Performance Characterization of the DOCSIS 1.1 HFC Network Protocol with Prioritized First Come First Served (P-FCFS) Load Scheduling Algorithm

by

Haru Alhassan

A dissertation submitted in partial satisfaction of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering
Faculty of Graduate Studies
University of Manitoba

0-612-79922-0

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION PAGE

PERFORMANCE CHARACTERIZATION OF THE DOCSIS 1.1 HFC NETWORK PROTOCOL WITH PRIORITIZED FIRST COME FIRST SERVED (P-FCFS) LOAD SCHEDULING ALGORITHM

BY

HARU ALHASSAN

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University

of Manitoba in partial fulfillment of the requirements of the degree

of

Master of Science

Abstract

The Data Over Cable Systems Interface Specification (DOCSIS) 1.1 is a Hybrid Fiber Coax (HFC) network protocol that enables the delivery of Internet Protocol (IP) traffic over the cable TV networks with significantly higher data rates in comparison to analog modems and Integrated Services Digital Network (ISDN) links.

The availability of greater bandwidth associated with the HFC network enables the delivery of high quality audio and video services. Such services require bounded delay characteristics. The aim of this project is to evaluate the capacity and upstream performance characteristic of the HFC network protocol for efficient delivery of various traffic class streams. The challenge is to optimize the DOCSIS 1.1 network performance within the framework of the limited available Quality of Service (QoS) features of the protocol.

The operation of DOCSIS 1.1 HFC network protocol was analyzed using OPNET software package. The OPNET simulation environment allows the overall network performance and that of individual node and traffic class streams to be monitored.

COMMITTEE SIGNATURE PAGE

This dissertation was presented

by

Haru Alhassan

It was defended on

August 19th, 2002

and approved by

(Signature) ______________________________

Committee Chairperson

Prof. R. D. McLeod

(Signature) ______________________________

Committee Member

Dr. Jose A. Rueda

(Signature) ______________________________

Committee Member

Dr. Muthucumaru Maheswaran

(Signature) ______________________________

Committee Member

Dr. K. Ferens (Norsat)

Dedication

This thesis is dedicated to my wife *Umma* and my two daughters, *Na'ima* and *Fadila*, and to all the people that assisted me directly or indirectly.

Acknowledgements

During my two-year study at the Department of Electrical and Computer Engineering I enjoyed the pleasure to work with my advisor, Prof R. D. McLeod. This thesis would not have been possible without his support and guidance.

I express my gratefulness to Dr J. A. Rueda, whose amazing patience, infinite help in all aspects, and non-stop support made me to achieve this objective. I am also thankful to the rest of the committee members, Dr. Muthucumaru Maheswaran and Dr. Ferens, for their suggestions in completing this work.

Finally, I would like to acknowledge Telecommunication Research Laboratory (TR*Labs*) for providing me with financial support and research facilities.

Contents

List of Tables

List of Figures

ABBREVIATIONS

ABR	Available Bit Rate
ATM	Asynchronous Transfer Mode
CPE	Customer Premises Equipment
CIR	Committed Information Rate
CBR	Constant Bit Rate
CMTS	Cable Modem Termination System
CM	Cable Modem
CM-CPI	Cable Modem to Customer Premises Equipment
CMTS-NSI	Cable Modem Termination System Network Side Interface
CMTS-RFI	Cable Modem Termination System Radio Frequency Interface
CATV	Community Antenna Television
DOCSIS	Data Over Cable Service Interface Specification
DHCP	Dynamic Host Configuration Protocol
DSL	Digital Subscriber Line
P-FCFS	Prioritized First Come First Served
HFC	Hybrid Fiber Coax
IP	Internet Protocol
IE	Information Elements
IETF	Internet Engineering Task Force
ICMP	Internet Control Message Protocol
ISDN	Integrated Services Digital Network
MAP	MAC Management Message
OPNET	Optimized Network Engineering Tool

Chapter 1

INTRODUCTION

1.1 Background

Rapid growth of the number of residential Internet users and increased bandwidth requirements of multimedia applications, have necessitated the introduction of an access network that would provide remote access with capabilities exceeding those of the Plain Old Telephony System (POTS) [35].

Community Antenna TeleVision (CATV) networks are an alternative network technology which have been deployed since the 60's and these offer a significant increase in spare bandwidth. However CATV networks are mainly used for analogue audio and video broadcasting from the Headend (HE) to the customers (downstream broadcasting). They are based on the tree and branch architecture with the broadcasting node at the root and the recipients at the leaves. The return path poses challenges such as noisy upstream channels and long propagation delays which render random access protocols highly inefficient [12]. Thus the need for a data protocol for point-to-point data communication that will make efficient use of the network spare capacity was intensified.

Organizations involved in the development of Hybrid Fiber Coax (HFC) network standard are: *a)* IEEE 802.14 group [16], [15], *b)* Data Over Cable Service Interface Specification (DOCSIS) group [17], *c)* Society of Cable Telecommunication Engineers (SCTE), *d)* Digital Audio Visual Council (DAVIC) and, *e)* Digital Video Broadcasting (DVB) group. The Internet Engineering Task Force (IETF) is contributing towards Internet Protocol (IP) delivery on top of cable systems through the IP Over Cable Data Network (IPCDN) group.

The DOCSIS specification was driven by the North American cable modem industry and was the first to be finalized. SCTE adopted the specification and submitted it to the International Telecommunication Union (ITU) for approval [35]. Therefore the DOCSIS was the first specification to reach international standard level as ITU-T recommendation *J*.112.

The DOCSIS specification has also been adopted by all the major vendors. Inter-operable HE equipment and Cable Modems (CMs) are expected. DOCSIS is expected to be the most widely available protocol providing inter-operable devices for high speed residential access. IP telephony and multimedia isochronous services are a market that cable operators are targeting in order to add value and to compete with the Digital Subscriber Lines (DSL) offered by Telephone Companies (Telcos) [36]. As a shared medium access, the capability of CATV networks in terms of delivering such services with current protocols has been questioned.

The alloted bandwidth is divided into several channels some dedicated to downstream communication (from the headend to the stations), while others are for upstream transmission (from the stations to the headend). Since the stations have to share the upstream channels they must implement a Media Access Control (MAC) protocol [1].

A slotted structure is used for information frames. Since stations may be transmitting on different portion of the upstream frame, the headend and the stations need to be synchronized in time. The synchronization could be done during the ranging process after a station has started up.

As stated in [5], data streams are segmented into fixed sized frames. The upstream and downstream frames have the same duration, which in most cases is longer than the maximum round trip delay in the network. There is a timing relationship between the upstream frames and downstream frames. For example, when an upstream frame arrives at the headend, a downstream frame could be issued. In the upstream frames carry data slots and contention slots. Data slots contain data packets sent by stations. Data slots may be assigned to stations by reservation or used on contention basis. When a station wants to reserve one of several data slots, it must transmit a request in a contention slot. If one or more stations send their request in the same contention slot, they collide. The downstream frame is called the MAC Management Message (MMM). The MMM contains synchronization information and feedback information about collision status and allocation. It also carries data from the headend to the stations.

The MAC protocol also support higher layer services such as Constant Bit Rate (CBR), Variable Bit Rate (VBR) and Available Bit Rate (ABR). To support a constant bit rate connection, the headend use bandwidth allocation mechanism called unsolicited grant service. The MAC needs also to differentiate between different connections in order to guarantee Quality of Service (QoS). This could be implemented via a priority mechanism.

The bandwidth request mechanism at the station operate as follows. The station may be able to send a request in a slot for one or more packets awaiting transmission. If a packet is generated, while station is resolving collision, the request size may be increased at the next transmission trial. The request size is however limited by the request field size. The station may also use piggybacked requests in data slots as an additional means for bandwidth request.

The bandwidth allocation is entirely controlled by the headend. First, it must decide the proportion of Contention Slots (CS) and Data Slots (DS) per frame or (CS/DS) ratio. This can be fixed or can vary depending on the traffic. The headend must then allocate the data slots to the stations. The headend may use schemes such as First Come First Serve (FCFS) or round robin to handle station request. It must also cope with constant bit rate connection and priorities.

A collision resolution algorithm needs to be implemented because request packets and possibly data packets are transmitted in contention fashion [12]. A wide variety of algorithms can be used. It must be noticed however that the characteristics of HFC networks protocol impose a number of constraints. First, since stations cannot monitor collisions, feedback information about contention requests must be provided by the headend. Secondly, the algorithm also takes into account the delay before a station receives the feedback information. If the number of contention slots is small and if they are located at the beginning of the frame, it may be possible for a station to receive feedback before the end of the frame. In this case, the station can attempt a retransmission in the next frame if its request collided. However, if the number of contention slots is large or variable, station may have to skip a frame before retransmitting a collided request.

Optimized Network Engineering Tool (OPNET) [22] provides a convenient environment for modeling and simulating DOCSIS HFC network protocol. Both the behavior and the performance of modeled HFC networks can be analyzed by performing discrete event simulations. OPNET consists of a number of tools, each one focusing on a particular stage of the modeling task, specification, simulation, data collection analysis. OPNET supports model specification with four tools, called Network Editor, Process Editor, Node Editor, Parameter Editor. OPNET models developed with those editors are structured hierarchically, in a manner that parallels real networks [22]. Traffic generators with arbitrary traffic characteristics can be created by user-defined OPNET process models. The priority bandwidth allocation mechanism can also be implemented.

1.2 Motivation

Cable operators are interested in deploying high-speed packet based communication system on cable television system that are capable of supporting a wide variety of services. The services include packet telephony service, video conferencing, $T1$ and frame relay equivalent service and many others.

The intended service will allow transparent bi-directional transfer of IP traffic, between the cable system headend and customer locations, over an *all-coaxial* or *hybrid-fiber coax* cable network.

The transmission path over the cable system is realized at the headend by CMTS, and at each customer location by a Cable Modem (CM). At the headend (or hub), the interface to the data-over-cable system is called Cable Modem Termination System-Network-

Side Interface (CMTS-NSI). At the customer locations, the interface is called the cable-modem-to-customer-premises-equipment (CM-CPI) as shown in figure 2.1. The intent is for operators to transparently transfer IP traffic between these interfaces, including but not limited to datagrams, DHCP, ICMP, broadcast and multicast.

1.3 Research Goals

This thesis focus on the capacity and performance characterization of the DOCSIS 1.1 protocol for the delivery of various traffic class streams, given the limited QoS features of the protocol. The emphasis will be more on the upstream side, due to the fact that the upstream is more vulnerable to noise, due to noise accumulation from the customers side and the MAC access mechanism that result in collisions which in turn results in degraded performance. Specifically, the objectives of the of this thesis are as follows:

1. Investigate the efficient classification methods for packet based multimedia (Voice, Video and Data) over the HFC network.

2. Develop an efficient packet classification method that can be incorporated in an HFC network environment in real time. The new mechanism should have less signaling overhead and storage of state information.

3. Investigate the efficient scheduling of packet-based multimedia (Voice, Video and Data) over the HFC network.

4. Conduct simulations in OPNET to evaluate the performance of the protocol for the delivery of various traffic class streams.

1.4 Existing Approach

Several approaches have been done in the literature to optimize the HFC network protocol so as to improve the network throughput, capacity and to minimize the network access delay. This is done mainly for the efficient delivery of multimedia applications (delay sensitive applications).

The approaches include re-designing the new HFC MAC protocol [9], modifying the existing MAC protocol [23], re-designing most efficient CMTS load scheduling algorithm [25, 24] and re-designing highly efficient contention resolution algorithm [8]. The collision as stated in [8] is one of the cause of throughput and capacity degradation and increase in network access delay. These approaches fail to improve the network capacity utilization more than 73%. This is due to the fact that, the upstream traffic plays a major role in determining the overall network performance.

One of the aims of this thesis is to improve the capacity utilization of the network well above 73% with existing HFC MAC protocol and CMTS load scheduling algorithm, by optimizing the upstream traffic.

1.5 Outline of the Thesis

In chapter 2, I examine the HFC network architecture and MAC protocol, while in chapter 3, I discuss the DOCSIS OPNET model components and their functional descriptions. Chapter 4 focuses on the traffic classification on the HFC network with specific application to DOCSIS 1.1 protocol. Simulation results are presented in chapter 5, while chapter 6 concludes the thesis.

Chapter 2

HFC NETWORK ARCHITECTURE AND MAC PROTOCOL

This chapter describes the characteristics of cable television network to be assumed for the purpose of operating a data-over-cable system (DOCSIS), followed by HFC MAC protocol operation. The interface specified in this thesis, is commonly referred to as DOCSIS 1.1, which is an extension of DOCSIS 1.0. These specifications are entirely backwards and forwards compatible with the previous specifications [17]. DOCSIS 1.1 compliant CMs must operate seamlessly with DOCSIS 1.0 CMTSs. DOCSIS 1.1 compliant CMTSs must seamlessly support DOCSIS 1.0 CMs.

2.1 DOCSIS 1.1 Reference Architecture

The reference architecture for the data-over-cable services and interface is shown in figure 2.1. The basic reference architecture involves four categories of interface [17].

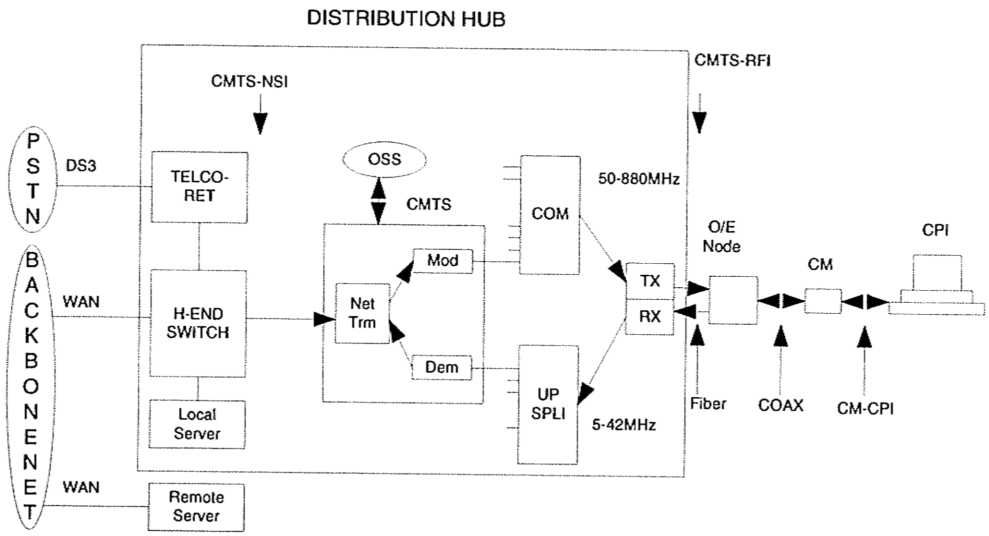

Figure 2.1: Data-Over-Cable Reference Architecture.

1. **Data Interfaces**

   There are the CM-CPI and CMTS-NSI, corresponding respectively to the cable-modem-to-customer-premises-equipment (CPE) interface (for example, between the customer's computer and the cable modem), and the cable modem termination system network-side interface between the cable modem termination system and the data network [11].

2. **Operation Support System Interface**

   These are network element management layer interfaces between the network elements and high-level Operation Support Systems (OSSs) which support the basic business processes [20].

