are based on the standard DOCSIS bandwidth requests. However, when $\mathbf{g} > \mathbf{m}$ either the *mu* or *lmu* methods may be applied by examining $\mathbf{g_m}$. If $\mathbf{g_m} = \mathbf{m}$, the allocations are based on the *mu* requests until the bounding method completes the MAP. Otherwise, the grant allocation is similar to that in the *lmu* method.

- Under the *lmu* method, requests are granted based on the maximum requested amount until the target MAP size is reached. If the MAP is below the target MAP size, the first *mu* request remaining in the CMTS queue is checked. If it can be allocated, it becomes the last allocation in the MAP. This is why this method is called *last minimum usable*.

Of the three grant allocation methods, only *any* is possible under DOCSIS 1.0 and 1.1. Before the actual simulated strategies can be described, the two overhead control methods will be discussed. *Mu* and *lmu* do not allocate bandwidth that cannot be fully used by a cable modem, so fragmentation is avoided.

4.2.4 Upstream Overhead

Any part of the upstream bandwidth that does not carry user data is considered overhead. This includes the packet headers, preamble bits, and error correction bits. The largest component of overhead involves the mini-slots used to request bandwidth and those used to perform station maintenance. The number of maintenance slots is a small percentage (2%) of the number of cable modems serviced by the upstream channel and is dependent on physical layer characteristics and cable modem clock drift. I do not address the effects of station maintenance mini-slots in this work.

As with MAP sizes and layout, DOCSIS does not provide any firm rules for controlling the number of contention slots in a MAP. A simple approach is to allocate a fixed number of contention slots in each MAP. This approach is not efficient because a value that is too large wastes bandwidth in high load cases, in which most cable modems use piggyback requests instead of contention slots to send their requests to the CMTS. A value that is too small could lead to more collisions and delays passing bandwidth requests to the CMTS under low to moderate loads. Therefore, varying the number of contention slots based on the load conditions is preferable. Some researchers have proposed complex schemes [12].

I propose a simple rule to vary the number of contention slots allocated in a MAP. For this method, the number of bandwidth request mini-slots is a percentage of the number of inactive cable modems. A cable modem is considered inactive unless it has a bandwidth request queued in the CMTS at the time a MAP is generated. Subtracting the CMTS queue length from the number of cable modems on a channel gives the inactive count. Obviously, this count decreases as the offered load increases, so the contention mini-slots are allocated in fewer numbers. The assumption here is that piggybacking will take over for the reduced contention slots and keep the protocol robust. I defined the minimum number of bandwidth request slots per MAP to be two, as a practical lower limit.

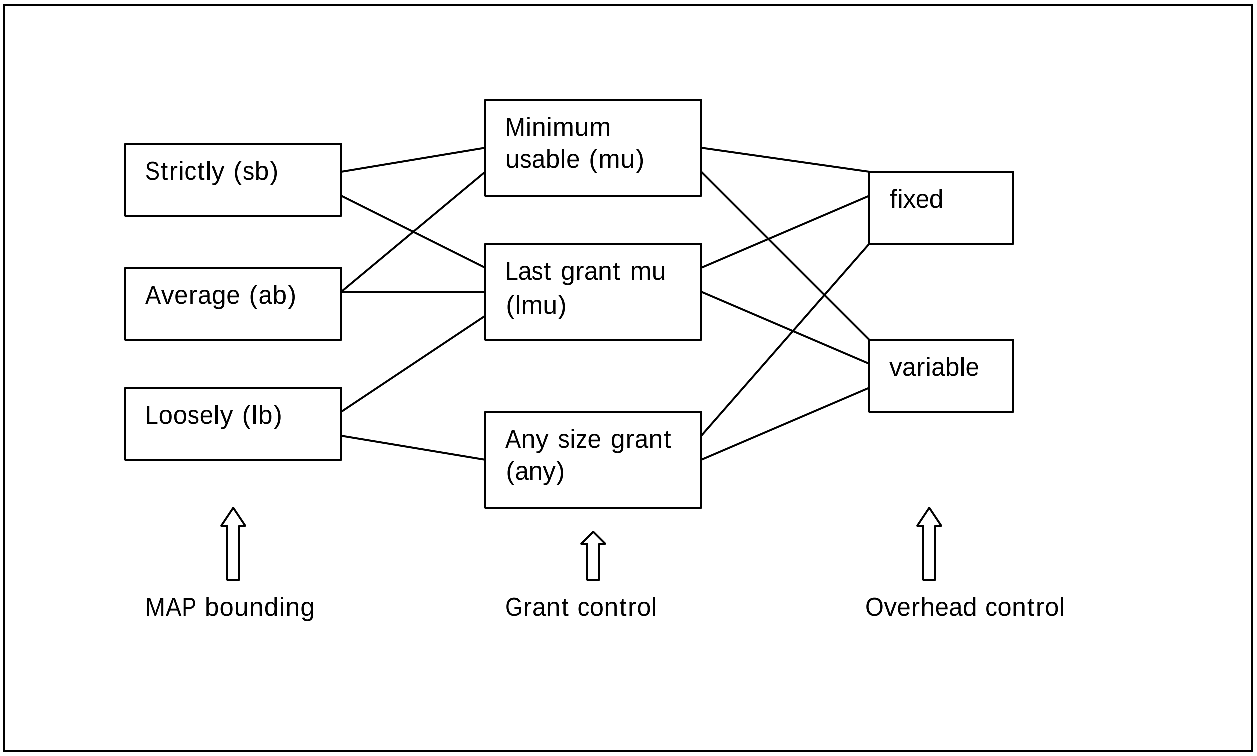


Figure 4-1. MAP Bandwidth Allocation Methods

4.2.5 Proposed MAP Creation Strategies

Figure 4-1 illustrates how each method may be joined with another to produce the twelve performance improvement strategies used in this work. A strategy is named by concatenating each method in the order of MAP bounding, grant control, and overhead control. Thus *sb-mu-var* indicates a strictly bound MAP creation with *mu* allocation request slots proportional to the number of inactive cable modems.

4.3 The Effects of Piggybacking

Piggybacking is defined as the transmission of two forms of data simultaneously. Usually the second form of data is used for signaling within the particular protocol. A well-known example is found in TCP, where an acknowledgement of received data segments may be indicated in the packet header of its data segment sent in the opposite direction. The advantage for TCP is the elimination of sending a packet purely for protocol overhead purposes. DOCSIS supports piggybacking for a reason other than saving bandwidth.

A DOCSIS cable modem may improve response time if it is able to send bandwidth requests upstream using piggybacking instead of contention slots. Piggybacking a request onto a data grant avoids delays due to collision avoidance back-off time and actual collisions. DOCSIS piggybacking is not entirely free because it requires the use of a header that is a few bytes larger than that used just for data. Piggybacking highlights the design issue of choosing the optimal number of contention mini-slots for a given MAP. Obviously, if most cable modems are piggybacking their bandwidth requests, then the need for contention mini-slots is reduced. The DOCSIS specification again leaves this matter open. So assuming that piggybacking is effective, I wish to determine the magnitude of the improvement it provides.

Piggybacking opportunities are dependent on the offered load of each cable modem and the frequency of grants. Clearly a cable modem that can only piggyback occasionally does not receive much overall benefit regardless of the performance improvement in the piggybacked transaction. Thus the question becomes one of whether a set of high load users may actually have better response times than a larger group of low load users.

To examine this topic, two scenarios are compared, both of which provide the same offered load from the perspective of the CMTS. Piggybacking is clearly effective in a protocol like TCP, where it is combined with a form of pipelining. DOCSIS does not allow pipelining of upstream bandwidth requests, so the effectiveness of piggybacking and the loads under which it becomes effective are stand alone characteristics. Varying the traffic mix is also very helpful in studying the final operational issue, the robustness and fairness of DOCSIS when a small group of heavy load users are present.

4.4 The Effects of Rogue Users

All cable modems download their operational parameters from the CMTS during the station registration process. During this registration process the CMTS can instruct cable modems to rate limit their upstream data flows. The existence of this feature implies that the DOCSIS protocol itself is not very effective in cases where users demand a disproportionately large amount of bandwidth. In any case, one or more DOCSIS users may run multi-user interactive games, peer-to-peer file sharing applications, and operate web servers.

Users who operate servers or run peer-to-peer file sharing applications are considered “rogues” because they send as much or more data than they receive. The few guidelines available for sizing upstream and downstream segments [3][4] assume most users receive much more data than they send or provide a relatively slow constant bit rate stream, like voice over IP. Clearly, deviation from this scenario needs investigation, especially if a few rogue users can affect the performance of all users. I wish to determine both the robustness and fairness of DOCSIS when a few rogue users are mixed with a much larger number of typical users.

To examine this situation, a small set of cable modems are designed to imitate rogue users and generate a much higher traffic load than the normal users. The mean response time of the two groups is compared, as well as user utilization. Robustness will be measured by the mean upstream latency, which is the time between which a packet is queued to go upstream and it is received by the CMTS. Note that this is not the same as the response time metric, which is a two-way measure. Fairness will be measured by the upstream utilization of the two groups. In the simulations for this thesis, rogue and normal users are active over the same time scales, so the effect of transient heavy upstream loads on response time is a topic for future research.

Chapter 5: Performance Analysis

In this chapter I explore the DOCSIS design and performance issues discussed in Chapter 4. There are four major design and implementation aspects that I will examine in this performance analysis.

- The benefit of fragmentation and an alternate way to achieve the same benefit without the implementation complexity and overhead.
- The effect of MAP size and grant allocation strategy on performance.
- The effectiveness of piggybacking bandwidth requests in grant regions.
- The impact of peer-to-peer networking traffic on normal users.

I use the simulator I developed and described in Chapter 3 for this performance analysis.

5.1 Baseline Parameters and Simulation Environment

The runtime environment used to derive the results shown here consists of a set of 40 SUN workstations running SunOS 5.9. Each machine computes the data for one offered load, so all results are produced in parallel. This is the general operating environment. The specific simulation parameters are described later in this section.

Each simulation run produces two output files. One file contains detailed data on the data packets seen at each cable modem and the CMTS. The second file contains data on the protocol entities such as mini-slot counts and MAP interarrival times. Post-processing with a PERL script reduces these files to a single file readable by gnuplot. To cover the desired range of offered loads (50 to 100%), 16 to 18 runs are needed. Unlike much of the prior work involving simulations, I do not vary the offered load by changing the number of cable modems on the channel. Keeping the number of cable modems constant is a better model of real world behavior. I contend the interesting aspects of DOCSIS networks are evident when many users are active and that the on-off behavior occurs in blocks. Thus, a great many users become active during the so called “prime time” hours and remain active over time scales that are long relative to the length of the simulations.

Clearly there are many degrees of freedom (see Appendix B) in the parameter space describing various network scenarios. Table 5-1 gives parameter values I used in most of my simulations. I took most of these parameters from published product literature and research papers. For example, specifying 200 users on an upstream channel follows from [4], where 200 users are approximately the maximum Cisco recommends for their widely used CMTS. In this case, this number allows the upstream channel to be saturated without specifying extreme packet arrival rates at individual cable modems. Another example is the selection of 3 or 6 for the number of rogue users on a channel. These numbers are reasonable assuming that a typical peer-to-peer application is Gnutella. While 10% or more of the users on a channel may be using Gnutella, only a very small set of them will be answering most of the queries [41]. The target MAP sizes of 100, 200, or 400 mini-slots come from the Cisco guidelines for upstream provisioning [10].

Table 5-1. Simulation Control Parameters Used in Experiments

Parameters	**Value**
Cable modem delay distribution	Random
Number of upstream channels	1
CPE packet size distribution	Poisson
Downstream packet mix	Triggered by upstream packets
Upstream bit rate	256kbps
Mini-slot size in 6.25 µs units (bytes)	8 (16)
CPE packet interarrival distribution	Exponential
Channel length	362 µseconds
Backoff start	4
Backoff end	8
Request slot ratio	Fixed:10% of the number of cable modems Variable: 10% of the number of inactive cable modems
Maintenance slot ratio	2% of target MAP size
Upstream modulation	QPSK
Downstream modulation	64QAM
Averaging interval	10 seconds
Training interval	5 seconds
Maximum batches	10
Variable parameters (one different per run)	**Value**
Number of rogue users (peer-to-peer)	0,3, or 6
CPE packet interarrival mean	Varies based on offered load
MAP advance (in mini-slots)	100, 200, or 400
CPE packet size mean	500 bytes (512 bytes for rogue users)
Number of cable modems on this channel	200, 25

I use the batch means method to analyze the statistical properties of the simulation output. The simulation run is divided into multiple durations called batches. The first batch, collected over the training interval, is used to warm up the network and is not used for statistics gathering. In each of the other batches, statistics are collected and averaged. After three or more batches are run, the means of the batches are averaged and the 95% confidence interval for average response time is calculated. If the half-width is less than 5% of the computed mean the simulation stops. Otherwise, one more batch is run and the confidence interval is again computed and checked. This process is repeated until the specified maximum number of batches is run. In the cases covered in section 5.2, this always results in the simulation terminating after four or five batches, rather than running to the limit of ten batches.

5.1.1 Performance Metrics

I used a variety of metrics to evaluate the performance of the example cable modem network. The most important of these are throughput (or utilization) and response time.

Table 5-2. Performance Metrics Used

Metric	Notes
Mean response time	Time units are seconds
Y_bar	Mean response time of batches 1 to current
Confidence interval	
Upstream utilization	% of capacity
Offered load	% of capacity, may > 100%
Number of cable modems	
Mean MAP interarrival time	
Total number of request regions	
Overused request region ratio	Overused/total request regions
Unused request region ratio	Unused/total request regions
Piggyback request ratio	Piggyback requests/total requests
Total number of bandwidth requests in contention slots	
Total number of piggybacked requests	
Total number of bandwidth requests	
Mean number of request mini-slots skipped while a grant is pending	
Mean number of request mini-slots skipped while deferring	
Mean number of request mini-slots skipped when a contention slot is used	
Mean wait time for a grant	Averaged across all cable modems
Minimum MAP size	
Maximum MAP size	
Mean MAP size	
Mean percentage of grant region waste	
Upstream utilization of regular users	
Upstream utilization of rogue users	May be zero if no rogues
Mean upstream latency of regular users	Not response time.
Mean upstream latency of rogue users	May be zero if no rogues

The complete set of metrics is indicated in Table 5-2. All metrics are computed per batch and some are computed for the entire run.

5.2 Efficient Upstream Channel Utilization without Fragmentation

Fragmentation is the present solution used by DOCSIS 1.1 cable modems to utilize the allocated bandwidth without waste. However, fragmentation considerably increases the complexity of the CMTS, which must reassemble fragments into original packets. In this section, I investigate the degree of bandwidth loss without fragmentation and alternative methods to mitigate the loss. The MAP creation strategies that use either the *mu* or *lmu* grant allocation methods (described in Chapter 4) are potential alternatives to fragmentation because the protocol has been modified to provide an additional byte of information, namely the minimum usable grant. Close examination of DOCSIS shows that conditions favoring fragmentation arise when the channel nears saturation and when the target MAP size is small. Therefore the results shown in this section include offered loads from 50% to 100% of the upstream capacity for the smallest target MAP size of 100 mini-slots. I initially use both a fixed and a variable number of request mini-slots in these comparisons.

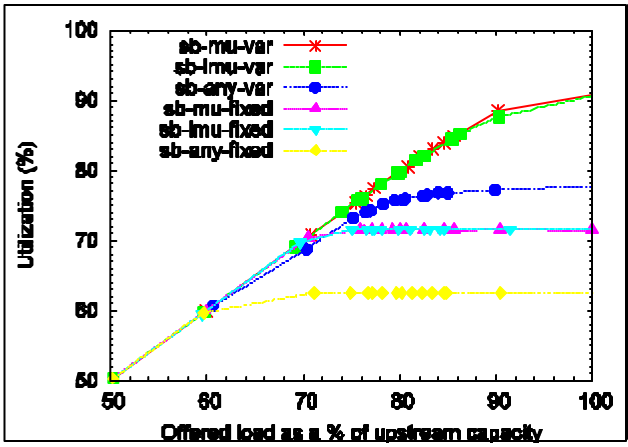


Figure 5-1. Effects of *mu*/*lmu* on utilization for MAP target size 100

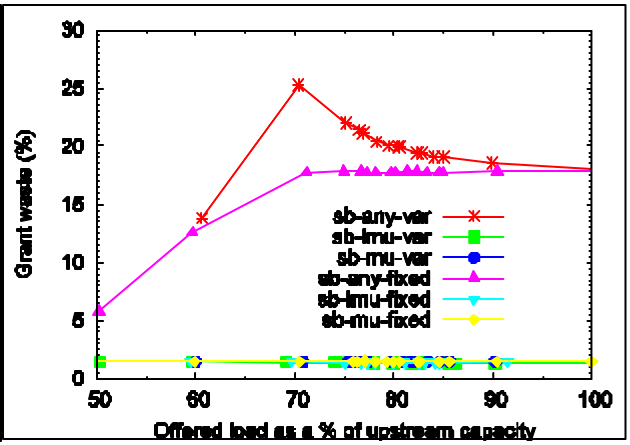


Figure 5-2. Percentage of allocated bandwidth wasted by strategy

It is necessary to compare strategies incorporating *mu* or *lmu* with others that do not. It is sufficient to show that *mu* and *lmu* provide utilization that approaches the theoretical maximum and that any waste is due to mini-slot rounding. Of course, *mu* and *lmu* must not be detrimental to the response time or this disadvantage will counteract the advantage. I also present results comparing the percentage of waste over all grants, which includes both mini-slot roundup and any unused bandwidth due to partial grants.