3. **Telephone Return Interfaces**

   This is an interface between the cable modem and a telephone return path.

4. **RF Interfaces**

- Between the cable modem and the cable network [11].
- Between the CMTS and the cable network, in the downstream direction (traffic towards the customer) [18].
- Between the CMTS and the cable network, in the upstream direction (traffic from the customer) [18].

2.2 Broadband Access Network

A coaxial-based broadband access network is assumed for the network. This may take the form of either an all-coax or hybrid-fiber coax (HFC) network [21, 18]. A cable network uses a shared-medium, tree-and-branch architecture with analog transmission. The key functional characteristics of the DOCSIS 1.1 cable network are the following:

1. Two-way transmission.
2. A maximum optical-electrical spacing between the CMTS and the most distant CM of 100 miles.

2.3 Frequency Plan

In the downstream direction, the cable system is assumed to have a passband with a lower edge between 50 and $54MHz$ and an upper edge that is implementation-dependent but is typically in the range of 300 to $864MHz$ [19]. Within the passband, NTSC analog television signals in $6MHz$ channels are assumed to be present, as well as other narrow-band and wideband digital signals.

In the upstream direction, the cable system may have a sub split $(5 - 30MHz)$ or extended sub split ($5-40$ or $5-42MHz$) passband. NTSC analog television signals in $6MHz$ channels may be present, as well as other signals.

2.3.1 Transmission Downstream

The RF channel transmission characteristics of the cable network in the downstream direction are described in table 2.1 [17]. These numbers assume total average power of the digital signal in a $6MHz$ channel bandwidth for carrier levels. For impairment levels, the numbers in table 2.1 assume average power in a bandwidth in which the impairment levels are measured. For analog signal levels, the numbers in table 2.1 assume peak envelop power in a $6MHz$ channel bandwidth.

2.3.2 Transmission Upstream

The RF channel transmission characteristics of the cable network in the upstream direction are described in table 2.2 [17].

2.3.3 Upstream Interval, "Minislots"

The upstream transmission time-line as stated in [5] is divided into intervals by the upstream bandwidth allocation mechanism. Each interval is an integral number of minislots. A "minislot" is the unit of granularity for upstream transmission opportunities.

Each interval is labeled with a usage code which defines both the type of traffic that can be transmitted during that interval and the physical-layer modulation encoding. A minislot is a power-of-two multiple of $6.25\mu s$ increments, i.e. $(2, 4, 8, 16, 32, 64$, or $128)$ times $6.25\mu s$.

2.3.4 Frame

A frame is a unit of data exchange between two or more entities at the data link layer. A MAC frame consists of a MAC Header (beginning with a Frame Control byte figure 2.4), and may incorporate a variable-length Protocol Data Unit (PDU). The variable-length PDU includes a pair of 48-bit addresses, data and a Cyclic Redundancy Checksum (CRC). In special case, the MAC Header may encapsulate multiple MAC frames in to single MAC frame.

2.3.5 Data Forwarding Through the CM and CMTS

Data forwarding through the CMTS can either be transparent bridging or may employ network-layer forwarding (routing, IP switching) as shown in figure 2.2. If network layer forwarding is used, then the CMTS should conform to IETF router requirements [RFC-1812] with respect to its CMTS-RFI and CMTS-NSI interfaces [17]. Data forwarding

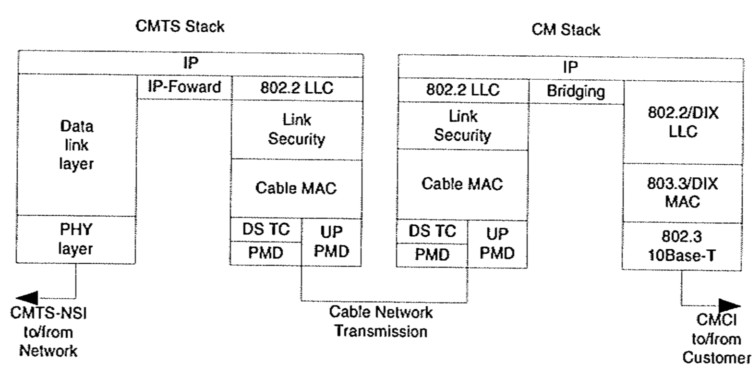

Figure 2.2: Data-Over-Cable CM to CMTS Forwarding, Protocol Stack.

through the CM is link-layer transparent bridging as shown in figure 2.2. Forwarding of IP traffic and other network layer protocols are supported.

Conceptually, the CMTS forwards data packets at two abstract interfaces, between the

CMTS-RFI and the CMTS-NSI, and between upstream and downstream channels.

2.3.6 Service Flows

The concept of Service Flows is central to the operation of the DOCSIS MAC protocol. Service Flows provide a traffic classification mechanism for upstream and downstream QoS management. In particular, they are integral to bandwidth allocation.

A service Flow ID (SFID) defines a particular unidirectional mapping between a CM and CMTS. Active upstream Service Flow IDs also have associated Service IDs or SIDs. Upstream bandwidth is allocated to SIDs, and hence to CMs, by the CMTS. Service IDs provide the mechanism by which upstream QoS is implemented.

2.4 DOCSIS MAC Frame Format

A MAC frame is the basic unit of transfer between MAC sublayer at the CMTS and the cable modem. The same basic structure is used in both the upstream and downstream direction [17]. MAC frames are variable in length. There are three distinct regions as

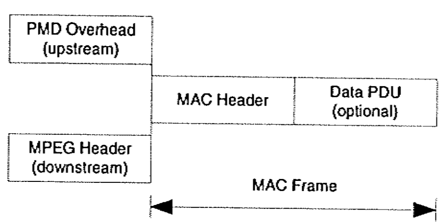

Figure 2.3: Generic MAC Frame Format.

shown in figure 2.3. Proceeding the MAC frame is either Physical Media Dependant (PMD) sublayer overhead (upstream) or an MPEG transmission convergence header (downstream). The first part of the MAC frame is the MAC Header. The MAC Header uniquely identifies the contents of the MAC frame. Following the header is an optional data PDU region.

2.4.1 PMD Overhead

In the upstream direction, the Physical (PHY) layer indicates the start of the MAC frame to the MAC sublayer. From the MAC sublayers perspective, it only needs to know the total amount of overhead so it can account for it in the Bandwidth Allocation process. The Forward Error Correction (FEC) overhead is spread throughout the MAC frame and is assumed to be transparent to the MAC data stream.

2.4.2 MAC Header Format

The MAC header format is as shown in figure 2.4 and table 2.3 shows the format of the headers. The frame Control (FC) field is the first byte and uniquely identifies the rest contents within the MAC Header. The FC field is followed by 3 bytes of MAC control, an optional Extended Header field (EHDR) plus a Header Check Sequence (HCS) to ensure the integrity of the MAC Header. The FC field is broken down into the FC-TYPE

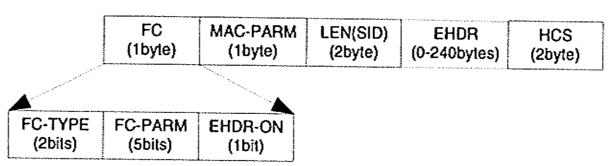

Figure 2.4: MAC Header Format.

sub-field, and EHDR_ON indication flag. The format of the FC field is as shown in table (2.4).

2.4.3 Packet-Based MAC Frames

The MAC sublayer support a variable-length Ethernet-type Packet Data PDU. Normally, the packet PDU is passed across the network in its entirety, including its original CRC. A unique Packet MAC Header is appended to the beginning. The frame format without an Extended Header is as shown in figure 2.5 and table 2.5. In addition to Ethernet type PDU, DOCSIS MAC also has the support of an ATM PDU.

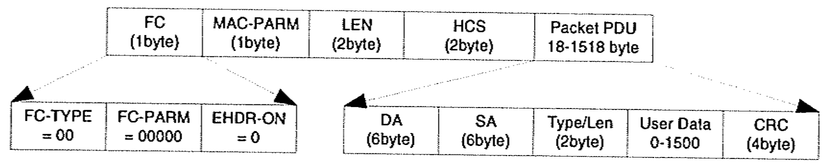

Figure 2.5: Ethernet 802.3 Packet Protocol Data Unit (PDU) Format.

2.5 Upstream Bandwidth Allocation

The upstream channel is modeled as a stream of minislots. The CMTS generate a time reference for identifying these slots. It also control access to these slots by the cable modems. It may grant some number of contiguous slots to CM for it to transmit a data PDU. The CM must time its transmission so that the CMTS receives it in the time reference specified.

The allocation MAP is a MAC management message transmitted by the CMTS on the downstream channel which describes, for some interval, the use to which the upstream minislot must be put. A given MAP may describe some slots as grants for particular station to transmit data in, other slots as available for contention transmission, and other slots as an opportunity for new station to join the link.

2.6 The Allocation MAP Management Message

The allocation MAP is a varying-length MAC Management message that is transmitted by the CMTS to define transmission opportunities on the upstream channel. It includes a fixed-length header followed by a variable number of information elements (IEs) [17]. Each information element defines the allowed usage for the range of minislots. There are basically eight types of information elements, as follows:

1. **The request IE**

   The request IE provides an upstream interval in which requests may be made for bandwidth for upstream data transmission. The character of this IE changes depending on the class of service ID. If broadcast, this is an invitation for CMs to contend for request. If unicast, this is an invitation for a particular CM to request bandwidth.

2. **The Request-Data IE**

   The Request-Data IE provides an upstream interval in which requests for bandwidth or short data packets may be transmitted.

3. **The Initial Maintenance IE**

   The Initial Maintenance IE provides an interval in which new stations may join the network. A long interval, equivalent to maximum round-trip propagation delay plus the transmission time of the Ranging Request message, must be provided to allow new stations to perform initial ranging.

4. **The station Maintenance IE**

   The Station Maintenance IE provides an interval in which stations are expected to perform some aspect of network maintenance, such as ranging and power adjustment.

5. **Short and Long Data Grant IEs**

   The Short and Long Data Grant IEs provide an opportunity for a CM to transmit one or more upstream PDUs. These IEs are issued either in response to a request from a station, or because of an administrative policy providing some amount of bandwidth to a particular station.

6. **Data Acknowledge IE**

   The Data Acknowledge IE acknowledges that a data PDU was received. The CM must have requested this acknowledgement within the data PDU.

7. **Expansion IE**

   The Expansion IE provides means of future expansion, if more than 32 bits are needed for future IEs.

8. **Null IE**

   A Null IE terminates all actual allocations in the list. All Data Acknowledge IEs and all Data Grant Pending IEs are followed by a Null IE.

2.7 Contention Resolution Algorithm

A collision resolution algorithms need to be implemented because request packets and possibly data packets are transmitted in contention fashion. There are basically two types of contention resolution algorithms. These algorithms are, adaptive $p-$persistence and the tree-based mechanisms.

2.7.1 Tree-based Contention Resolution Algorithms

The principle of the $n-$ary tree based contention resolution algorithms [6], is that when a collision occurs, all the stations involved in this collision split into n subsets, each of them randomly select a number between 1 and n. The basic idea is to allow different subsets to resolve their collisions sequentially. The stations in the first subset retransmit first, while the subsets from 2 to n wait for their turn.

One can see the waiting subset as a stack. The position in the stack represents the number of slots the station have to wait before it can retransmit its request. If a second collision occurs, the first subset split again. The subsets that are already waiting in the stack must be shifted up by $n-1$ positions in the stack to leave room for the new stations that collided. If no collision occur, the stations occupying the lowest stack level can transmit.

In its original definition, the algorithm assumed that stations receive feedback immediately after they have sent a request, so in case of collision they can retransmit immediately. However, in the HFC system the stations must wait at least until the beginning of the next frame before they receive feedback from the headend and thus are able to retransmit.

In order to accommodate the delay in feedback, the algorithm is as follows for the HFC system. Let's consider frame $j-1$ containing c_{j-1} collided minislots. All stations involved in a collided minislot, split into n subsets, each choosing a random number between 1 and n. The stations involved in the first collided slot, are dispatched in the first n slots, those involved in the second collided slot, use the n next slots, and more generally, the stations involved in the i^{th} collided slot, select a subset between $(i*n+1)$ and $((i+1)*n)$ slots.

If frame j contains p contention slots, then the first p subsets are able to retransmit, the i^{th} subset transmitting in the i^{th} slot. The other subset wait in the stack. If c_j new collisions occur in frame j, the waiting subset must be shifted by $(n * c_j - p)$ positions in the stack, to make room for new subsets.

New packet arrivals may be handled in two different ways. If tree-based algorithm is non blocking [6], new stations transmit without waiting in any slot selected randomly. On the other hand, if the algorithm is blocking, the new comers are not allowed to use a slot reserved for collision resolution. In other words, they are directly put on top of the stack. When new stations are able to transmit, they randomly select a slot among the remaining available slots.

2.7.2 Adaptive $p-$Persistence Contention Resolution Algorithm

The adaptive $p-$persistence algorithm proposed in [4] is an adaptation of stabilized $ALOHA$ protocol to frames with multiple contention slots. Newly active stations and stations resolving collisions have an equal probability of access p ($p-$persistent) to contention slots within a frame. p is determined by an estimate of the number of the backlogged stations, computed by the CMTS and sent to the stations in the downstream frames. The estimate $\hat{N}(j+1)$ [4] of the number of the backlogged stations in the $j+1^{th}$ frame is determined by:

$$\mathbf{X} = min[n, \hat{N}(j-1) - n_i(j-1) - n_s(j-1) + \frac{n_c(j-1)}{(e-2)} + \frac{MS(j-1)}{e}] \tag{2.1}$$

$$\hat{N}(j+1) = max\{\mathbf{X}, MS(j+1)\} \tag{2.2}$$

Where n is the number of stations, $MS(j)$ is the number of minislots in the j^{th} frame, $n_i(j)$, $n_s(j)$, $n_c(j)$, are respectively the number of idle, successful and collided minislots in the j^{th} frame.

Here, the estimate for the $(j+1)^{th}$ frame is determined by the parameters of the $(j-1)^{th}$ frame. This is due to the feedback from the $(j-1)^{th}$ frame not being received at all stations before the beginning of the j^{th} frame when frames consist of all minislots .

However the number of minislots allowed in a frame may be restricted so that feedback for a frame is received at all stations before the beginning of the next frame. If that is the case, then the estimate for the j^{th} frame is determined by the parameters of the $(j-1)^{th}$ frame.

When a station needs to make a request in frame j, it generates a random number i_j uniformly distributed in the interval $[1, \hat{N}(j)]$. If i_j is less or equal to the number of minislots in the frame it will make its request in the i_j^{th} minislot, otherwise it will attempt to make its request in the next frame using the estimate for that particular frame.

2.8 Fixed Contention/Data(CS/DS) Ratio

The fixed allocation is simple, however the optimum CS/DS ratio depends on the traffic pattern. It was shown in [12] that when the ratio of CS/DS increases, the mean delays at higher applied loads become larger because less DSs are available to data transmission. Lower CS/DS ratio performs well for smaller packet size, but as the average packet sizes became larger, this result in larger request sizes so that less contention minislots are needed.