Looking first at Figure 5-1, one observes that the strategies using *mu* and *lmu* for grant allocation achieve the best utilization. This value peaks at ~92%, very close to the theoretical peak of 96% for the parameters used. The *sb-any-var* strategy represents the scenario with no fragmentation and no alternative to fragmentation. The utilization is about 10% less at 90% offered load. This shows the benefit of using *mu* information on the performance. It is more interesting to examine the actual percentage of allocated bandwidth that is wasted by the cable modems. For an average packet size of 500 bytes and a mini-slot that holds 16 bytes, 32 full mini-slots are needed on average. If the mean amount of waste due to roundup is .5 mini-slots, then the mean amount of waste for these parameters is .5/32 ˜ 1.5%. An inefficient mechanism will show a larger wastage than 1.5%. I plot the wasted bandwidth as a percentage of allocated bandwidth in Figure 5-2. From this figure it is clear that without fragmentation the waste can be as high as 25% and the *mu* and *lmu* alternatives to fragmentation completely mitigate this effect. Clearly, strategies incorporating either *mu* or *lmu* can achieve the same performance improvements as fragmentation, but without any reassembly overhead or tracking of DOCSIS 1.0 cable modems.

Given that fragmentation requires increasing the upstream channel overhead a small amount due to the special header it uses, the additional 8 bit field needed to implement *mu/lmu* is not a significant additional overhead, especially compared to the overhead incurred due to the fragmentation header. Thus, strategies using *mu/lmu* cost no more than fragmentation in terms of upstream overhead, eliminate reassembly overhead, and provide the same performance.

5.3 Improving DOCSIS Performance Through MAP Creation Strategies

I am interested in determining a suitable method to create MAPs given a target MAP size. (The target MAP size may be dependent on a variety of parameters such as performance guarantees, so the issue of appropriate target MAP size is not considered here. However, my results are applicable to several realistic target MAP sizes.) Twelve different MAP creation

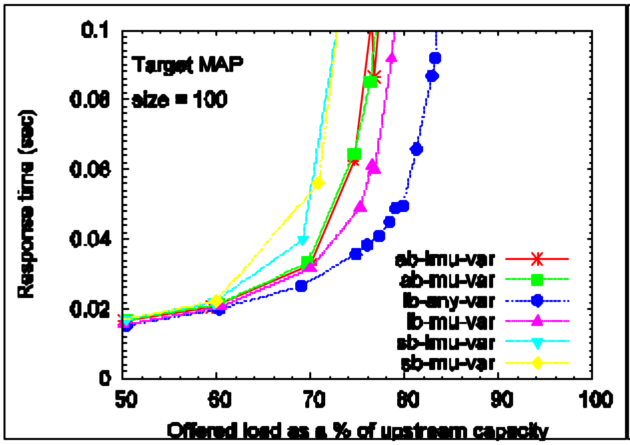


Figure 5-3. Response time for target MAP size 100

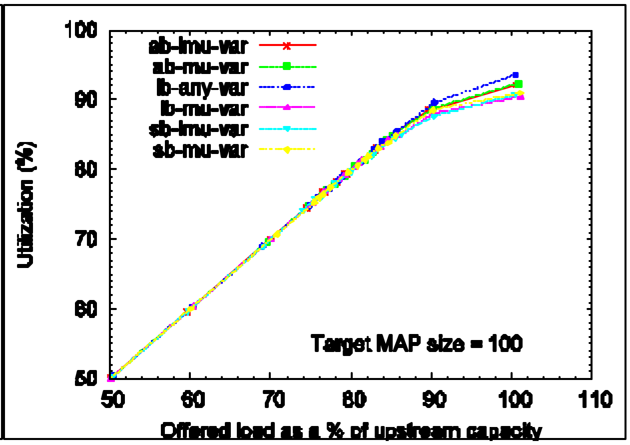


Figure 5-4. Utilization for target MAP size 100

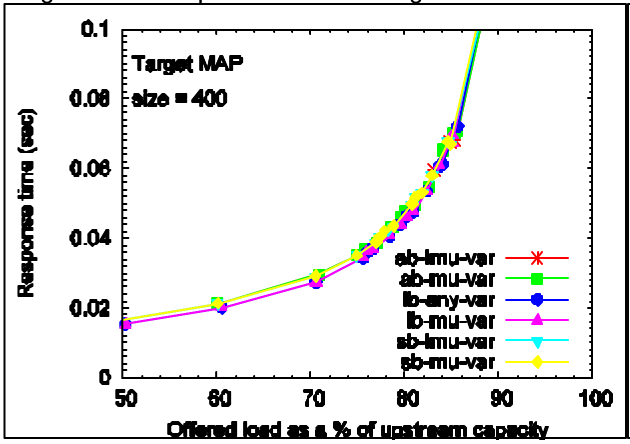


Figure 5-5. Response time for target MAP size 400

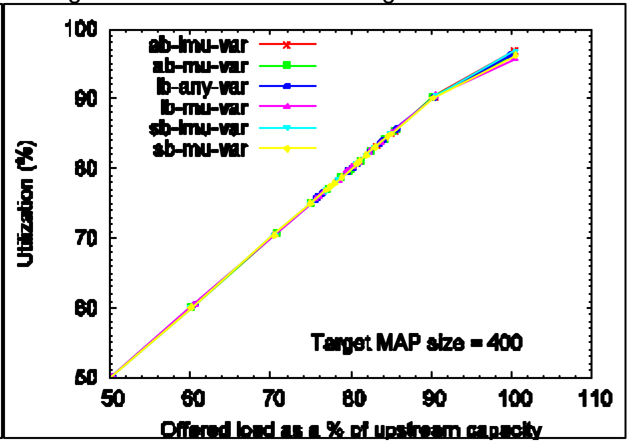


Figure 5-6. Utilization for target MAP size 400

strategies are tested and compared here. My goal is to identify a single strategy that produces the best response time and upstream utilization.

The most obvious discriminator is that of overhead control, where the *fixed* method under performs the *variable* method across all strategies (see Figure 5-1). This under performance is clearly seen in the lower utilization as the upstream approaches saturation and in the more rapid increase in response time. Subsequent comparisons will only be between strategies using *variable* overhead control.

Figures 5-3 through 5-6 illustrate an interesting effect of the target MAP size. As the target MAP size is increased, the performance differences between the different strategies become less significant. In all cases, strategies using the *loosely bound* allocation method give the best performance, as utilization is maximized while the response time growth with respect to the offered load is minimized. Figure 5-7 compares only the *lb-*-var* strategies across all target map sizes.

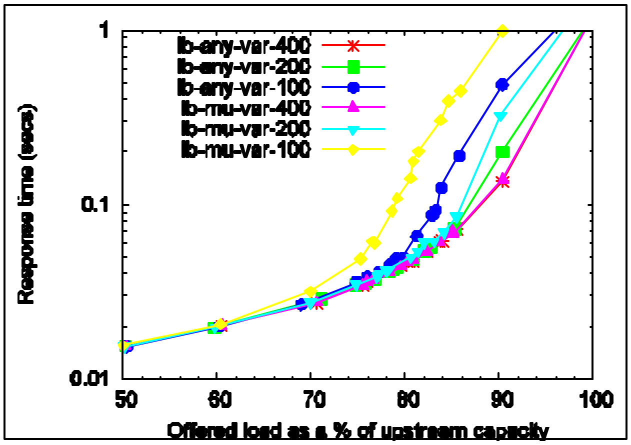


Figure 5-7. Response time for *lb-*-var* over all MAP sizes

The reason *lb-*-var* performs well is twofold. First, the loose boundary allows slightly more grants per map, thus decreasing the wait time in the CMTS queue. Second, since overhead is incurred on a per MAP basis, the *variable* method of contention slot allocation is more effective when the MAP already has more grants. Figure 5-7 shows that larger MAP sizes are better for response time as upstream saturation is approached. This result is due to the fact that all strategies used here assume that some contention slots must be present in every MAP. Thus, *for a given cable modem*, grants will be spaced slightly farther apart when small MAPs are used and this increases the mean response time. Figure 5-7 illustrates this effect.

These results suggest that the use of a strategy like *lb-mu-var* is important to maintaining the best overall performance on upstream channels as the offered load exceeds ~75%. They also imply that differences caused by target MAP sizes can be avoided by using a slightly more complex form of *lb-mu-var*. In this variant, not tested here, the number of overhead slots would be allocated for a fixed period of time rather than per MAP. In the case of a target size of 100 mini-slots, only every fourth MAP would contain contention slots, which would decrease the grant interarrival time. This strategy will be a topic for future research.

5.4 The Utility of Piggybacking

Piggybacking bandwidth requests onto an existing data grant is a feature of DOCSIS 1.1 and newer versions. While the mechanism is simple, nothing in the DOCSIS specification suggests the conditions under which piggybacking might actually provide a performance improvement. For piggybacking to be effective, a cable modem must have packets waiting to go upstream when a grant arrives. Clearly there are traffic patterns, such as a very bursty source, that can provide a high offered load but be ineffective at piggybacking.

To evaluate the effectiveness of piggybacking, I simulated two networks, each with the same offered load but with a large difference in the number of cable modems on the upstream. For a given load, the network with fewer cable modems uses the piggybacking feature more effectively. I used the *lb-*-var* strategies, but the results are applicable to other MAP creation strategies. These results appear in Figure 5-8.

The experimental results shown in this section illustrate two points: (1) piggybacking, when it is effective, does offer a worthwhile performance gain. This improvement is noticeable above 75% offered load where piggybacking decreases the response time by 50%. Of course, at very high loads, the improvement is irrelevant because a user is likely to consider both response times to be equally bad, and (2) the DOCSIS worst case scenario is created when a large number of users each generate a small portion of the load.

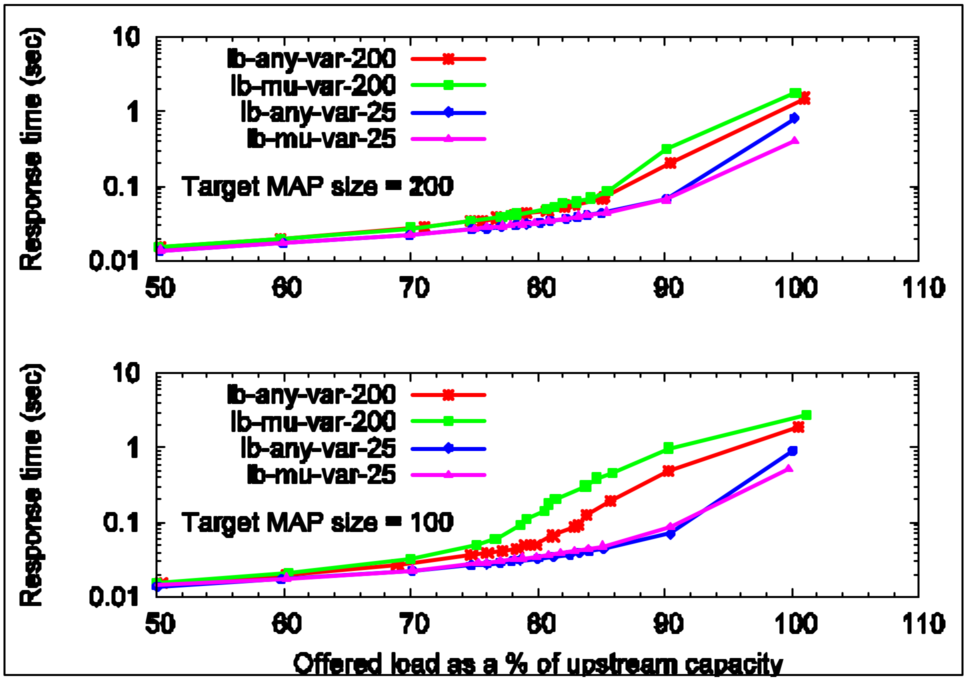


Figure 5-8. Response time improvement due to piggybacking

5.5 The Effects of Rogue Users on Performance and Fairness

While a protocol may perform well under typical or average conditions, the true test of its performance is its robustness and fairness under extreme conditions. Robustness is essentially defined as how well performance is sustained under adverse conditions near the capacity limits of the channel. Fairness refers to how much variance in performance can be caused by a small group of users who make extreme demands on the protocol. A fair protocol should ensure that these rogue users should neither obtain better performance nor degrade the performance of typical users.

The well-known TCP congestion avoidance algorithm ensures fair sharing of bandwidth across a link if all connections have roughly equal delay. For UDP traffic, however, there is no connection control, so the source with the greatest load dominates the bandwidth usage. In such cases, fairness should be ensured at the application or MAC level. For a cable modem network, this implies DOCSIS should enforce fair bandwidth sharing. DOCSIS does not appear to have such a control mechanism built in. In fact DOCSIS specifies that cable modems may receive from the CMTS an upstream bandwidth limit, along with other operational parameters. So I investigated how well the request/grant feedback limits support robustness and fairness. MAP creation is the key to both robustness and fairness and the strategies I propose in Chapter 4 must pass this test in order to be considered serious candidates for implementation in a CMTS.

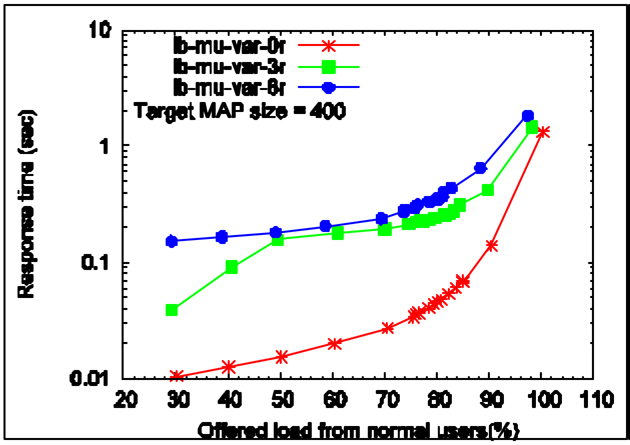


Figure 5-9. Response time of normal users

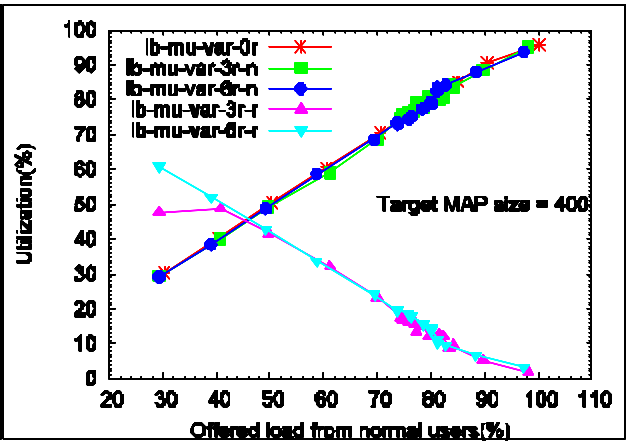


Figure 5-10. Utilization as an illustration of fairness

These experiments explore these two issues by mixing traffic from normal users and rogue users. For these simulations each rogue user generates a load equal to 16% of the upstream capacity (256 kb/s). With one scenario having three rogues and the other scenario six, the normal traffic competes with a standing load of 48% in the first case and 96% in the latter. Rogue users are designated from the given population of cable modems (200), so the actual number of normal users is either 197 or 194. The combined offered load ranges from 78% to 196% across these experiments. This causes the cable modem queues to overflow and packets to be discarded. Figure 5-9 shows the response time as a function of applied load from the normal cable modems. It is clear that rogue users increase normal users' response time significantly, often by a factor of 8 to 10. The increase in response time is similar to that seen for saturating network loads. However, the rogue users do not affect the throughput achieved, as illustrated in Figure 5-10.

5.6 Summary of Results

The results described in this section have provided insight into the following design and implementation issues found in the DOCSIS protocol:

- The utility of fragmentation and the demonstration of a simple protocol modification to sustain this utility while avoiding the fragmentation overhead.
- The performance variance caused by different MAP creation strategies and the identification of a good, simple strategy for implementation.
- The effectiveness of piggybacking and the conditions under which it becomes useful toward boosting response time.
- The fairness and robustness of DOCSIS networks under the influence of rogue users who make unreasonable demands for upstream bandwidth.

Chapter 6: Conclusions and Future Work

Cable modems have become one of the two dominant technologies for delivering broadband Internet services to the home. Recent surveys indicate that broadband users are now the majority [11]. This means that the performance of the DOCSIS protocol is a key issue in the delivery of Internet access. As I have noted earlier in this thesis, DOCSIS is unusual among the major network protocols in that an important implementation aspect, that of MAP creation, is left undefined. Vendors are free to enhance (or degrade) the performance of their DOCSIS products due to this open-ended aspect of the protocol.

I have explored the ramifications of this issue using a DOCSIS simulator that I developed specifically for this purpose. Several different MAP creation strategies have been tested using the simulator. The simulator was designed to facilitate studies of channel utilization and response time under different network loads. This allowed me to investigate a potential protocol enhancement intended to avoid fragmentation and the associated overhead. In the areas of performance, I have studied the degree of response time improvement achievable with the DOCSIS piggyback feature and the effect of rogue users on network robustness and fairness. The results show that the open-ended nature of DOCSIS does not unconditionally guarantee performance and that the protocol can be enhanced in a simple but useful way. Additionally, the results suggest that the implementers of DOCSIS networks need to pay close attention to the number of cable modems on each upstream channel.