2.9 Variable Contention/Data (CS/DS) Ratio

The variable allocation schemes adapt the CS/DS ratio according to the traffic. In the scheme used in [12] the headend computes the required number of data slots and converts the remaining slots to contention minislots. This algorithm is referred to as data priority allocation scheme since the number of data slots is computed first. A different approach proposed in [34] consists of computing the required number of minislots and leaves the remainder as data slots. This is referred to as minislot priority allocation algorithm. The proposed algorithm in [34] modified in [12] is as follows:

$$j_x = max\{\lceil\frac{3S}{3+mk_e}\rceil, 1\} \tag{2.3}$$

Where S is the maximum number of data slots in a frame, m is the number of minislots resulting from the conversion of a data slot to minislots, and k_e is the estimated average number of data slots reserved per request and computed according to:

$$k_e = max\{\lfloor k * f_m \rfloor, 1\} \tag{2.4}$$

with f_m being an estimate factor set to 0.5. k is updated at the headend based on the request size l_r and using $\alpha = \frac{1}{16}$

$$k_{i+1} = l_r * \alpha + (1-\alpha) * k_i \tag{2.5}$$

Then, the number of minislots MS in frame j is given by

$$MS(j) = \begin{cases} m(j_x - 1) & \text{if } j_x \geq 2 \text{ and } \frac{(j_x-1)mk_e}{3} \geq S - j_x \\ mj_x & \text{else} \end{cases} \tag{2.6}$$

It should be noted that, the allocation does not depend on the number of requests accumulated at the headend and also a minimum of one data slot is always converted to minislot.

Table 2.1: Downstream RF Channel Transmission Characteristics.

Parameter	Value
Frequency range	Downstream $88MHz$ to $860MHz$
RF channel spacing	$6MHz$
Transit delay from headend to most distant customer	$\leq 0.800msec$
Carrier-to-noise ratio in $6MHz$ band	$\geq 35dB$
Carrier-to-Composite second order distortion ratio	$\geq 41dB$
Carrier-to-Composite second triple beat distortion ratio	$\geq 41dB$
Carrier-to-Cross-modulation ratio	$\geq 41dB$
Carrier-to-any other discrete interference (ingress)	$\geq 41dB$
Amplitude ripple	$3dB$ within the design bandwidth
Group delay ripple	$75ns$ within the design bandwidth
Micro-reflections bound for dominant echo	$-20dBc$ @ $\leq 1.5\mu sec$, $-30dBc$ @ $> 1.5\mu sec$ $-10dBc$ @ $\leq 0.5\mu sec$, $-15dBc$ @ $\leq 1.0\mu sec$
Carrier hum modulation	$\leq -26dBc$
Burst noise	$\leq 25\mu sec$
Maximum analog video carrier level at the CM input	$17dBmV$
Maximum number of analog carriers	121

Table 2.2: Upstream RF Channel Transmission Characteristics.

Parameter	Value
Frequency range	5 to $42MHz$ edge to edge
Transit delay from the most distant to the nearest CM or CMTS	$\leq 0.800msec$
Carrier-to-interference plus ingress	$\geq 25dB$
Carrier hum modulation	$\leq -23dBc$
Burst noise	$\leq 10\mu sec$ at a $1KHz$
Amplitude ripple $5 - 42MHz$	$0.5dB/MHz$
Group delay ripple $5 - 42MHz$	$200ns/MHz$
Micro-reflections	$-10dBc$ @ $\leq 0.5\mu sec$ $-20dBc$ @ $\leq 1.0\mu sec$ $-30dBc$ @ $> 1.0\mu sec$

Table 2.3: Generic MAC Header Format.

MAC Header Field	Usage	Size
FC	Frame Control: Identifies type of MAC Header	8bits
MAC_PARM	Parameter field is dependent on FC: if $EHDR\_ON = 1$, used for EHDR field length if concatenated frame used for MAC frame count else for request indicates number of mini-slots	8bits
LEN(SID)	The length of MAC frame	16bits
EHDR	Extended MAC Header	$0 - 240$bytes
HCS	MAC Header Check Sequence	2bytes
	Length of a MAC Header	6bytes + EHDR

Table 2.4: Frame Control (FC) Field Format.

FC Field	Usage	Size
FC_TYPE	MAC Frame Control Type field: 00 Packet PDU MAC Header 01 ATM PDU MAC Header 10 Reserved PDU MAC Header 11 MAC Specific Header	2bits
FC_PARM	Parameter bits, dependent on FC_TYPE	5bits
EHDR_ON	When $= 1$, indicates that EHDR field is present	1bit

Table 2.5: DOCSIS 1.1 Packet PDU Format.

FIELD	Usage	Size
FC_TYPE	FC_TYPE = 00; Pkt MAC Header EHDR_ON = 0, if there is no EHDR 1 there is an EHDR	8bits
MAC_PARM	MAC_PARM = x, Set to zero if there is no EHDR Otherwise set to length of EHDR	8bits
LEN	LEN = $n + x$ Length of packet PDU in bytes + EHDR	16bits
EHDR	Extended MAC Header, if present	0-240bytes
HCS	MAC Header Check Sequence	16bytes
Packet Data	Packet PDU DA - 48 bit Destination Address SA - 48 bit Source Address Type/Len - 16 bit Ethernet Type or Length Fields User Data (variable, 0 - 1500bytes) CRC - 32bit CRC over Packet PDU	n bytes
	Length of Packet MAC frame	$6 + x + n$ bytes

Chapter 3

DOCSIS OPNET MODEL

This chapter presents the methods used to specify the internal structure of the DOCSIS 1.1 network objects representing the CM, CMTS statistics collection and cloud. The next sections will define the capabilities of various types of modules and connections, and illustrate their specific uses within the node models. The remaining sections describe the techniques used in creating specific types of models, including traffic sources and queues.

3.1 Model Architecture

The model used for simulations was *Common Simulation Framework v.*13 (*CSF*13).The CSF is a baseline model based on the OPNET simulation package. Originally it was developed as a joint initiative by MIL3 INC and Cablelabs in order to produce the core model for simulating data protocols of HFC networks including the DOCSIS and the IEEE 802.14 protocols. Some suitable modifications were applied to the model in order to generate the required traffic sources and to optimize the model for simulating networks with up to 800 nodes. DOCSIS 1.1 model is split into three general components, as shown in figure 3.1. The Cable Modem Termination System (CMTS), the Cable Modem (CM),

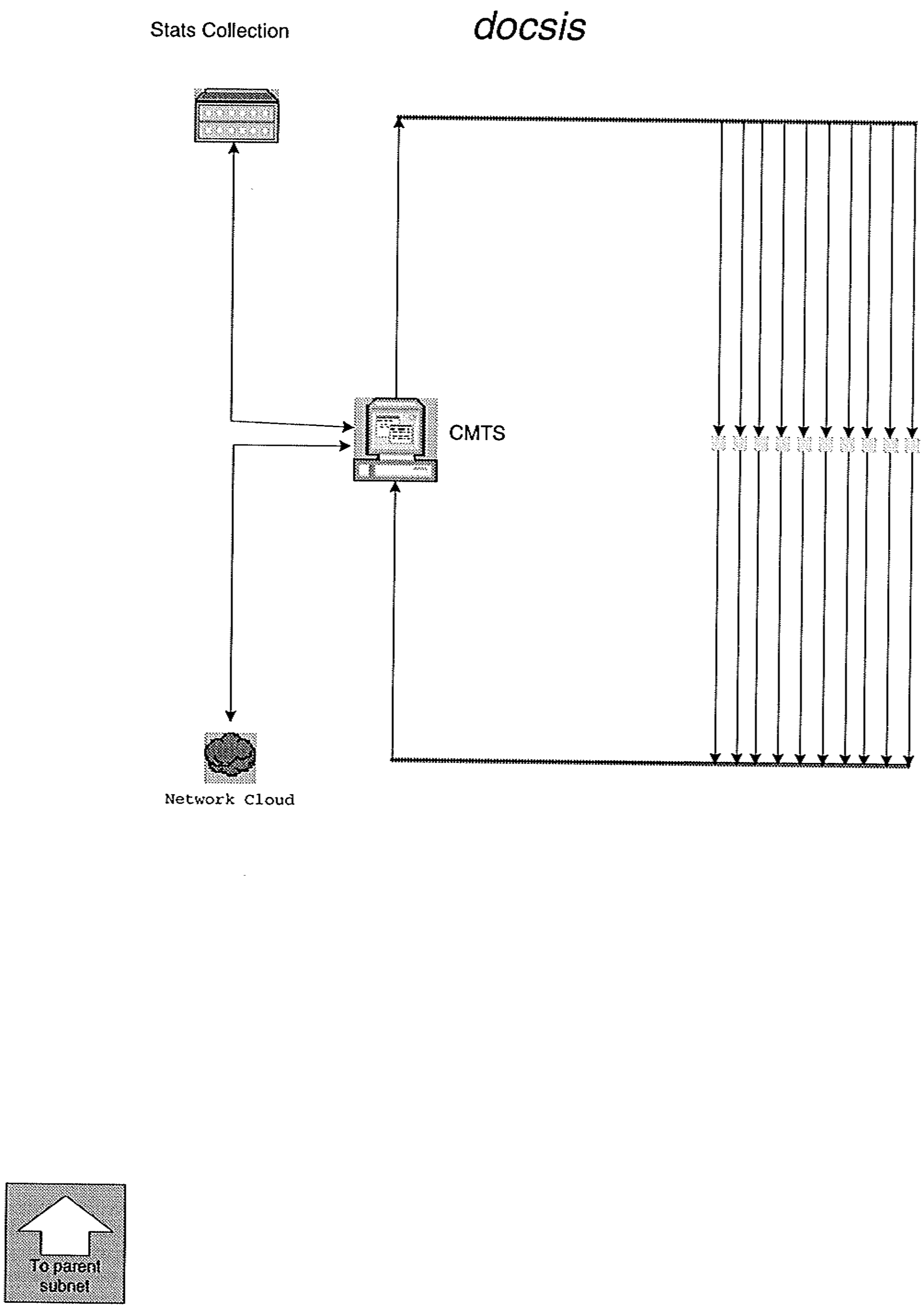

Figure 3.1: Cable TV Network Topology.

the Statistic Collection, the Network Cloud and the traffic sources used to generate data to be transmitted across the cable network.

The transmission path over the cable system is realized at the headend by the CMTS, and at each customer location by a CM. The CMTS performs protocol functions that encompass such traffic flow. The CM embodies the client side of the DOCSIS protocol and maintains a single queue for an attached device. The present functionality includes ranging, quality of service negotiation, request regions, request/data regions and adherence to upstream channel descriptions.

The model also contains a number of traffic sources for generating traffic to be communicated to the CMTS. These sources include a Poisson data source, a continuous bit-stream source, an HTTP source and voice/video data sources.

3.1.1 Cable Modem (CM)

The principal function of the cable modem system is to transmit packets from the subscriber to the headend across the cable network on the upstream channel. The node architecture for a CM is depicted in figure 3.2, each module in the architecture will be described in terms of its general functionality.

- **MAC**

  The basic functionality of this module is to take data from various higher layer traffic sources and transmit on the upstream channel. This is accomplished by communicating with the CMTS to allocate the necessary bandwidth. The traffic is initially queued until a transmission opportunity is found, at which point the traffic is sent to the physical layer module *(bus_tx)* for transmission.

  It is the responsibility of this module to receive traffic sent by the CMTS on the

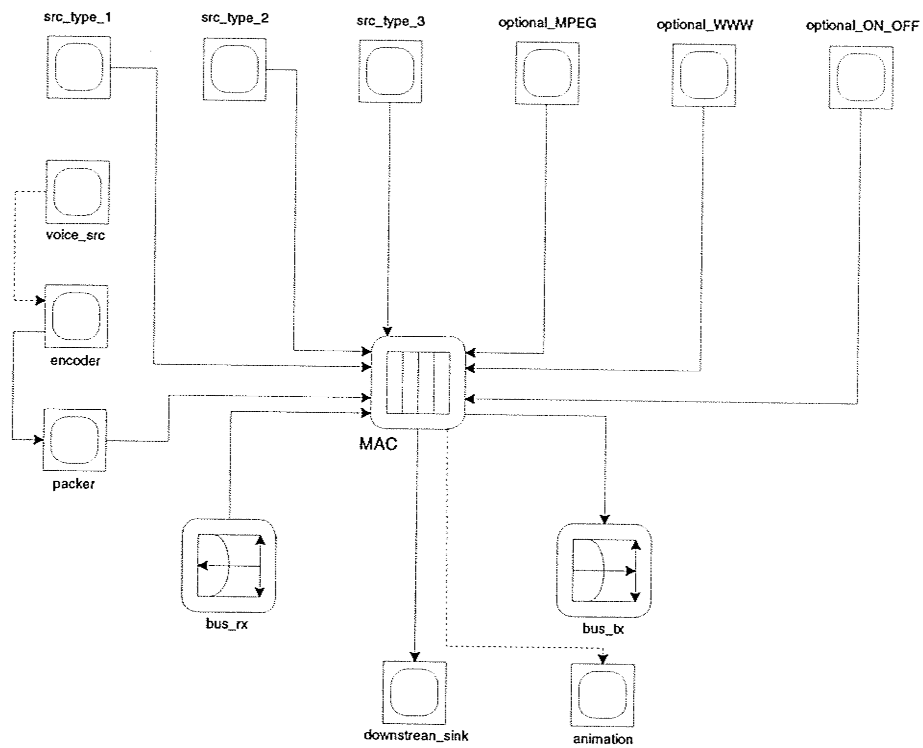

Figure 3.2: Cable Modem (CM) Node.

downstream channel. Once traffic is received by the physical layer module *(bus_rx)* it is forwarded to the MAC for processing. The MAC distinguishes application traffic from management traffic and forwards the application traffic to the higher layer module *(Downstream_sink)*.

- **Downstream_sink**

  This module receives traffic forwarded by the lower layer module *(MAC)*. Its sole responsibility is to record statistics and destroy the data, freeing up memory in the simulation.

- **bus_rx**

  This module receives traffic sent on the downstream channel from the CMTS. After receiving a complete packet, it sends the traffic to the higher layer module *(MAC)*.

- **bus_tx**

  This module transmits any traffic received from the higher layer module *(MAC)* on to the upstream data bus. The rate at which the information is transmitted is based on the data rate of the bus.

- **src_type_1**

  This module generates type 1 application traffic and sends it to the lower layer module *(MAC)* for transmission on the upstream channel to the CMTS.

- **src_type_2**

  This module generates type 2 application traffic and sends it to the lower layer module *(MAC)* for transmission on the upstream channel to the CMTS.

- **src_type_3**

  This module generates type 3 application traffic and sends it to the lower layer module *(MAC)* for transmission on the upstream channel to the CMTS.

- **optional_MPEG**

  This module generates MPEG (video) application traffic and sends it to the lower layer module *(MAC)* for transmission on the upstream channel to the CMTS.

- **optional_WWW**

  This module generates WWW application traffic and sends it to the lower layer module *(MAC)* for transmission on the upstream channel to the CMTS.

- **optional_ON_OFF**

  This module generates ON-OFF application traffic and sends it to the lower layer module *(MAC)* for transmission on the upstream channel to the CMTS.

- **voice_src**

  This module controls the voice speech levels that are fed into the encoder. Once a voice call is established, speech activity levels (talk or silence) are fed into the encoder for controlling the conversion of the voice signal into data samples.