As reviewed in Chapter 2, some research into mini-slot allocation methods has been done using simulations. In [12] the idea is to allocate contention mini-slots based on the recent history of bandwidth requests. I believe the simple mechanism here produces an equally effective allocations strategy without the necessity of statistical calculations in the CMTS.

Future work:

Based on my research, I believe that there are strong reasons for CMTS implementations to be programmable. Simply put, it is conceivable that MAP creation strategies and other parameters should be service-provider implemented, beyond the default behavior the vendor may provide. I also believe that the type-length-value (TLV) nature of passing parameters in protocol administration headers allows the flexibility to define a header that could be used, for example, to implement *mu*. One may argue that this approach requires that cable modems also be programmable, which they are in a sense now due to the amount of configuration data they can download from the CMTS.

I have considered a number of tasks that will extend this work and support future research in the area of DOCSIS network performance. The first of these concerns the implementation of the simulator. The simulator needs more complete error checking and run-time efficiency improvements. The latter can be provided by scheduling grant region arrivals only at the cable modem receiving the grant instead of using the present relay scheme. It still remains necessary for each contention region to be seen at each cable modem in order for the proper number of slots to be deferred.

More elaborate simulation scenarios require that each cable modem have an independent traffic pattern. Currently, only two concurrent traffic distributions are allowed. Also, additional statistical distributions should be included.

One final but major enhancement is the addition of layer 3 protocol support. While TCP would be the most common and useful protocol to study, other protocol modules could be added. This is important because TCP performance will clearly be less robust than the layer 2 study done in this thesis due to cable modem queue overflow at high loads. Gauging the TCP performance is a very practical issue since the most common user application, Web browsing, is TCP based.

The results described in this thesis must be evaluated in terms of the existing body of research on broadband CATV data networks and the practical expectations of users. An example of the latter concerns the issue of response time. While the simulations show bounded response times even with offered loads exceeding 75%, there is also a large amount of queue overflow occurring in the cable modems under these conditions. Thus, this packet loss will have an adverse effect on the TCP flows that are generated by each user. While the individual packet response time may be reasonable, the response time at the TCP layer will be much worst. Of course, the TCP layer will apply congestion avoidance to limit packet losses, so the magnitude of the performance loss is not intuitive. This point will be addressed as a future research topic.

Appendix A – DOCSIS Upstream Characteristics

Table A-1.QPSK Modulation Data Rate Parameters

QPSK Modulation

Bits per symbol	Symbol Rate	Bits per second	Mini-slot size	bits per mini-slot	bytes / mini-slot	No of mini-slot for frame	Max grant size in mini-slots	Bits per grant	Time per grant (secs)	Max future time	Minslots on the wire
2	160000	320000	2	4	0.5	3000	255	1020	0.003188	0.0512	57.92
2	160000	320000	4	8	1	1500	255	2040	0.006375	0.1024	28.96
2	160000	**320000**	**8**	**16**	**2**	**750**	**255**	**4080**	**0.01275**	**0.2048**	14.48
2	160000	320000	16	32	4	375	255	8160	0.0255	0.4096	7.24
2	160000	320000	32	64	8	188	255	16320	0.051	0.8192	3.62
2	160000	320000	64	128	16	94	255	32640	0.102	1.6384	1.81
2	160000	320000	128	256	32	47	255	65280	0.204	3.2768	0.905
2	320000	640000	2	8	1	1500	255	2040	0.003188	0.0512	57.92
2	320000	640000	4	16	2	750	255	4080	0.006375	0.1024	28.96
2	320000	**640000**	**8**	**32**	**4**	**375**	**255**	**8160**	**0.01275**	**0.2048**	14.48
2	320000	640000	16	64	8	188	255	16320	0.0255	0.4096	7.24
2	320000	640000	32	128	16	94	255	32640	0.051	0.8192	3.62
2	320000	640000	64	256	32	47	255	65280	0.102	1.6384	1.81
2	320000	640000	128	512	64	24	255	130560	0.204	3.2768	0.905
2	640000	1280000	2	16	2	750	255	4080	0.003188	0.0512	57.92
2	640000	1280000	4	32	4	375	255	8160	0.006375	0.1024	28.96
2	640000	**1280000**	**8**	**64**	**8**	**188**	**255**	**16320**	**0.01275**	**0.2048**	14.48
2	640000	1280000	16	128	16	94	255	32640	0.0255	0.4096	7.24
2	640000	1280000	32	256	32	47	255	65280	0.051	0.8192	3.62
2	640000	1280000	64	512	64	24	255	130560	0.102	1.6384	1.81
2	640000	1280000	128	1024	128	12	255	261120	0.204	3.2768	0.905
2	1280000	2560000	2	32	4	375	255	8160	0.003188	0.0512	57.92
2	1280000	2560000	4	64	8	188	255	16320	0.006375	0.1024	28.96
2	1280000	**2560000**	**8**	**128**	**16**	**94**	**255**	**32640**	**0.01275**	**0.2048**	14.48
2	1280000	2560000	16	256	32	47	255	65280	0.0255	0.4096	7.24
2	1280000	2560000	32	512	64	24	255	130560	0.051	0.8192	3.62
2	1280000	2560000	64	1024	128	12	255	261120	0.102	1.6384	1.81
2	1280000	2560000	128	2048	256	6	255	522240	0.204	3.2768	0.905
2	2560000	5120000	2	64	8	188	255	16320	0.003188	0.0512	57.92
2	2560000	5120000	4	128	16	94	255	32640	0.006375	0.1024	28.96
2	2560000	**5120000**	**8**	**256**	**32**	**47**	**255**	**65280**	**0.01275**	**0.2048**	14.48
2	2560000	5120000	16	512	64	24	255	130560	0.0255	0.4096	7.24
2	2560000	5120000	32	1024	128	12	255	261120	0.051	0.8192	3.62
2	2560000	5120000	64	2048	256	6	255	522240	0.102	1.6384	1.81
2	2560000	5120000	128	4096	512	3	255	1044480	0.204	3.2768	0.905

Table A-2. 16QAM Modulation Data Rate Parameters

16QAM Modulation

Bits per symbol	Symbol Rate	Bits per second	Mini-slot size	bits per mini-slot	bytes / mini-slot	No of mini-slot for frame	Max grant size in mini-slots	Bits per grant	Time per grant (secs)	Max future time	Minslots on the wire
4	160000	640000	2	8	1	1500	255	2040	0.003188	0.0512	57.92
4	160000	640000	4	16	2	750	255	4080	0.006375	0.1024	28.96
4	160000	**640000**	**8**	**32**	**4**	**375**	**255**	**8160**	**0.01275**	**0.2048**	14.48
4	160000	640000	16	64	8	188	255	16320	0.0255	0.4096	7.24
4	160000	640000	32	128	16	94	255	32640	0.051	0.8192	3.62
4	160000	640000	64	256	32	47	255	65280	0.102	1.6384	1.81
4	160000	640000	128	512	64	24	255	130560	0.204	3.2768	0.905
4	320000	1280000	2	16	2	750	255	4080	0.003188	0.0512	57.92
4	320000	1280000	4	32	4	375	255	8160	0.006375	0.1024	28.96
4	320000	**1280000**	**8**	**64**	**8**	**188**	**255**	**16320**	**0.01275**	**0.2048**	14.48
4	320000	1280000	16	128	16	94	255	32640	0.0255	0.4096	7.24
4	320000	1280000	32	256	32	47	255	65280	0.051	0.8192	3.62
4	320000	1280000	64	512	64	24	255	130560	0.102	1.6384	1.81
4	320000	1280000	128	1024	128	12	255	261120	0.204	3.2768	0.905
4	640000	2560000	2	32	4	375	255	8160	0.003188	0.0512	57.92
4	640000	2560000	4	64	8	188	255	16320	0.006375	0.1024	28.96
4	640000	**2560000**	**8**	**128**	**16**	**94**	**255**	**32640**	**0.01275**	**0.2048**	14.48
4	640000	2560000	16	256	32	47	255	65280	0.0255	0.4096	7.24
4	640000	2560000	32	512	64	24	255	130560	0.051	0.8192	3.62
4	640000	2560000	64	1024	128	12	255	261120	0.102	1.6384	1.81
4	640000	2560000	128	2048	256	6	255	522240	0.204	3.2768	0.905
4	1280000	5120000	2	64	8	188	255	16320	0.003188	0.0512	57.92
4	1280000	5120000	4	128	16	94	255	32640	0.006375	0.1024	28.96
4	1280000	**5120000**	**8**	**256**	**32**	**47**	**255**	**65280**	**0.01275**	**0.2048**	14.48
4	1280000	5120000	16	512	64	24	255	130560	0.0255	0.4096	7.24
4	1280000	5120000	32	1024	128	12	255	261120	0.051	0.8192	3.62
4	1280000	5120000	64	2048	256	6	255	522240	0.102	1.6384	1.81
4	1280000	5120000	128	4096	512	3	255	1044480	0.204	3.2768	0.905
4	2560000	10240000	2	128	16	94	255	32640	0.003188	0.0512	57.92
4	2560000	10240000	4	256	32	47	255	65280	0.006375	0.1024	28.96
4	2560000	**10240000**	**8**	**512**	**64**	**24**	**255**	**130560**	**0.01275**	**0.2048**	14.48
4	2560000	10240000	16	1024	128	12	255	261120	0.0255	0.4096	7.24
4	2560000	10240000	32	2048	256	6	255	522240	0.051	0.8192	3.62
4	2560000	10240000	64	4096	512	3	255	1044480	0.102	1.6384	1.81
4	2560000	10240000	128	8192	1024	2	255	2088960	0.204	3.2768	0.905