- **Encoder**

  This module is responsible for taking the speech activity levels and modulating them into discrete representations. The encoder implements digital signal processing function for converting speech into data samples. These data samples are then fed into the pecketizer.

- **Packer**

  This module is responsible for accepting the speech data samples and encapsulating them into a data packet to be sent to the lower layer module *(MAC)* for communication across the cable network.

- **Animation**

  This module is responsible for drawing the custom animation depicting the instantaneous depth of the MAC queue. Each time a packet is inserted or removed from the queue, the statistic feeding the Animation module is altered to show the new queue size. This trigger an interrupt that invokes the animation for updating the custom graph.

3.1.2 Cable Modem Termination System (CMTS)

The main function of CMTS is to schedule transmissions on the upstream channel and manage flow on the downstream channel. Downstream channels are point-to-point whereas upstream channels are shared. The node architecture for the CMTS is as shown in figure 3.3. The following is the functional description of each node:

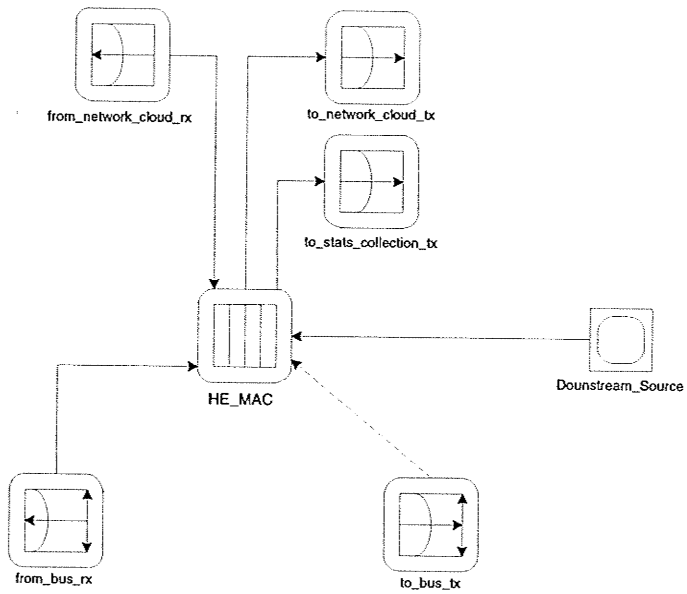

Figure 3.3: Cable Modem Termination System (CMTS) Node.

- **HE_MAC**

  The *(HE_MAC)* module has two general purposes:

  1. To receive and process information from the lower and higher layers.
  2. To send processed information to the lower and higher layers.

  The *(HE_MAC)* receives traffic sent on the upstream channel from a CM. It determines whether the traffic is a request for bandwidth or application traffic. If it is a request for bandwidth, the *(HE_MAC)* is responsible for invoking the appropriate scheduling algorithm for allocating the required bandwidth. If the lower layer is application traffic, the *(HE_MAC)* will forward the traffic to the higher layer module *(to_stats_ccollection_tx)*.

For information received from the higher layer module *(Downstream_source)*, the *(HE_MAC)* will process the information and schedule it to be sent to the destination CM. At the appropriate time, the information will be sent to the lower layer module *(to_bus_tx)* to be sent on the downstream channel. In order for the *(HE_MAC)* to be aware when the lower layer transmitter is idle, a statistic wire denoting the busy status of the transmitter is fed back into the *(HE_MAC)*. The *(HE_MAC)* will only send data to the transmitter if the transmitter is idle.

- **Downstream_source**

  This module is responsible for sending application traffic downstream to destination CM. Once the application traffic has been generated, it is fed down to the lower layer module *(HE_MAC)* for encapsulation and communication on the downstream channel.

- **to_bus_tx**

  This module transmit traffic received from the higher layer module *(HE_MAC)* on to the downstream bus. The rate at which information is transmitted is based on the data rate of the bus.

- **from_bus_rx**

  This module receives traffic sent on the upstream channel from a CM. After receiving a complete packet, it sends the traffic on the higher layer module *(HE_MAC)*.

- **to_stats_collection_tx**

  This module communicates application traffic sent from the lower layer module *(HE_MAC)* to the point-to-point link connected to the statistic collection object. The statistic collection object will process the data, record appropriate statistics and destroy the application traffic.

- **to_network_cloud_tx**

  This module takes application traffic sent from the lower layer module *(HE_MAC)* destined for a device outside the cable network and sends this information to the cloud object.

- **from_network_cloud_rx**

  This module takes application traffic sent from the cloud object and forwards it to the lower layer module *(HE_MAC)* for communication across the cable network.

3.1.3 Stats Collection

The function of the Stats Collection node is to receive application traffic forwarded from the CMTS, parse the traffic based on its type, and send it to the appropriate sink module as shown in figure 3.4. The various modules in this node model are:

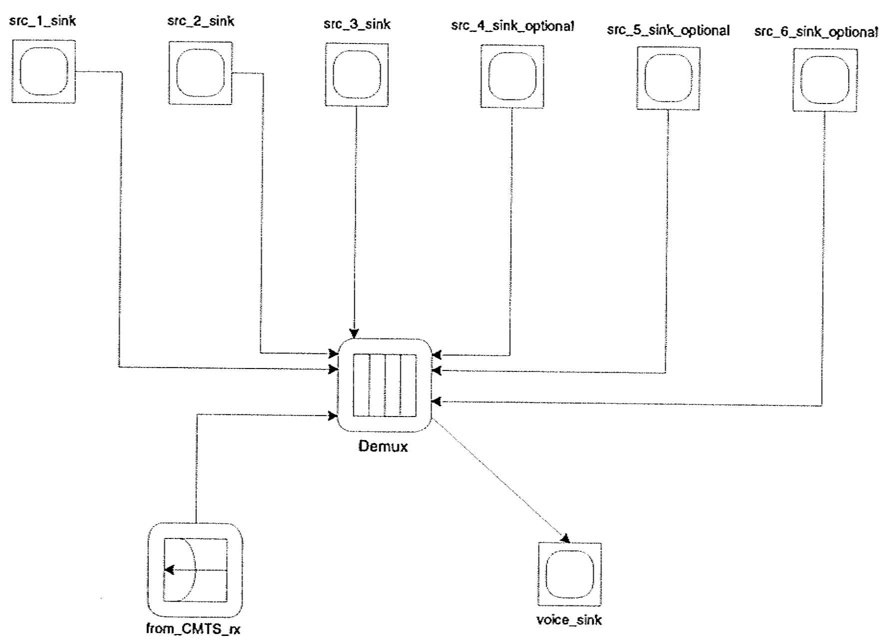

Figure 3.4: Statistic Collection Node.

- **Demux**

  This module is responsible for accepting application traffic forwarded from the lower layer module *(from_CMTS_rx)*, decomposing the traffic and sending it to the appropriate higher layer module (sink) for statistic collection and packet destruction. The decomposition involves assessing what type of application traffic has been received and forwarding on the appropriate output stream.

- **from_CMTS_rx**

  This module receives application traffic sent across the point-to-point link by the CMTS. After receiving a complete packet, the traffic is sent to the higher module *(Demux)*.

- **voice_sink**

  This module receives application traffic forwarded by the lower layer module *(Demux)*. It is responsible for recording voice related statistics and destroying the voice packet, freeing up the memory.

- **src_1_sink**

  This module receives type 1 application traffic forwarded by the lower layer module *(Demux)*. It is responsible for recording type 1 related statistics and destroying the voice packet, freeing up memory.

- **src_2_sink**

  This module receives type 2 application traffic forwarded by the lower layer module *(Demux)*. It is responsible for recording type 2 related statistics and destroying the voice packet, freeing up memory.

- **src_3_sink**

  This module receives type 3 application traffic forwarded by the lower layer module

(Demux). It is responsible for recording type 3 related statistics and destroying the voice packet, freeing up memory.

- **src_4_sink_optional**

  This module receives MPEG application traffic forwarded by the lower layer module *(Demux)*. It is responsible for recording MPEG related statistics and destroying the voice packet, freeing up memory.

- **src_5_sink_optional**

  This module receives WWW application traffic forwarded by the lower layer module *(Demux)*. It is responsible for recording WWW related statistics and destroying the voice packet, freeing up memory.

- **src_6_sink_optional**

  This module receives ON-OFF application traffic forwarded by the lower layer module *(Demux)*. It is responsible for recording ON-OFF related statistics and destroying the voice packet, freeing up memory.

3.2 Process Behavior

This section discusses the underlying process models and their behavior. For each process model, only the main responsibilities are discussed. The process model discussion incorporates three main categories, the CM the CMTS, and the application layer.

3.2.1 Cable Modem Manager Process

The cable modem medium access control *(MAC)* layer is represented in *OPNET* by the *docsis_cm_mgr* process model. The behavior of this process model can be broken into three categories, initialization and ranging, upstream communication, and downstream communication. Figure 3.5 depicts the CM manager process model.

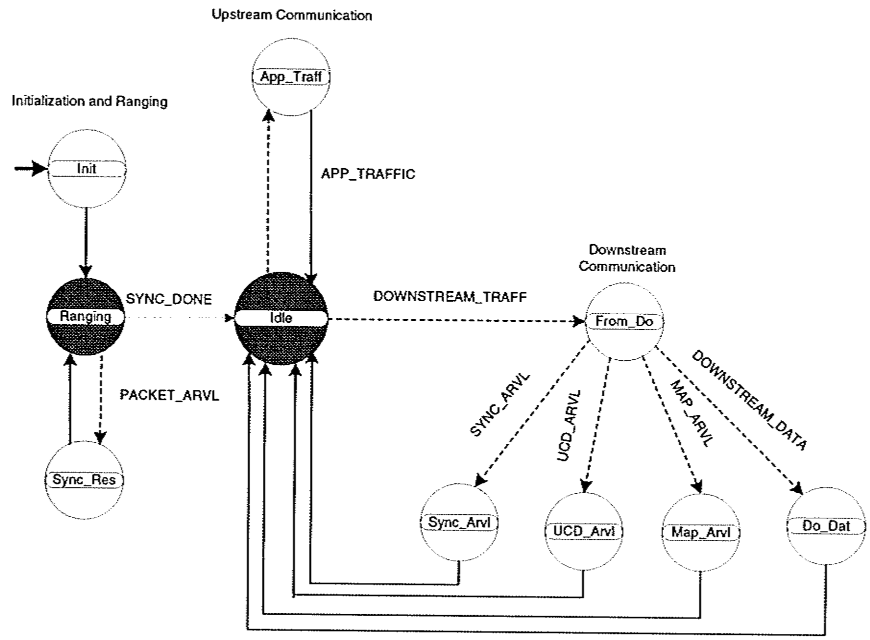

Figure 3.5: Cable Modem Manager Process.

3.2.2 Initialization and Ranging

Most Process models require some amount of initialization. It is important to designate an area of the process model to perform this initialization. The CM manager process does that in the initial state (the first state executed for the first process invocation).

The initialization is responsible for the following:

- Allocation of unique Serial Identification Number (SID).
- Assignment of CM priorities.
- Obtaining values for attributes and QoS characteristics.

When communicating traffic to the CMTS, the CM is responsible for making sure it gets to the CMTS at the proper time. Since upstream channels is divided in to multiple time slots (called minislots), the CM need to be sure a packet arrives at the CMTS on a time slot boundary. In order to guarantee this, the CM must synchronize its time clock to that of the CMTS. This require how long it takes the packet to travel from the CM to the CMTS. Ranging does just that. At initialization the CM sends a ranging request packet to the CMTS. Embedded in the ranging packet is the QoS information that will be extracted by the CMTS and used for bandwidth allocation purposes.

After ranging is completed, as denoted by receipt of a ranging response on the downstream channel from the CMTS, the CM invokes a child process for controlling all communication on the upstream channel. Synchronization and QoS information is passed to the child process for use during communication and bandwidth request.

3.2.3 Upstream Communication

When application traffic is sent to the MAC from the higher layer, the upstream communication section of the process model is invoked. The sole responsibility of this section is to invoke the child process for sending the application packet on the upstream channel.

3.2.4 Downstream Communication

When traffic is received by the MAC from the lower layer, the downstream communication section of the process model is invoked. The CM supports a set of expected types of traffic, upstream channel descriptor messages (UCD), synchronization messages (Sync), bandwidth allocation maps (MAP), and application traffic. The responsibility of this section is to determine what type of traffic has been received and process the traffic accordingly.

The CM will expect to receive periodic UCDs, Syncs, and Maps. When Map is received, the child process is invoked to parse the Map for information related to allocated bandwidth for this CM. When receiving application traffic, the traffic is forwarded to the higher layer process model responsible for recording application-related statistics.

- **Transitions**

  The process model contains the transitions between states as shown in table 3.1.

3.2.5 Cable Modem Child Process (docsis_cm_child)

As part of the cable modem MAC layer, a child process called *docsis_cm_child* exists with the responsibilities for all upstream communication. Figure 3.6 depicts the CM child process. The functional description of the states in the CM child process is as follows:

- **Init**

  The initialization state is responsible for initializing all variables, obtaining simulation and model attributes, and extracting information passed from the manger process such as QoS and Serial Identification Number (SID).

- **Q_Empty**

  This state is entered when the traffic queue for sending information on the upstream channel is empty. It waits for other two possible events, a Map arrival or a higher layer application packet arrival.

- **No_Request_Outstanding**

  This state is responsible for awaiting a transmission opportunity from a future map.

Table 3.1: Downstream Communication Transitions.

Condition	Description
SYNC_DONE	A SELF interrupt indicating the timeout for a ranging request.
PACKET_ARRIVAL	A STRM interrupt indicating the arrival of a ranging response packet.
APPLICATION_TRAFFIC	A STRM interrupt indicating the arrival of higher layer application traffic.
DOWNSTREAM_TRAFFIC	A STRM interrupt indicating the arrival of lower layer traffic.
DOWNSTREAM_DATA	Indicates application traffic has arrived on the downstream channel.
MAP_ARRIVAL	Indicates a Map message has arrived on the downstream channel.
UCD_ARRIVAL	Indicates a UCD message has arrived on the downstream channel.
SYNC_ARRIVAL	Indicates Sync message has arrived on the downstream channel.