Appendix B – Glossary of Simulation Control Parameters

Here the term "floating point" means that a number can be given as a decimal fraction or in exponential format. If there is no fractional part, the decimal point may be omitted and the value will be parsed correctly. White space is ignored, but all keywords are case sensitive. If shown, the equal sign ("=") is required for correct parsing. Certain parameters are set for each upstream channel. A channel is specified by number and this defines the "channel context" for all subsequent channel parameters until a different channel is specified.

averaging-interval ? The duration of sample collection for each batch specified in seconds. There is no default. This value is a strictly positive floating point quantity. Example: averaging-interval = 10.0.

backoff-end ? The exponent of the power of two that gives the largest number of transmit opportunities that may be deferred when attempting collision avoidance in contention slots. This is an integer greater than or equal to **backoff-start** and less than or equal to eight. There is no default. This parameter applies to the current channel context.

backoff-start ? The exponent of the power of two that gives the initial upper bound of the number of transmit opportunities to defer during collision avoidance. This is a integer greater than or equal to zero and less than or equal to **backoff-end**. There is no default. This parameter applies to the current channel context.

cable-modem ? Sets the subsequent parameter context to that of a particular cable modem within the current channel context. This parameter is currently ignored.

cable-modem-buffer-size ? Defines the size of the upstream packet buffer in bytes. This value must be a positive integer and defaults to 16384. Example: cable-modem-buffer-size = 1024

cable-modem-mute ? A positive integer used to derive the set of cable modems that do not generate any upstream load during the muting interval. For example, if it set to two, all even numbered SIDs will exhibit on/off behavior. There is no default and this value is ignored unless **mute-time** is a positive value.

channel-length ? The maximum length of the current channel given as a floating point number of seconds. There is no default. This value must be greater than zero and less than 724e-6 seconds.

cmts-source-interarrival ? This parameter controls packets arriving at the CMTS to be sent downstream. It specifies the packet arrival distribution function and the parameters needed by that function. The support distribution types are CONSTANT, UNIFORM, and EXPO. The CONSTANT and EXPO distributions have a single parameter, which is the interarrival time or mean interarrival time respectively. The UNIFORM distribution has two parameters, which are the minimum and maximum interarrival times. The times are all given in floating point seconds and there is no default. Example: cmts-source-interarrival = UNIFORM 1e-1 2.0

cmts-source-packetsize ? This parameter controls packets arriving at the CMTS to be sent downstream. It gives the distribution function that controls the packet size (in bytes) and the parameters required by the given function. The function choices are CONSTANT_INT, UNIFORM_INT, and POISSON. A single integer gives either the constant packet size or the mean packet size in the case of the poisson distribution. A pair of integers gives the minimum and maximum sizes for the uniform distribution. All integers must be strictly positive and there is no default. Example: cmts-source-packetsize = POISSON 340

confidence-interval ? This floating point value gives the confidence interval width used to stop the simulation when the mean response time across the last batch falls within this interval around the mean of all batches. It defaults to 0.95.

confidence-interval-stop ? If this keyword is present, the simulation will stop using the confidence interval criteria. If it is absent the simulation will run for the specified number of batches.

cpe-source-interarrival ? See the definition for **cmts-source-interarrival**. This parameter controls the packets arriving at a cable modem to be sent upstream.

cpe-source-packetsize ? See the definition for **cmts-source-packetsize**. This parameter controls the packets arriving at a cable modem to be sent upstream.

downstream-modulation ? This parameter must be the ASCII string “64QAM” or “256QAM”. There is no default. It controls the bandwidth of the downstream channel.

ethernet-frames ? If this keyword appears, all packets arriving at a cable modem to go upstream are shaped to look like Ethernet frames. In other words, the “packet” is considered only the data part and will be sent as one or more Ethernet frames. This means that no upstream transmission is less than 64 bytes or greater than 1518 bytes. The default is to not create Ethernet frames.

grant-control ? The parameter is used to control how grants are created in the MAP builder process. It can take one of three ASCII values: lmu, any, and mu. There is no default.

immediate-data-slot-ratio ? This value specifies the ratio of immediate data slots to the MAP target value. It is a floating point value between 0.0 and 1.0. This parameter is currently ignored.

maintenance-slot-ratio ? This value specifies the ration of station maintenance slots to the MAP target value. It is a floating point value between 0.0 and 1.0, with a typical value of 0.02. There is no default. This parameter applies to the current channel context.

make-output-files ? The use of this keyword will override the simulator sending its output to STDIO and STDERR. Instead, statistics will be written to a file with this naming scheme: *ch<number of channels>nai<no of batches>_<DD>_<MM>_<HH>:<MM>:<SS>.out*. The day, month, hour, minute, and second are taken to be the time the simulation is run. The simulation bookkeeping trace will be written to a file with the same name except the extension is *trc* instead of *out*. This parameter may not be used when either **output-file** or **trace-file** is used. The purpose of this automatic name creation scheme is to allow multiple simulation runs to write output files into the same directory with filename conflicts.

map-advance ? This integer value gives the target MAP size in mini-slots. There is no default. This parameter applies to the current channel context.

map-bounds ? This parameter controls the creation of MAPs by specifying the way the target MAP size is handled. The three possible values are: “strict”, “average”, and “loose.” There is no default.

map-policy-ratio ? The use of this parameter is deprecated and it should not be used.

mini-slot-size ? This integer gives the mini-slot size in DOCSIS timeticks. There is no default. This parameter applies to the current channel context.

mute-time ? This floating point value gives the time cable modems in the muted group do not receive packets from the CPE source. It must be strictly positive. The default value is zero, which indicates that there is no on/off behavior and any muted group is ignored.

no-ave-intervals ? This integer value gives the number of batches over which data should be accumulated. It must be strictly positive. There is no default.

no-of-cable-modems ? This integer value gives the number of cable modems on the current channel. It must be strictly positive and a reasonable limit is 200. There is no default.

no-of-channels ? This integer gives the number of upstream channels within the range of one to six. The channels are numbered zero through five when specifying the channel context. There is no default.

output-file ? This parameter defines an ASCII string that is the filename to be used for the simulation results. Use of this parameters overrides the default of writing these results to STDOUT. There is no default file name. This parameter must not be used in the same configuration file as the **make-output-files** keyword.

overhead-control ? This parameter defines one of two method of overhead control used during MAP creation: “fixed” or “variable.” There is no default. Example: overhead-control = fixed

packet-limit-down ? This integer value specifies the number of packets sent downstream that will terminate the simulation. It must be a positive integer and the default value is zero. A value of zero means the simulation will not terminate by downstream packet count.

packet-limit-up ? Same as **packet-limit-down** except counts packets in the upstream. This value refers to the total of all packets received from each upstream channel.

request-slot-ratio ? This value specifies the ration of grant request slots to the MAP target value. It is a floating point value between 0.0 and 1.0, with a typical value of 0.1. There is no default. Note that the number of grant request slots in a MAP will be further scaled down if **overhead-control** is specified to be “variable.” This parameter applies to the current channel context.

re-seed ? The use of this keyword causes the CSIM random number generator to be reseeded between batches. The default action is not to reseed.

rogue-cpe-src-interarrival ? Same as **cpe-source-interarrival** except it applies only to the rogue modems in the current channel context. There is no default. This parameter is ignored if there are no rogue modems specified.

rogue-cpe-src-packetsize ? Same as **cpe-source-packetsize** except it applies only to the rogue modems in the current channel context. There is no default. This parameter is ignored if there are no rogue modems specified.

rogue-user ? This integer parameter specifies the SID of a cable modem that belongs to the set of rogue users. These cable modems use the alternate packet interarrival and size distributions. Multiple rogue users may be created by using multiple instances of this parameter. There is no default. The SID must be between zero and one less than the number of cable modems on the current channel.

show-modem-details ? This keyword causes all cable modem counters to be printed at the end of each batch. The default output shows only channel and CMTS level statistics.

sim-time-limit ? The simulation terminates after this much elapsed time. This is a floating point value and the default action is to terminate the simulation after a given number of batches.

stop-after-maps ? If this parameter is given, the simulation will terminate after this many MAPs have been sent downstream. The value must be a strictly positive integer. Example: stop-after-maps = 10

time-scale-factor ? This parameter is obsolete and is ignored if used.

title ? This parameter is any string of characters to be printed as a title in the statistics output file. There is no default. White space is allowed in the title. Example: title = This is a test

trace-file ? This parameters specifies an ASCII string to be used as the trace output filename. This name overrides the use of STDERR. This parameter should not be used in the configuration file as **make-output-files**.

training-interval ? This value gives the number of seconds the simulation will run while discarding the contents of all counters. After this interval, all statistics and counters are retained and averaged. This purpose is to ignore data during the start of the simulation because this data may skew the statistics. This is a floating point value greater than or equal to zero. There is no default.

trigger-action ? This parameter gives one of three ASCII keywords to provide additional downstream packets: “ECHO”, “RANDOM” or “FIXED.” These actions do not apply unless the keyword **up-triggers-down** appears in the configuration. ECHO means to send the same packet back downstream. FIXED means to send a packet of a specified fixed size back in response. RANDOM means to send a packet with a random size drawn from a distribution with the mean given by the **trigger-parameter**. Note that the creation time of the upstream packet is copied into the down stream packet. This allows an accurate response time to be computed for each upstream packet.

trigger-parameter ? This floating point value gives the size of the packet to send downstream for the FIXED case. For the RANDOM case, it gives the mean packet size. There is no default. See **up-triggers-down** for an explanation of this feature.

upstream-modulation ? This is an ASCII string for the type of upstream modulation to apply to the current channel context. Values may be either “QPSK” or “16QAM.” There is no default.

upstream-symbol-rate ? This is a integer specifying one of the following values: 160000, 320000, 640000, 1280000, 2560000. There is no default. This parameter applies to the current channel context.

up-triggers-down ? Normally the packets going downstream have no relationship to those going upstream. The use of this keyword establishes a relationship by have the CMTS generate downstream packets on receipt of an upstream packet. These packets are directed back to the cable modem that sent the upstream packet. See **trigger-action** for a description of the three ways the CMTS can insert these packets. The default action is to not trigger downstream packets. Note that these packets are interspersed with any that are created by the CMTS source.

Appendix C - Cable Internet Abbreviations and Acronyms

802.14	IEEE’s Cable TV MAC and PHY Protocol Working Group
CATV	Cable TV
CMTS	Cable Modem Termination System
CPE	Customer premise equipment (PCs)
COAX	Coaxial cable
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
DAVIC	Digital Audio Video Council
DBS	Direct Broadcast Satellite
DOCSIS	Data Over Cable Service Interface Specification
DQDB	Distributed Queue Dual Bus
DSL	Digital Subscriber Line
DVB	Digital Video Broadcast
FEC	Forward Error Correction
FSK	Frequency Shift Key
FTTC	Fiber-to-the-Curb
HDSL	High bit-rate Digital Subscriber Line
Headend	The source of downstream video feeds in a CATV system
HFC	Hybrid Fiber/Coaxial
HSCDS	High-Speed Cable Data Service
IEEE	The Institute of Electronic and Electrical Engineering
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPCDN	IP Over Cable Data Network working group of the IETF
ISO	International Organization for Standardization
ISP	Internet Service Provider
ITU	International Telecommunications Union
KB	Kilobyte
LAN	Local Area Network
LLC	Logical Link Control
MAC	Media Access Control (layer of OSI Reference Model)
MAN	Metropolitan Area Network
MAP	Mini-slot Allocation Packet
MB	Megabyte
Mbps	Megabits per second
MCNS	Multimedia Cable Network System Partners Ltd.
MHz	Megahertz
MMDS	Multichannel Multipoint Distribution Service
NTSC	National Television Standards Committee
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open System Interconnect (Reference Model for networking protocols)
PC	Personal Computer
PHY	Physical (layer of OSI Reference Model)
POTS	Plain Old Telephone Service
QAM	Quadrature Amplitude Modulation
QOS	Quality of Service
QPSK	Quaternary Phase Shift Keying
SID	Service Identifier
S-CDMA	Synchronous Code Division Multiple Access

TCP/IP	Transmission Control Protocol/Internet Protocol
TDMA	Time Division Multiple Access
WAN	Wide Area Network

References

1. Radio Frequency Interface Specification, Data-Over-Cable Service Interface Specifications, document SP-RFI-I04-980724, Cable Television Laboratories, Inc., 1997.
2. Cisco uBR7246 Universal Broadband Router Features, release 11.3 XA1, Cisco Systems, http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113xa/ubr7246.htm#xtocid130910, June 1998.
3. Multimedia Traffic Engineering for HFC Networks, a White Paper on Data, Voice, and Video over IP. http://www.cisco.com/warp/customer/cc/so/cuso/sp/hfcn_wp.pdf, Cisco Systems, 1999.
4. What is the Maximum Number of Users per CMTS?. http://www.cisco.com/warp/customer/109/max_number_cmts.html, Cisco Systems.
5. National Cable & Telecommunications Association 2003 Year-End Industry Overview.NCTA, http://www.ncta.com/pdf_files/Overview.pdf, 2003.
6. Understanding Data Throughput in a DOCSIS World, document 19220, Cisco Systems, http://www.cisco.com/warp/public/109/data_thruput_docsis_world_19220.pdf, Jun. 2003.
7. Cable Datacom News, Kinetic Strategies, Inc., http://www.cabledatacomnews.com/cmic/cmic16a.html
8. http://www.stargus.com
9. DOCSIS 1.1 for the Cisco CMTS. http://www.cisco.com/en/US/products/hw/cable/ps2217/products_feature_guide_chapter09186a008019b57f.html#wp1194137, Cisco Systems, 2003.
10. Cable Modem Provisioning Scenarios. Document 49778, http://www.cisco.com/warp/public/109/modem_provision.pdf, Cisco Systems, July 2004.
11. U.S. Broadband Connections Reach Critical Mass, Nielsen Netratings, http://www.nielsen-netratings.com/pr/pr_040818.pdf, August 18, 2004.
12. Abi-Nassif, F., Lee, W.C., and Stavrakakis, I. Offered Load Estimation in a Multimedia Cable Network. *1999 IEEE International Conference on Communications, ICC '99*, Vol. 1, 140-146, 1999.
13. Adams, M. and Dulchinos, D. OpenCable. *IEEE Communications Magazine*, 98-105, June 2000.
14. Azzam, A. High-Speed Cable Modems. McGraw-Hill, New York, 1997.
15. Carroll, M.D. Aligning the Initial Maintenance Intervals of Cable Modem Upstream Channels. *IEEE Communications Magazine*, 140-146, Sept. 2003.
16. Chandrasekaran, G., Hawa, M, and Petr, D. Preliminary Performance Evaluation of QoS in DOCSIS 1.1. Technical Report 22736-01, Information and Telecommunication Technology Center, University of Kansas, Jan. 2003.
17. Chatterjee, S. TCP Behavior over HFC Cable Modem Networks. *Computer Communications*, Vol. 22, No. 14, 1321-1332, Sept. 1999.
18. Chu, K., Tai, Y., Lee, W., and Chung, P. A Novel Mechanism for Providing Service Differentiation over CATV Network. *Computer Communications* (25), 1214-1229, 2002.
19. Ciciora, W., Farmer, J., and Large, D. Modern Cable Television Technology. Morgan Kaufmann, San Francisco, 576-581, 1999.