It is entered when application traffic is waiting to be sent, however the current Map contains no transmission opportunity. This state has the potential to receive either a Map arrival or a higher layer application packet arrival. In the event of the Map arrival, a transmission opportunity is sought in the new Map for transmission of the queued application traffic. If more application traffic arrives from the higher layer,

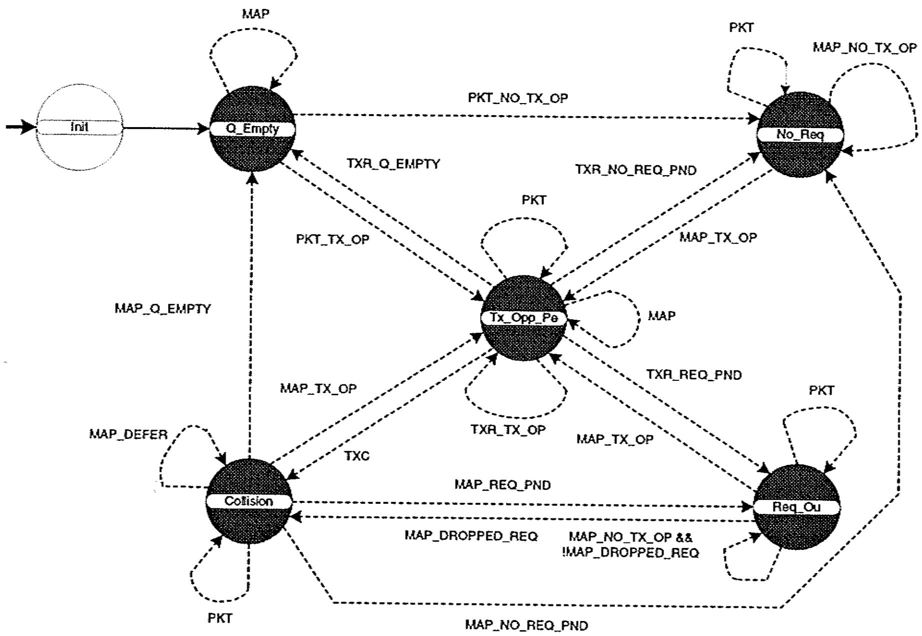

Figure 3.6: Docsis Cable Modem Child Process.

the application traffic is inserted in the queue behind other pending transmissions.

- **Tx_Opp_Pending**

  This state is responsible for awaiting a pending transmission. When a transmission opportunity is found, transmission will not immediately occur since the process must wait the appropriate time to transmit the traffic. Traffic can be sent either in contention slots or reserved slots, depending on the type of traffic and the configuration of the modem. The process waits for an event signaling the appropriate slot has arrived for transmission.

  In the event either a Map or additional application traffic is received, no operation are necessary since the process is busy sending traffic. The arriving application traffic will be queued and the next time a transmission opportunity is required, the

Map information will be parsed to search for this opportunity.

- **Collision Resolution**

  Since certain traffic can be sent in contention slots within the Map (such as bandwidth allocation requests), there exist a potential for a collision to occur and the information sent to the CMTS to be lost. If traffic is sent in a contention slot, this state is entered to perform collision detection. In the event a collision is detected, a deference interval is applied via a binary truncated backoff algorithm and a retransmission attempt is made. The collision detection is observed by parsing the Map and observing the an acknowledgement for that transmission by the CMTS.

- **Request_Outstanding**

  This state is responsible for awaiting an outstanding request for bandwidth. The state is entered when the CMTS sends a request pending signal, indicating that the request for bandwidth has been received, but allocation is pending. The CM cannot send the data until this bandwidth has been allocated. Therefore, any additional application traffic must be queued and any Map arrivals will be parsed looking for the allocated bandwidth.

- **Transitions**

  The process model contains the transitions between states as shown in table 3.2.

3.2.6 Cable Modem Termination System (CMTS)

The Cable Modem Termination System MAC layer is represented in OPNET by the *(docsis_cmts_mgr)* process model. The behavior of this model can be broken into three categories, initialization and ranging, upstream communication, and downstream communication. Figure 3.7 depicts the CMTS manager process model.

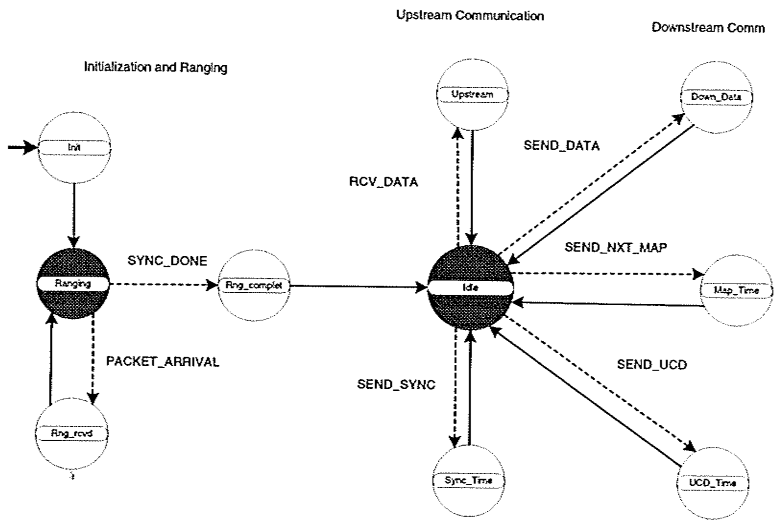

Figure 3.7: Docsis Cable Modem Termination System Manager Process.

- **Initialization and Ranging**

  The CMTS manager process performs initialization, as do most process models. The initialization is responsible for the following:

  - Creating appropriate lists for bandwidth allocation.
  - Obtaining values for attributes.
  - Establishing Map characteristics.
  - Registering statistics.

  The CMTS is mainly responsible for contention resolution in the upstream direction. It must maintain the scheduling algorithms used to allocate bandwidth and control the communication of traffic on the downstream channel.

  Prior to performing bandwidth allocation for any CM, all active CMs must be synchronized to the CMTS time clock. This occurs during the ranging process, as in the CM process model. During ranging phase, the CMTS must be responsible for accepting ranging request information from the CM and responding with a ranging response. The ranging response will contain the delay offset to be used by the CM.

Included in the ranging request packet is the QoS-related information for the single service flow established for each CM. The QoS information tells the CMTS whether the CM is requesting best effort, real-time polling, or unsolicited grant type of service. The CMTS will take this information and create the appropriate schedulers to handle bandwidth allocation for this service flow.

After ranging is completed, the CMTS will send the periodic Upstream Channel Descriptor (UCD), Sync, and Map messages. The initial Map will contain only contention slots, since no bandwidth has yet been requested by any CM.

- **Upstream Communication**

  When traffic is received by the MAC from the lower layer, the upstream communication section of the process model is invoked. The CMTS is responsible for accepting application traffic and request for bandwidth. For application traffic the, the traffic is forwarded to the statistic collection object for statistic reporting and packet destruction. For bandwidth allocation the requests, the request is inserted into a request queue, where it will be processed the next time a map is generated.

- **Downstream Communication**

  During Downstream communication, the CMTS is responsible for sending the UCD messages, Sync messages, Map messages, and application traffic. UCD, Sync, and Map messages are sent periodically. The Map message is created using the scheduling algorithms and informs the CM as to which slots are reserved and which are contention. All traffic communicated on the downstream is controlled by a child process, which is responsible for dequeuing and transmitting the information to the appropriate CMs.

- **Transitions**

  The model contains the transition between states as shown in table 3.3.

3.2.7 Cable Modem Termination System Transmission Controller

As part of the CMTS MAC layer, a child process called *(docsis_cmts_txctl)* exists with the responsibility of all downstream communication. Figure 3.8 depicts the CMTS Transmission Controller child process model. The following is the description of the states of the process:

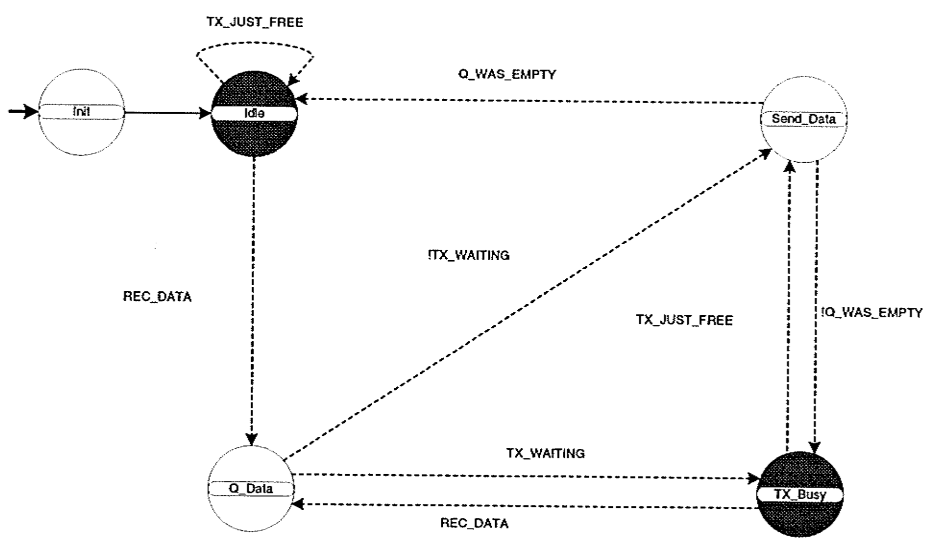

Figure 3.8: Docsis Cable Modem Termination System Transmission Controller Process.

- **Init**

  The initialization state performs two functions, obtain the process ID and performs interrupt type-based registration to direct all statistic interrupts directly to this process without having to go through the manager process.

- **Idle**

  This state waits for an event to occur. The possible events are either an invocation from the manager process signaling the requirement to enqueue or transmit information on the downstream channel, or a statistic wire indicating the transmitter is busy or idle.

- **Q_Data**

  In the event the transmitter is busy, if additional information is requested to be sent, the information must be enqueued. The model contains two queues, one for management messages and other for application traffic. Depending on the type of traffic received, the appropriate queue will be used to store the information until it arrives at the head of the queue and until the transmitter is idle.

- **Send_Data**

  If the transmitter is idle and there exists information to be sent on the downstream channel, this state dequeues the information and sends it to the transmitter. The management queue has priority over the application traffic queue, so any data will first be extracted from the management queue and sent.

- **TX_Busy**

  If the transmitter is busy and either application or management information is requested to be transmitted, this state enqueues the information for later transmission.

- **Transitions**

  The process model contains the transitions as shown in table 3.4.

Table 3.2: Docsis Child Process Transitions.

Condition	Description
PKT	Indicate the arrival of application Traffic.
PKT_NO_TX_OP	Indicate the arrival of application traffic, but the current parsed Map does not contain any transmission opportunities.
PKT_TX_OP	Indicates the arrival of application traffic, where the current parsed Map contains a transmission opportunity.
MAP	Indicates the arrival of a Map.
MAP_TX_OP	Indicates the arrival of a Map, where the Map contains a transmission opportunity.
MAP_NO_TX_OP	Indicates the arrival of a Map, but the Map does not contain a transmission opportunity.
MAP_DROPPED_REQ	Indicates the arrival of a Map signally the the request for bandwidth has been dropped.
MAP_REQ_PND	Indicates the arrival of a Map signally the request for bandwidth is pending.
MAP_DEFER	Indicates that a Map has arrived without an acknowledgement for traffic having been sent in a contention slot. The process will defer and attempt to communicate the information again.
MAP_NO_REQ_PND	Indicates the arrival of a Map, signally no request is pending.
MAP_Q_EMPTY	Indicates the arrival of a Map when the application traffic queue is empty.
TXR_TX_OP	Indicates the arrival of a reserved slot for transmission.

TXR_Q_EMPTY	Indicates the arrival of reserved slot for transmission, but the queue is empty (no traffic to send)
TXR_REQ_PND	Indicates the arrival of a reserved slot for transmission, but the current request for bandwidth is pending.
TXR_NO_REQ_PND	Indicates the arrival of a reserved slot for transmission, but there is no pending request for bandwidth.
TXC	Indicates the arrival of a contention slot for transmission.

Table 3.3: Docsis Cable Modem Termination System Manager Transitions.

Condition	Description
SYNC_DONE	A SELF interrupt indicating the timeout for a ranging request.
PACKET_ARRIVAL	A STRM interrupt indicating the arrival of a ranging request.
RCV_DATA	A STRM interrupt indicating the arrival of traffic from the lower layer.
SEND_DATA	A STRM interrupt indicating the arrival of application traffic from a higher layer
SEND_NXT_MAP	A SELF interrupt indicating that a Map message will be generated.
SEND_UCD	A SELF interrupt indicating that a UCD message will be generated.
SEND_SYNC	A SELF interrupt indicating that a Sync message will be generated.

Table 3.4: Docsis Cable Modem Termination System Transmission Controller Transitions.

Condition	Description
REC_DATA	Indicates the process has information that need to be enqueued.
Q_WAS_EMPTY	Indicates the queue is empty.
!Q_WAS_EMPTY	Indicates the queue is not empty.
TX_WAITING	Indicates information is waiting to be transmitted, but the transmitter is currently busy.
!TX_WAITING	Indicates information is waiting to be transmitted and it can be sent immediately, since the transmitter is idle.
TX_JUST_FREE	Indicates the transmitter just completed transmitting information and has now become idle, and there exist further information in the queue for transmission.

Chapter 4

TRAFFIC CLASSIFICATION ON HFC NETWORK

The aim of this chapter is to develop a tool that can accurately, quickly and simply classify customer traffic streams in an HFC network environment. It is expected that the result of traffic classification can be used by the CMTS for efficient bandwidth scheduling.

In a high speed data communication networks, such as HFC networks, there are two conflicting forces. On the other hand, there is a customer (network user) who wishes to view the network as a transparent entity, a pipe through which his data flows. On the other hand is the cable network provider, who wishes to maximize efficiency by connecting many users to the network as possible. The trade-off between these two goals is the QoS that the user requires. In essence, communication networking represents a game to be played by the user and the network provider, wherein each seeks to maximize his benefits [27]. The user can specify his source parameters during connection setup.

4.1 The Case for Traffic Classification

In general, the specification of the source characteristics is not an easy task. The literature is replete with methods and models for characterizing sources and their parameters which can be found in [7, 29, 13, 37, 26, 3, 2, 37, 14, 28, 30], where much work has been performed in the measurement and modeling of data, voice and video sources. Unfortunately, it may not be possible to estimate the source characteristics, especially if the source type is unknown (to the user). In this case it is the network provider's responsibility to characterize a traffic stream since a source traffic descriptor is required by the CMTS.

Note that if some methods cited above are performed off-line, then the source information may not be timely. Since the HFC networks operate at high data rates, short term source statistics may be important to those methods and thus congestion control. In general, however, statistical methods observe "longer term" trend in sources, thus potentially missing these "high frequency" source fluctuations. And of course, parametric or model-based methods do not allow the possibility of new traffic types for applications or services not yet available.

There is a need for a method that can accurately and quickly determine the traffic class of a given traffic stream, without information of its statistics, this is a non-parametric approach. For the best range of application, it should be able to be used off-line when traffic streams are available before hand, and on-line when traffic streams are too non-stationary or unknown.

The traffic classification method proposed can be thought of as a transformation operator, from the statistical domain to the traffic descriptor domain. That is, usually traffic streams are thought of in terms of their statistics, such as first and second order moments. Using the statistical tools of the above citations, various characteristics of sources can be observed, and sources can be classified.

Some of these classifications lead to familiar sources, such as geometric, Interrupted Bernoulli Process (IBP) or Markov-Modulated Bernoulli Process (MMBP), to mention just a few. However, these "statistically well known" sources are well known simply because they provide tractability in analytical methods. Unfortunately, they do not appear with any regularity in most of the "real" [27] and anticipated traffic streams.

The novel approach used in this work, however, does not rely on traditional statistical measures. It is based on the simultaneous measurement of traffic at different time scales. The data can be represented in an appropriate form, processed and organized in an array of vectors. From this array, higher order statistical measures can be derived. The descriptors (e.g. *variance-time curve, Hurst-parameter*) calculated in this way can be used to characterize the traffic, since the descriptors are different for different traffic streams.

4.2 The Classification Algorithm

The algorithm as referenced in [27] provides a method for characterizing transmissions in a packet network which may or may not be carried in real time. The method involves dividing each source into a plurality of sequential packets with each packet having address data defining an intended address, information data defining information to be transmitted and data defining source identity, like in DOCSIS 1.1 CM system.

The network also defines a train of sequential packet transport locations into which packets are loaded for transmission, such that some packet transport locations in a train contains packet and some packet transport locations are empty and such that, when there is more than one different source, the train contains packets from a different sources in sequential arrangement as determined by the network.

It also involves monitoring a train of packet transport locations to determine which packet transport locations are empty and which contains packet, and generating a series of data elements each corresponding to a respective one of the packet transport locations and each identifying whether the respective packet transport location is empty or contains a packet.

The statistical analysis is then carried out on series of data elements in real time to determine the characteristics of the transmissions. Using the data bits, the analysis includes for each register, calculating sample variance of a set of successive observations of the register contents and from the variances estimating the value of *Hurst-parameter (H)*. H is derived from the slop of the line which approximates the behavior of a plot of the logarithm of the variances of the registers versus the value of the sequential indices of the registers.

The characteristics of different traffic streams can be characterized by variance time curves. A set of variance vectors for different traffic streams can be used to train a classifier.

4.3 The DOCSIS 1.1 MAC Sublayer

In oder to understand the application of the algorithm in the DOCSIS 1.1 cable modem system, it is important to briefly re-visit the DOCSIS 1.1 MAC operation. The detail operation of DOCSIS 1.1 MAC is discussed in chapter 2.

The upstream channel in DOCSIS 1.1 is characterized by many transmitters (CMs) and one receiver (the CMTS). All the CMs listen to all frames transmitted on the downstream channel upon which they are registered and accept those where the destination match the CM itself or CPEs reached via cable-modem-to-customer-premises-equipment interface (CM-CPI) port.

Time in the upstream channel is slotted, providing for *Time Division Multiple Access (TDMA)* at regulated time ticks. The CMTS provide the time reference and controls the allowed usage of each interval. Intervals may be granted for transmissions by particular CM, or for contention by all CMs.

The CMTS must generate the time reference for identifying these slots. It must also control access to these slots by the CMs. For example, the CMTS may grant some number of contiguous slots to a CM for it to transmit a data PDU. The CM must then time its transmission so that the CMTS receive it in the reference specified.

4.3.1 Bandwidth Allocation (MAP)

Figure 4.1 illustrate the bandwidth allocation MAP. It consists of minislots, each minislot is numbered relative to a master reference by the CMTS. The clocking information is distributed to the CMs by means of $SYNC$ packets.

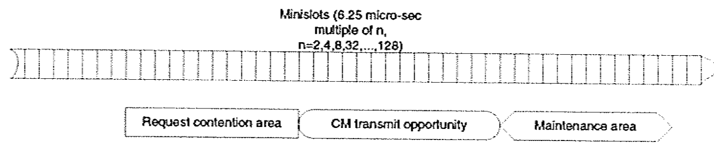

Figure 4.1: Bandwidth Allocation MAP.

The allocation Map is a MAC management message transmitted by the CMTS on the downstream channel which describes, for some interval, the uses to which the upstream minislots must be put. A given Map may describe some slots as grants for a particular station to transmit data in (*CM_tx_opportunity*), other slots are available for contention transmission (*Request_contention_area*), and other slots as an opportunity for new station to join the link (*maintenance*).

4.4 Application of the Algorithm in DOCSIS 1.1

Since DOCSIS 1.1 protocol involves granting the timely minislots, within the MAC management message to the CMs by the CMTS as an opportunity to transmit data packets, the CMTS can monitor the packet arrival information within the minislots allocated to the to the CMs.

Using the packet arrival information within the minislot allocation, the CMTS can use the above traffic classification algorithm in real-time to predict the pattern of the traffic streams generated upstream.

4.5 Classifier Architecture

The classification can be done using two approaches, the first is using the *Hurst-parameter* and the second method is by using a neural network.

4.5.1 Classification Using Hurst-parameter

From the set of variance vectors, the *Hurst-parameter* can be calculated. The *time-variance* curve is a plot of the logarithm of the variances against the sequential indices of the registers [33, 32]. The *Hurst-parameter* is given by ($H = 1 - (\frac{\beta}{2})$), where β is the slop of the *time-variance* curve.

4.5.2 Classification Using Neural Network

The classification in this problem can also be considered as a pattern recognition. Each type of traffic stream is expected to have different pattern. Since neural networks are excellent pattern recognizers [27], they can be used to learn the traffic patterns (descriptors) and then classify the traffic streams. Traffic from known sources can be used to train the neural network. A set of variance vectors can be used to train the neural network. A faster training method is to be adopted to avoid unnecessary training delay.

Once trained, a neural network can classify the traffic quickly and efficiently in real time, and can classify even more complicated traffic sources. The feed-forward network architecture [31] with back-propagation algorithm was used.

Chapter 5

SIMULATIONS

The model used for simulations was the *Common Simulation Framework v.13 (CSF13)*. The CSF is a baseline model, based on the OPNET simulation package. Suitable modifications were applied to the model in order to generate the the required traffic sources and optimize the model for simulating networks with up to 800 nodes.

The parameters that were used for the simulation runs are given in table 5.1. The values selected for the simulation parameters are typical values used in actual implementation and/or default values proposed in the DOCSIS 1.1 specification. A modified implementation of the traffic sources provided by the CSF, was used to generate the traffic streams.

The sources create packets of constant and varying length between simulation runs from 64 to 1000 bytes, with a constant and Poisson's interarrival rate. The time, t, that each source starts generating packets was exponentially distributed with a mean of $2s$. This prevents CMs from issuing the first request simultaneously and causing excessive collisions at the beginning of the simulation.

Table 5.1: Simulation Parameters.

Parameter	value
Upstream Channel Capacity	$5.12Mbps$
Downstream Channel Capacity (64QAM channel including $MPEG2 - TS$ overhead)	$26.97Mbps$
Number of CMs	$2 - 700$
Minislot Size	$32Bytes$
Maximum number of minislots in MAP	4096
Maximum number of Information Elements (IEs) in MAP	240
Data Back-off start	4
Data Back-off end	8
Offered load	$0 - 12Mbps$

In all the tests performed, only the traffic streams are assumed present in the upstream channel with no background load. The offered load generated by the traffic streams is the load that arrives to the CM from the Customer Premises Equipment (CPE) interface. The most popular CPE interface in DOCSIS 1.1 CMs is a $10Mbps$.Therefore a stream would include the Ethernet MAC interface plus higher protocol overheads. Therefore not all of the stream capacity is available to the traffic applications. Such overhead should be taken into account when considering the delivery of certain applications with specific bandwidth requirements.

The effective throughput reference in [35, 36] for the different streams as a function of the packet size and protocol stack used is given in table 5.2. It is evident that al-

Table 5.2: Effective Bandwidth per Stream with Different Packet Sizes.

Effective bandwidth in Kbps Packet Size (bytes)					
Stream		64	128	256	512
8Kbps	UDP/IP	2.75Kbps	5.38Kbps	6.69Kbps	7.34Kbps
	TCP/IP	1.25Kbps	4.63Kbps	6.31Kbps	7.16Kbps
16Kbps	UDP/IP	5.50Kbps	10.75Kbps	13.38Kbps	14.69Kbps
	TCP/IP	2.50Kbps	9.25Kbps	12.63Kbps	14.31Kbps
32Kbps	UDP/IP	11.00Kbps	21.50Kbps	26.75Kbps	29.38Kbps
	TCP/IP	5.00Kbps	18.50Kbps	25.25Kbps	28.63Kbps
64Kbps	UDP/IP	22.00Kbps	43.00Kbps	53.50Kbps	58.75Kbps
	TCP/IP	10.00Kbps	37.00Kbps	50.50Kbps	57.25Kbps

though the lower packet sizes provide faster interaction, the available bandwidth to the application is significantly reduced. Several simulation scenarios were considered in order to address the following issues:

- The capacity of the upstream channel for different traffic streams.
- The effects of packet size in terms of mean delay and system throughput.
- The effects of data compression and classification on throughput and the modem population.

5.1 Classified and Unclassified Traffic Analysis

The objective of this simulation scenario is to investigate the effect of traffic classification on the DOCSIS HFC network protocol. In this scenario the network model that has been used consists of 200 CMs generating bursty traffic with Poisson's distribution. In

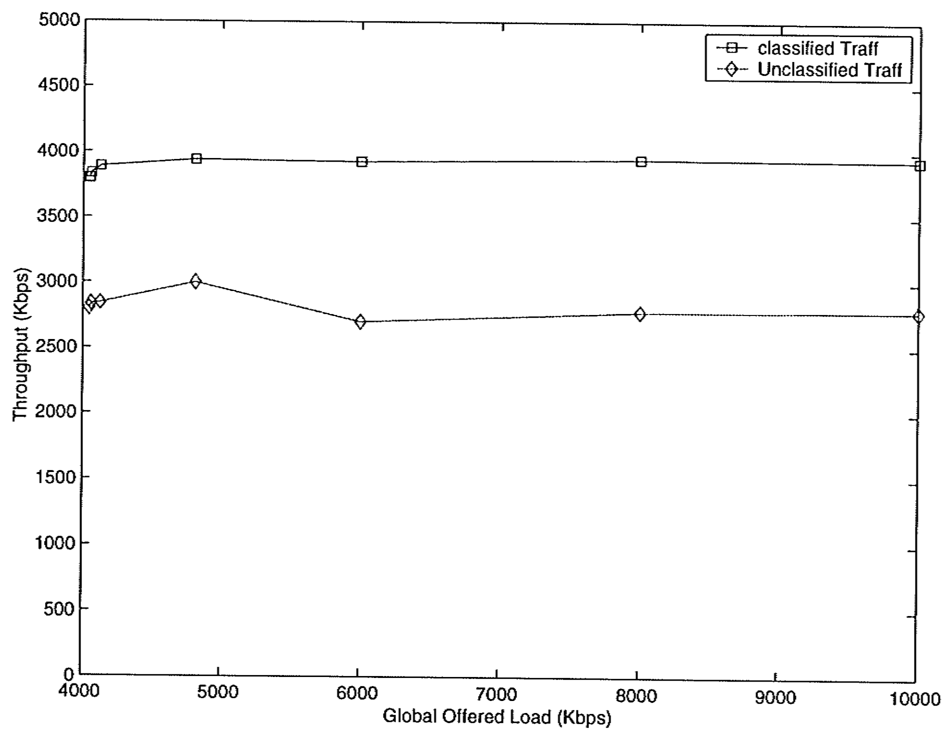

Figure 5.1: Maximum Throughput for Classified and Unclassified Traffic vs Global Offered Load.

the fist case the traffic is classified by the CMTS classifier while invoking the bandwidth allocation mechanism, while in the second case the traffic is unclassified.

Figure 5.1 show the system throughput versus global offered load. It can be seen that, with classified traffic the overall network throughput is as high as $4000Kbps$. This gives the network utilization of 80%. While for unclassified traffic the maximum attainable network throughput is only $3000Kbps$. This gives the network utilization of 60%.

The difference in throughput is due to the fact that for classified case the bandwidth is allocated on priority base with regard to the traffic QoS requirement (delay, jitter, throughput etc). In the case of unclassified traffic each stream get a share of bandwidth allocation from the CMTS with the same priority, without regard to the traffic QoS requirement. This cause excessive collision on the network, and result in a bandwidth

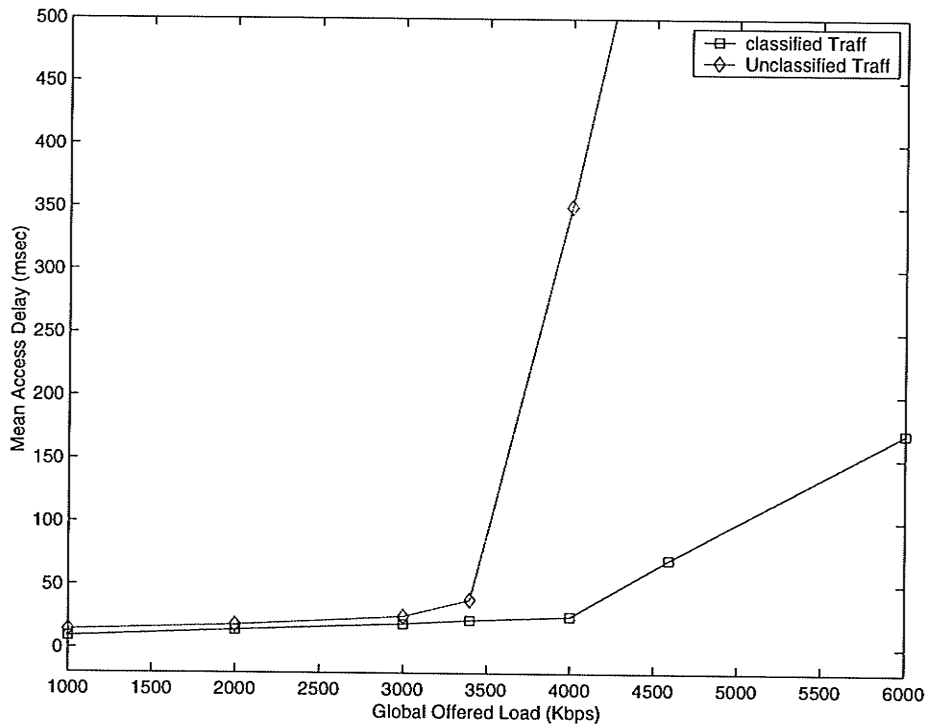

Figure 5.2: Mean Access Delay for Classified and Unclassified Traffic vs Global Offered Load.

wastage and a long delay in forwarding a packet from the CM to the CMTS.

Figure 5.2 represent the curves for the network access delay for the two cases versus global offered load, for the same reason mentioned earlier, it can be seen that, the network access delay for unclassified traffic case is between $12-400ms$ up to offered load of $4200Kbps$, and that for classified traffic case is between $8-300ms$. Beyond $4200Kbps$ the network access delay for the unclassified case assymtotically increased. That for classified case linearly increased to $1600ms$ at $6000Kbps$ offered load.

5.2 Analysis of VBR Traffic with P-FCFS

In this simulation scenario the network has the same condition as in section 5.1, except that in this case the traffic is classified in to eight classes with varying QoS requirement. The CMTS bandwidth scheduling algorithm was designed to give preferential treatment to higher priority traffic in the MAC queue.

The *class-0* traffic being the highest priority receive the highest probability of bandwidth allocation by the CMTS and is dequeued earlier in the MAC queue. The next priority level is *class-1* then followed by *class-2* then *class-3* through *class-7*. The *class-7* is the least priority traffic and is the last traffic to be dequeued from the MAC queue, it also has the least share of the bandwidth allocation at saturation.