20. Cohen, R. and Ramanathan, S. TCP for High Performance in Hybrid Fiber Coaxial Broad-Band Access Networks. *IEEE/ACM Transactions on Networks*, Vol. 6, No. 1, 15-29, Feb. 1998.
21. Dail, J.E., Dajer, M.A., Li, C., Magill, P.D., Siller, C.A., Sriram, K., and Whitaker, N.A. Adaptive digital access protocol: a MAC protocol for multiservice broadband access networks. *IEEE Communications Magazine*, Vol. 34, No.3, 104-112, Mar. 1996.
22. Domdom, R., Espey, B., Goodman, M., Jones, K., Lim, V., and Patek, S. Transient Analysis of DOCSIS 1.1 Cable Modem Networks. *2000 IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 3, 2263-2268, Oct. 2000.
23. Donaldson, G. and Jones, D. Cable Television Broadband Network Architectures. *IEEE Communications Magazine*, 122-126, June 2001.
24. Droubi, M., Idirene, N., and Chen, C. Dynamic Bandwidth Allocation for the HFC DOCSIS MAC Protocol. *Proc. of the Ninth International Conference on Computer Communications and Networks*, 54-60, Oct.2000.
25. Dutta-Roy, A. An Overview of Cable Modem Technology and Market Perspectives. *IEEE Communications Magazine*, 81-88, June 2001.
26. Elloumi, O., Golmie, N., Afifi, H., and Su, D. A Simulation-based study of TCP Dynamics over HFC networks. *Computer Networks* (21), 307-323, 2000.
27. Fellows, D. and Jones, D. DOCSIS Cable Modem Technology. *IEEE Communications Magazine*, 202-209, Mar. 2001.
28. Golmie, N., Mouveaux, F., and Su, D. A Comparison of MAC Protocols for Hybrid Fiber/Coax Networks: IEEE 802.14 vs. MCNS. *1999 IEEE International Conference on Communications, ICC '99,* Vol. 1, 266-272, 1999.
29. Golmie, N., Saintillan, Y., and Su, D.H. A Review of Contention Resolution Algorithms for IEEE 802.14 Networks. *IEEE Communications Surveys*, 2-12, http://www.comsoc.org/pubs/surveys, First Quarter 1999.
30. Heyaime-Duverge, C. and Prabhu, V.K. Traffic-based bandwidth allocation for DOCSIS cable networks. *Proc. of Eleventh International Conference on Computer Communications and Networks*, 586-590, Oct. 2002.
31. Huitema, C. IPv6 The New Internet Protocol. Prentice Hall PTR, New Jersey, p. 18-19, 1996.
32. Ju, H. and Liao, W. Adaptive Scheduling in DOCSIS-based CATV Networks. *Proc. of the Eleventh International Conference on Computer Communications and Networks*, 543-547, Oct. 2002.
33. Ju, H. and Liao, W. Fast Request Transmission in DOCSIS-based CATV Networks. *Proc. of IEEE International Conference on Multimedia and Expo 2002*, Vol. 1, 377-380, Aug. 2002.
34. Juan, Y. Erroneous MPEG Packet Synchronization in the MCNS/SCTE/ITU-T J.83 Annex B Standard. *IEEE Trans. on Consumer Electronics*, Vol. 44 No. 3, 963-968, 1998.
35. Kaza, T. Application Performance and System Impact over DOCSIS Networks in a Lossy System. Master's thesis, Dept. of Computer Science, University of Missouri Kansas City, 2001.
36. Kent, C.A. and Mogul, J.C. Fragmentation Considered Harmful. *Computer Communications Review,* ACM SIGCOMM, Vol. 24, No. 1, 75-87, 1995.
37. Laubach, M. Residential Area CATV Broadband Internet Technology: Current Status. Internet Protocol Journal, Cisco Systems,13-26.
38. Limb, J. and Sala, D. A Protocol for Efficient Transfer of Data over Hybrid Fiber/Coax Systems. *IEEE/ACM Trans. on Networking*, Vol. 5, No. 6, 872-881, December 1997.

39. Lin, Y., Huang, C., and Yin, W. Allocation and Scheduling Algorithms for IEEE 802.14 and MCNS in Hybrid Fiber Coaxial Networks. *IEEE Transactions on Broadcasting*, Vol. 44, No. 4, 427-435, Dec. 1998.
40. Lin, Y., Yin, W., and Huang, C. An Investigation into HFC MAC Protocols: Mechanisms, Implementation, and Research Issues. *IEEE Communications Surveys*, 2-13, http://www.comsoc.org/pubs/surveys, third quarter 2000.
41. Markatos, E.P. Tracing a large-scale peer to peer system: an hour in the life of Gnutella. *Second IEEE/ACM Symposium on Cluster Computing and the Grid*, 56-65, May 21-24, 2002.
42. Martin, J. and Shrivastav, N. Modeling the DOCSIS 1.1/2.0 MAC Protocol. *Twelfth International Conference on Computer Communications and Networks*, Dallas TX, Oct. 20-22, 2003.
43. Perkins, S. and Gatherer, A. Two-way Broadband CATV-HFC Networks: state-of-the art and future trends. *Computer Networks* (31), 313-326, 1999.
44. Rabbat, R. and Siu, K. QoS Support for Integrated Services over CATV. *IEEE Communications Magazine*, 64-68, Jan. 1999.
45. Sala, D. and Limb, J. Comparison of Contention Resolution Algorithms for a Cable Modem MAC Protocol. *Proc. of Int. Zurich Seminar on Broadband Communications*, Zurich, 83-90, Feb.1998.
46. Sala, D., Limb, J., and Khaunte, S. Adaptive Control Mechanism for Cable Modem MAC Protocols. *Proceedings of INFOCOM '98*, San Francisco CA, March 29-April 2, 1998.
47. Schnitzer, J.K. Fault and performance surveillance of DOCSIS network segments. Master's thesis, Interdisciplinary Telecommunications Dept., U. of Colorado at Boulder, 2001.
48. Schwetman, H.D. and Childress, D. User's Guide CSIM18 Simulation Engine (C++ Version), Mesquite Software, 1996.
49. Sdralia, V., Smythe, C., Tzerefos, P., and Cvetkovic, S. Performance Characterisation of the MCNS DOCSIS 1.0 CATV Protocol with Prioritised First Come First Served Scheduling. *IEEE Trans. on Broadcasting*, Vol. 45, No. 2, 196-205, June 1999.
50. Sdralia, V., Tzerefos, P., and Smythe, C. Recovery Analysis of the DOCSIS Protocol after Service Disruption. *IEEE Transactions on Broadcasting*, Vol. 47, No. 4, 377-385, Dec. 2001.
51. Sriram, K. Performance of ATM and Variable Length Packet Access in Broadband HFC and Wireless Networks. *Proc. IEEE 1998 International Conference on Universal Personal Communications*, Vol. 1, 495-501, 1998.
52. Stallings, W. Handbook of Computer Communications Standards, Vol. 1. Macmillan, New York, 1987, 69-100.
53. Tzerefos, P., Sdralia, V., Smythe, C., and Cvetkovic, S. Delivery of Low Bit Rate Isochronous Streams Over the DOCSIS 1.0 Cable Television Protocol. *IEEE Trans. on Broadcasting*, Vol. 45, No. 2, 206-214, June 1999.
54. Tzerefos, P., Smythe, C., Stergiou, I., and Cvetkovic, S. Standards for High Speed Digital Communications Over Cable TV Networks. *6th IEE Conference on Telecommunications*, pub. No. 451, 224-229, March 29-April 1, 1998.
55. Walrand, J. and Varaiya, P. High-Performance Computer Networks. Morgan Kaufmann, San Francisco, 112-117, 1996.
56. Xiao, X., Chew, Y.H., Seah, W.K.G., and Ko, C.C. Performance Analysis for Voice and Data Integration in Hybrid fiber/coax Networks. *Computer Networks* (36), 323-341, 2001.

57. Yin, W., and Lin, Y. Statistically Optimized Minislot Allocation for Initial and Collision Resolution in Hybrid Fiber Coaxial Networks. *IEEE Journal on Selected Areas in Telecommunications*, Vol. 18, no. 9, 1764-1773, Sept. 2000.
58. Yin, W., Wu, C, and Lin, Y. Two-Phase Minislot Scheduling Algorithm for HFC QoS Services Provisioning. *IEICE Trans. on Communications,* Vol. E85-B, No. 3, 582-593, Mar. 2002.

VITA

Richard Lee Murphy was born in New Jersey on October 28, 1952. His parents, Joan and Dolph Murphy, were working there temporarily and they returned to Texas shortly thereafter. Richard attended Burges High School in El Paso, where his calculus teacher introduced him to computer programming. He attended the University of Texas at El Paso from the summer of 1971 to May 1972 and majored in mathematics. He transferred to the University of Texas at Austin in the fall of 1972 and soon changed his major to the new undergraduate computer science degree. He received the Bachelor of Arts degree in computer science with a minor in mathematics in December 1974. After attended graduate school in computer science for two years, he entered the professional world as a computer programmer in Austin. He also met his future spouse, Mary Bancroft, while programming for the Center for the Study of Human Resources at UT Austin.

After leaving the employment of UT Austin, he worked for Tracor Corporation and Radian Corporation as a software designer. From Austin, Richard moved to Dallas to join E-Systems, Inc. In 1981 he, his wife Mary and young son Spenser moved to Puerto Rico, where he worked at the National Astronomy and Ionosphere Center, more commonly known as the Arecibo Observatory. At the time of his return to Texas in 1986, he was Head of the Computer Department of the Observatory.

Richard next worked for the Southwest Research Institute in San Antonio, where he is still presently employed. He has worked in Space Sciences and now is the senior technical advisor for networks and telecommunications.

He entered the graduate program in computer science at the University of Texas at San Antonio in 1999. His research interests are in network performance and modeling.

MEMBERSHIPS:

Association for Computing Machinery

PUBLICATIONS:

"Comparing the Efficiency of the Internet Protocols to DECNET," USENIX Conference Proceedings. Dallas, Texas,February 1988.

"The Southwest Data Display and Archive System: A Tool for Space Physics," Earth and Space Sciences Information System Conference Proceedings.Pasadena, California, February 1992.