Figure 5.3 represent the network throughput for all the eight traffic classes versus global offered load. The throughput per class is linear with respect to offered load until the load reaches $4200Kbps$, which gives $495Kbps$ throughput per class, or a system throughput of $(495x8) = 3960Kbps$. The maximum system throughput is $4000Kbps$ (from figure 5.1) and so as the offered load exceeds $4200Kbps$ the lower priority services (*class-7* at first, followed by *class-6*, etc.) receive decreasing throughput. It can also be seen that, the throughput of *class-0* through *class-3* linearly increased even beyond offered load of $10,000Kbps$. This means that these classes can be suitable for traffic that require high

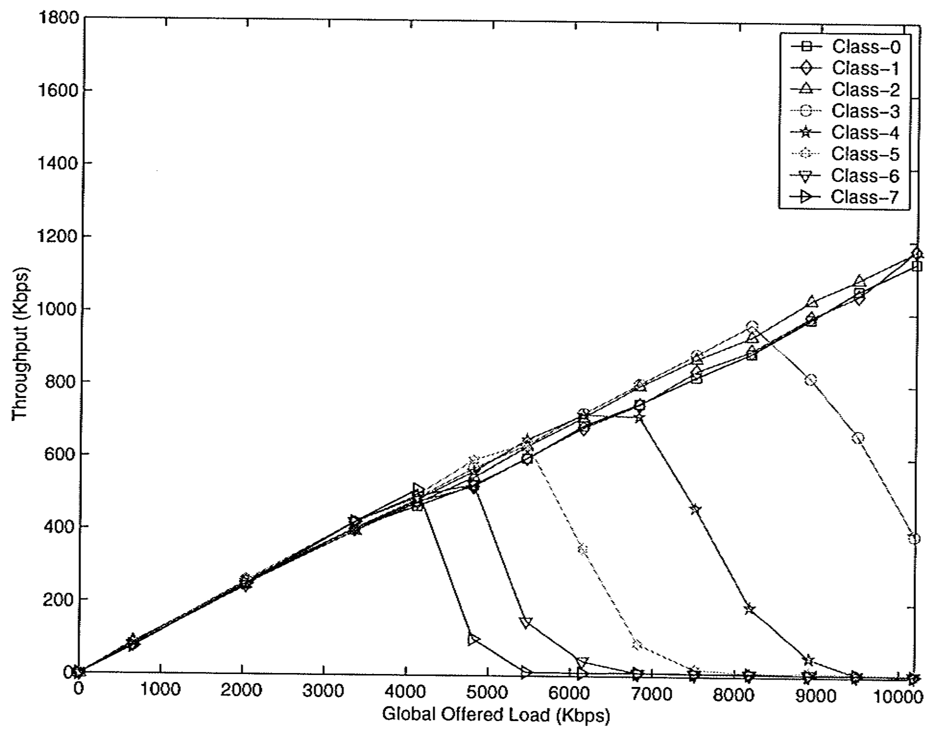

Figure 5.3: Throughput for the Classified Traffic vs Global Offered Load.

bandwidth.

The Prioritized First Come First Served (P-FCFS) CMTS load scheduling algorithm means that at saturation the low priority services have their bandwidth allocation reduced. The associated mean access delay (expressed as the mean time between the packets arriving at the CM's MAC and it reaching the CMTS) is shown in figure 5.4.

For the offered loads below $4200Kbps$ the mean access delay varies between $8 - 400ms$ for *class-0*, and $10 - 500ms$ for *class-7*. With excess offered load the mean access delay becomes around $1000ms$ for *class-0* through *2* and assymtotically large for *class-3* through *7*.

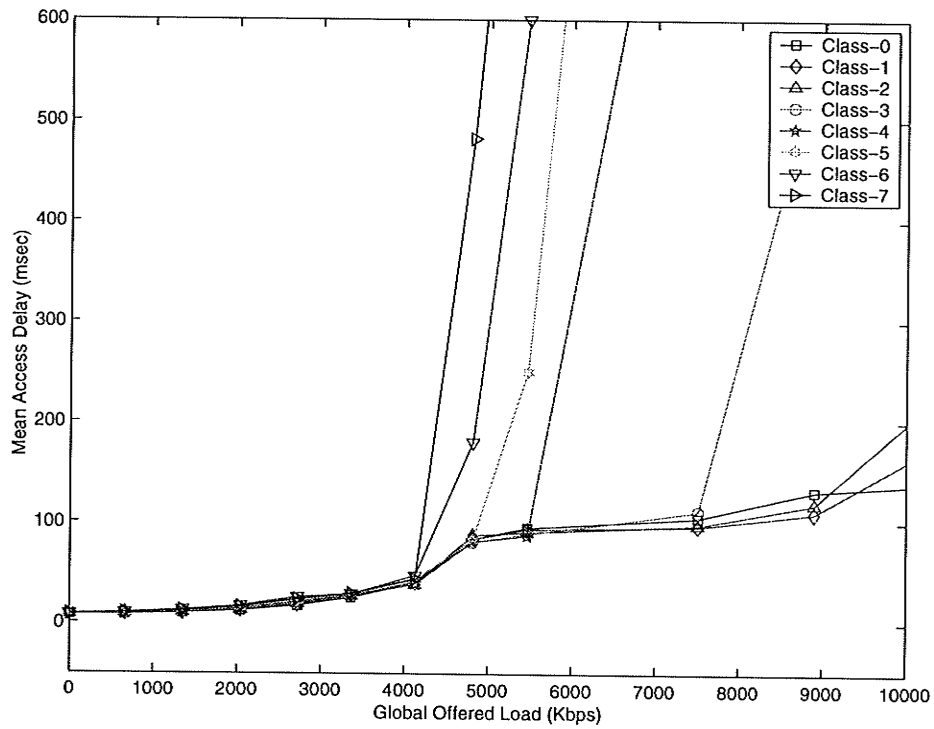

Figure 5.4: Mean Access Delay for Classified Traffic vs Global Offered Load.

When the offered load exceeds the channel capacity the transmission period described by the MAP reaches its maximum size $4096 minislots$. As the number of requests for the higher priority classes increases then the allocations for the lower priorities are reduced, lowest priority first, until it becomes starved of service, at which point the next lower priority receives a reduced service support. Eventually at the end only the highest priority class will have service.

The variation in offered load was controlled by varying the length of the message and the number of active CMs (each CM attempts to send messages of the same length). The maximum throughput is $1000 Kbps$ lower than the channel capacity due to protocol overhead (from medium access control and physical layer). At saturation, the order in which the CMTS allocates transmission requests in the MAP becomes significant, once the MAP is full no further requests can be serviced and so unserviced requests are termed 'pending' i.e. waiting for the next MAP allocation. When saturation is first reached this means that the last request for the lowest priority traffic will be unsuccessful and as the offered load increases the number of unsuccessful requests increases for the lowest priority until none can be serviced. At that point the next lower priority starts to receive a starved service. In most extreme cases only some of the highest priority requests will receive i.e., if the offered load from *class-0* CMs exceeds $4000Kbps$ then some of those CMs will also suffer service starvation.

5.3 Constant Bit Rate Traffic Streams (CBR)

In all these tests only the CBR traffic streams are assumed present in the upstream channel with no background load. The offered load generated by the CBR traffic is the load that arrives to the CM from the Customer Premises Equipment (CPE) Interface via the most popular CPE interface in DOCSIS 1.1 CMs $10Mbps$ Ethernet link.

5.3.1 Channel Capacity

This test addressed the worst case scenario of low bit rate CBR streams in which all the transmitted packets are were minimum length Ethernet frames. Since the concatenation feature is not enabled, the CMs were forced to issue requests on a per packet basis either via the contention minislots or the piggyback mechanism. Eventually as the load increased all requests were piggybacked. Figure 5.5 represent the mean packet delay as a function of active CMs. The system saturation point is a number of CMs beyond which the delay increases sharply. After this point the network is unable to provide a finite packet delay and thus the CBR services fail. For the $128Kbps$ CBR stream case the saturation point is as low as 32 CMs i.e. 80% of the maximum theoretical.

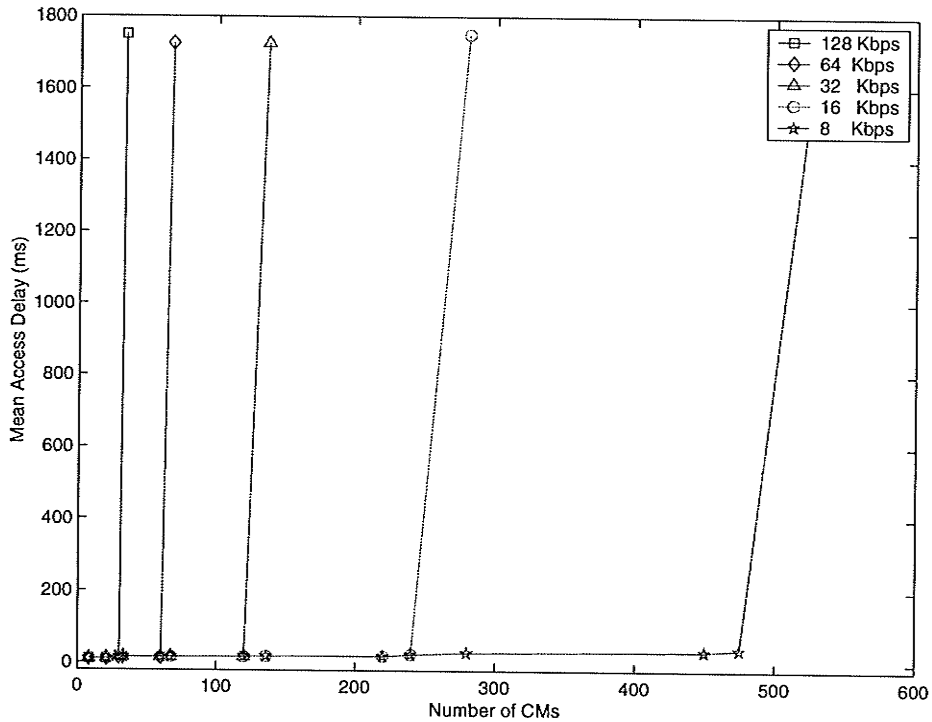

Figure 5.5: Mean Access vs Number of Stations.

It should be noted that the threshold value of 32 CMs is the number of simultaneously active CMs. Certain CBR applications such as telephony with silence suppression, have moments of inactivity during a session and so the number of supported active sessions might be higher than suggested.

The mean delay as a function of throughput is given in figure 5.6. For reasonably low packet delay ($< 20ms$) the corresponding system throughput is was less than $3650Kbps$ i.e. 73% of the total upstream capacity. The maximum throughput achieved is shown more clearly in figure 5.7. In all the cases the system throughput increases almost linearly with respect to the cable modem population up to the point of saturation. The increase of the number of CMs beyond this point has marginal effect. For all the cases the total

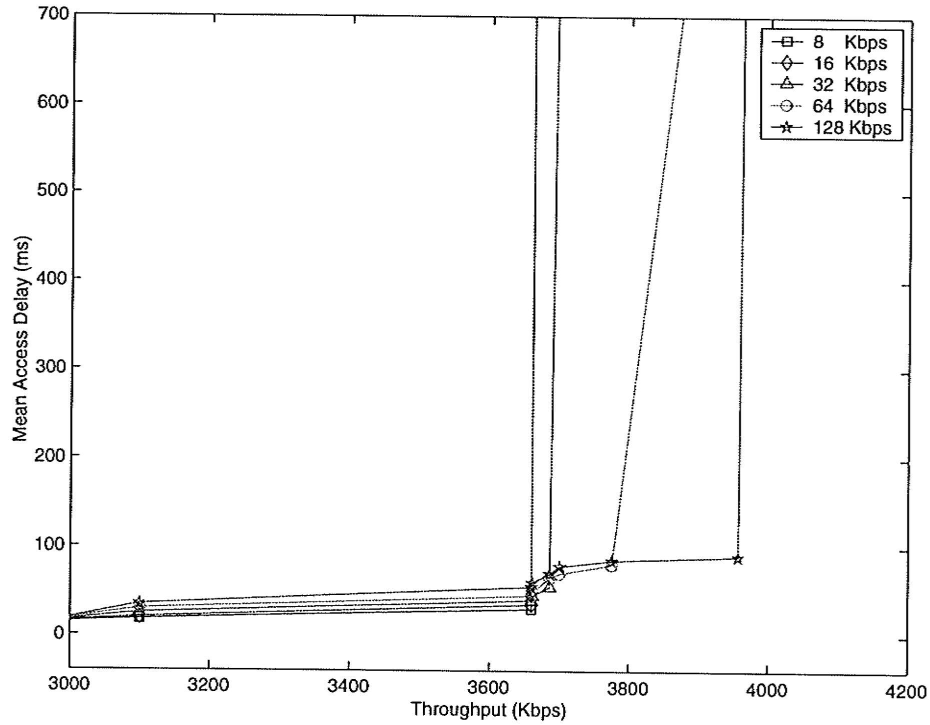

Figure 5.6: Mean Access Delay vs Throughput.

system throughput is between $3750Kbps$ and $4200Kbps$. This constitutes 75% and 84% of the total capacity.

5.3.2 Effect of Packet Size on CBR traffic

Here the variation of packet size on system performance is examined. The results from the 8 and $64Kbps$ streams are presented in figure 5.8 through 5.11. In both case the increase in packet size results in an increase in the number of streams supported and the maximum system throughput.

For the mean delay of the $8Kbps$ stream in figure 5.8, it is apparent that when the packet size is $512bytes$ even with CM population of 475 the upstream channel is not saturated. However with $64byte$ packet system saturation is evident past the 375 CMs

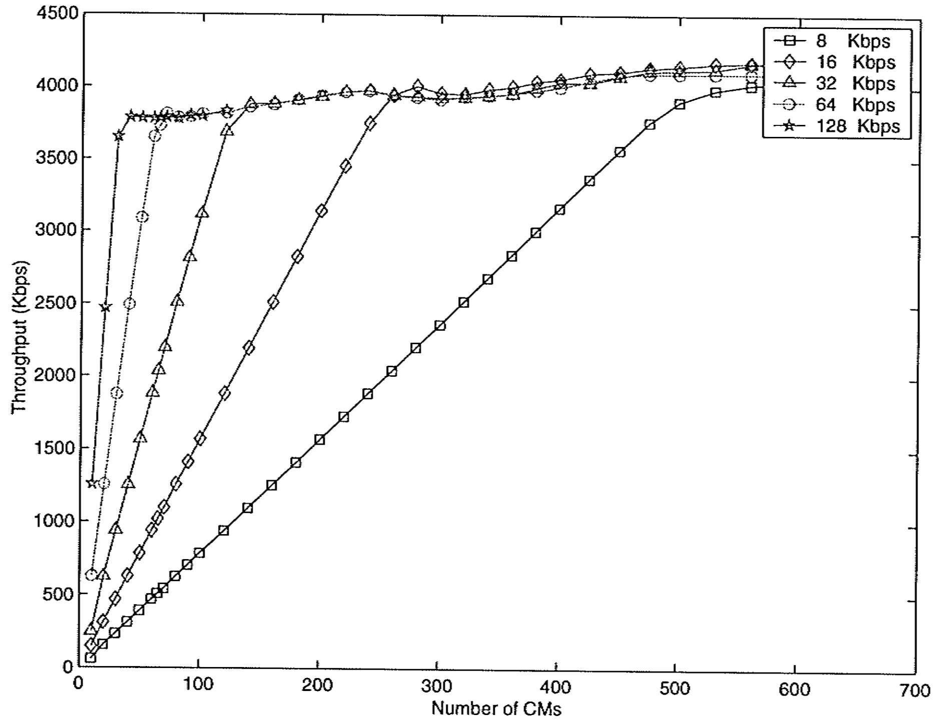

Figure 5.7: Throughput vs Number of Stations.

mark. A similar behavior is noticed in the $64Kbps$ case in figure 5.10, up to 51 and 55 streams are supported for 64 and $512byte$ packets respectively. In terms of system throughput, $64byte$ packets yield slightly over $3920Kbps$, while $512bytes$ packets increases this to $3960Kbps$ which corresponds to $40Kbps$ channel capacity increase (figure 5.11). Results for $8Kbps$ streams are similar, as seen from figure 5.9. The $64byte$ packets yield $3920Kbps$ while $512byte$ packets yield $4040Kbps$. The intermediate packet sizes, 128 and 256 bytes also offer significant advantages when compared to the $64byte$ case.

5.4 Discussion

The result presented indicate that CBR streams can be supported by DOCSIS 1.1 CATV network, even with limited resource allocation capabilities of the protocol. There are how-

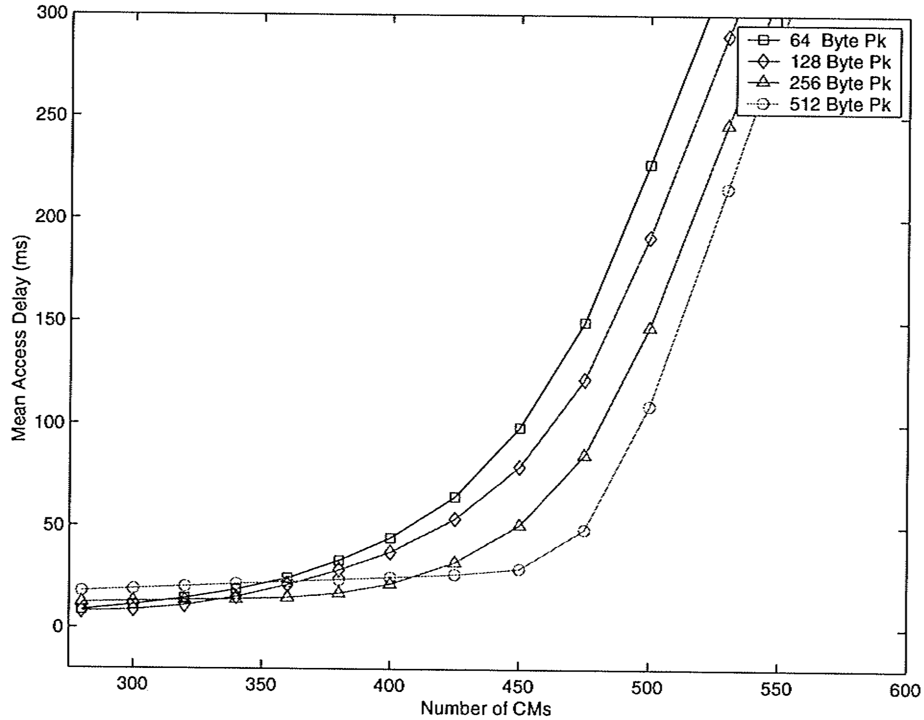

Figure 5.8: Mean Access Delay for 8 Kbps Stream vs Number of Stations.

ever issues that need to be considered. The general trend from the result was that the system would comfortably support a number of active streams, given a small increase in packet delay ($< 20ms$) as user population increases, up to threshold point. After that point even slight increase in the number of CMs results in system instability. Performance deterioration is not gradual and mean packet delay increases rapidly after the threshold point. In the case of CBR streams few extra CMs would mean difference between acceptable performance and unusable service.

Limiting the number of active CMs per upstream channel to the threshold number indicated by simulations, would be satisfactory solution in case where a single channel is dedicated to a certain type of service such as IP telephony or digital audio broadcasting by radio stations. Alternatively the operator needs to be able to predict accurately the

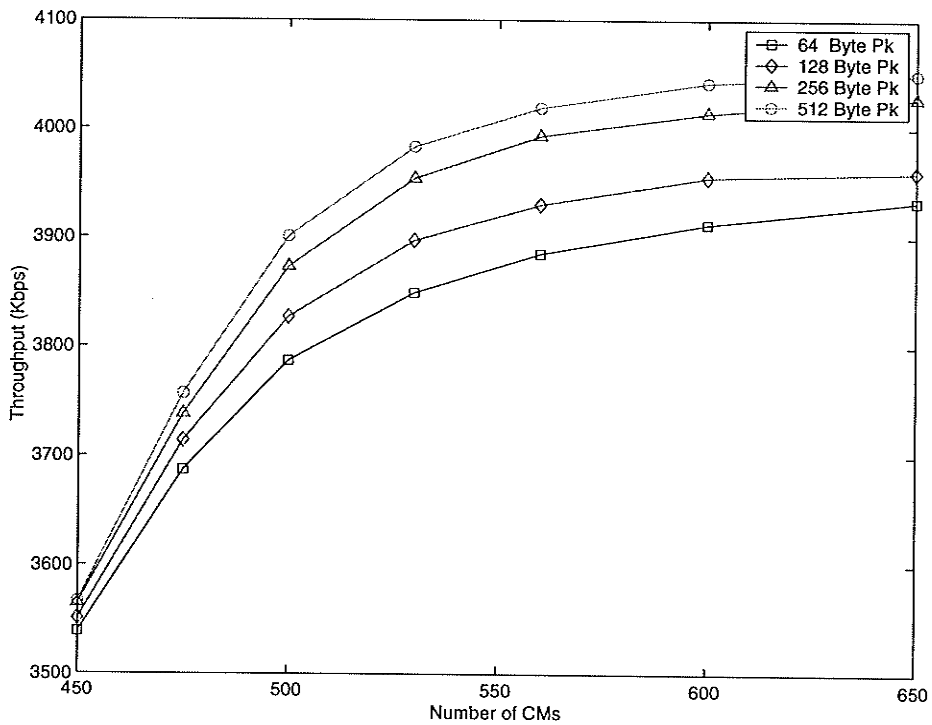

Figure 5.9: Throughput for 8 Kbps Stream vs Number of Stations.

type of streams any one subscriber would initiate at any one time. Accurate traffic prediction is difficult and so a more flexible blocking scheduler with the ability to reserve bandwidth on demand is needed.

The use of Committed Information Rate (CIR) could provide an intermediate solution but it has limitations [36]. By definition the CMTS should not allow CMs to join the network, if the total CIR requirements of the new CM and of the ones already registered exceeds the channel capacity. However results in this thesis indicated that the effective channel capacity depends on the type of streams delivered and their characteristics and can be as low as 60% of the nominal upstream bandwidth. Therefore, the number and type of CBR streams that can be supported is not constant. The scheduler must be able to monitor the metrics of streams already served and block requests for new

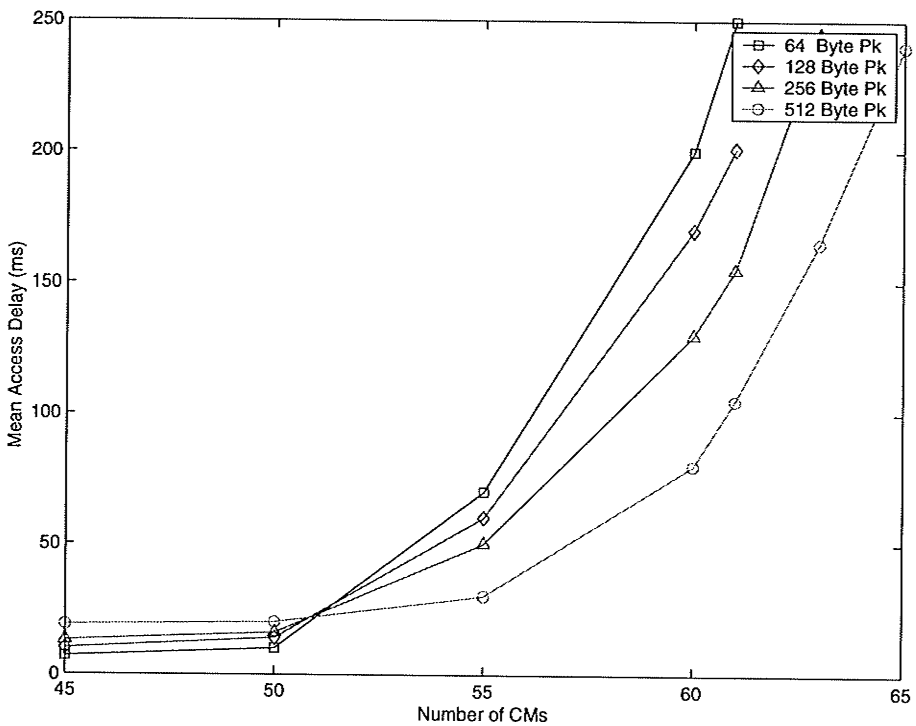

Figure 5.10: Mean Access Delay for 64 Kbps Stream vs Number of Stations.

streams that would cause these metrics to exceed critical values. Possible metrics would would be the delay, the jitter and the throughput that each CM is allocated per unit time.

There are issues of the inability of the system to utilize the rest of the 16 – 20% of the available bandwidth. The first reason is due to MAC and PHY protocol and encapsulation overheads. The second reason for the loss is attributed to the bandwidth allocation algorithm, the collision resolution and the unused contention minislots described in each MAP. More specifically a CM can not transmit a packet before the request for bandwidth, either explicit or piggybacked, reaches the CMTS, the request is processed, a grant is sent via MAP and finally the allocated minislot time arrives. The result of this mechanism is idle time in the upstream channel.

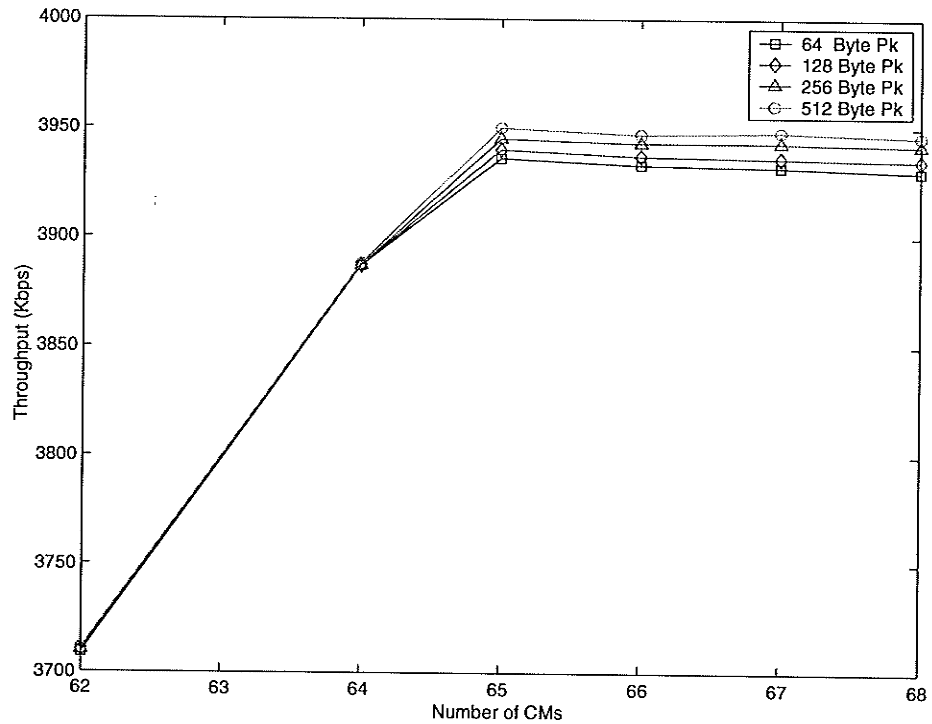

Figure 5.11: Throughput for 64 Kbps Stream vs Number of Stations.

The percentage of the idle time is reduced as the CMs population increases. The reason is that during the time one CM is idle another is making use of the channel. However higher numbers of CMs result in more collisions forcing CMs to backoff and not make use of the channel. Under heavy load both of these effects, collisions and idle time while waiting for a grant, are diminished. At very high loads all CMs are piggybacking their request, avoiding collisions, while all the upstream minislots are reserved, and therefore used, by the CM population. In this case utilization loss is only incurred by the unused contention minislots.

5.5 Traffic Characterization

Figure 5.12 shows the traffic classification using time variance time curves. The three signals represent FTP(top), Video(center) and Audio(bottom). The $Hurst - parameter$ was calculated from their respective time variance curves as follows.

$H = 1 - (\frac{\beta}{2})$

Where β is the slop of time-variance curve. It can be seen that the steeper the slop of the variance-time curve the smaller the value of H. It is interesting to see that, the three traffic sources are clearly characterized by their $Hurst - prameter$. The variances were calculated in real time from the respective packet inter-arrival information from which the $Hurst - parameter$ is calculated.

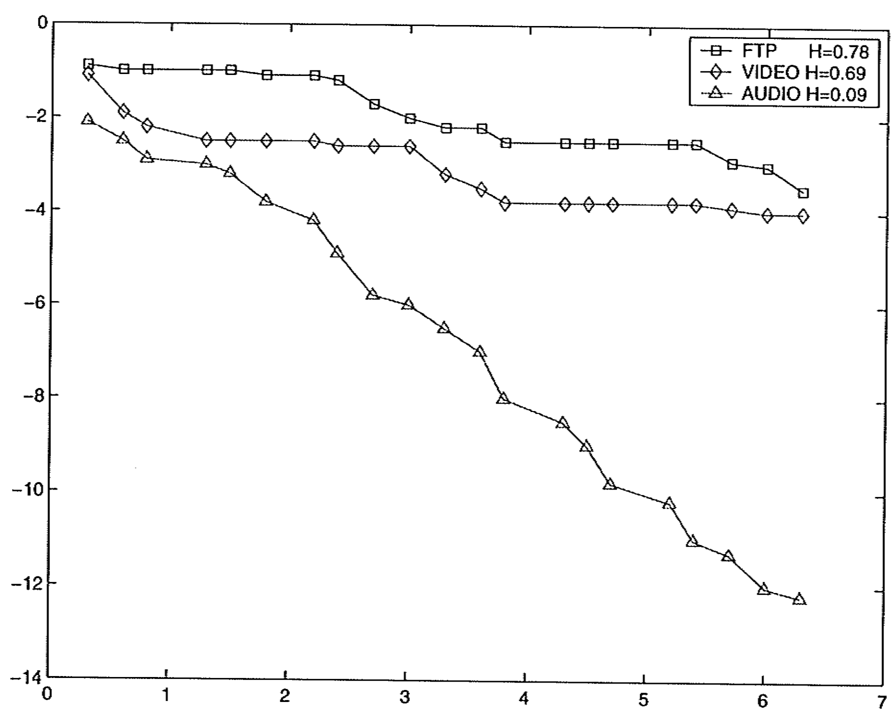

Figure 5.12: Traffic Characterization Using Hurst-parameter.

5.6 Neural Network Classifier

Another way to efficiently classify the traffic sources is to use a neural network approach. This is because classification problem of this nature can be considered as a pattern recognition. Since neural networks are excellent pattern recognizers, they can be used to learn the traffic patterns (descriptors) and then classify the traffic.

A set of variance vectors of the captured traffic in the previous section was also used to train a neural network. Two layer network was used, with *tansigmoid* transfer function in the hidden layer and *logsig* transfer function in the output layer. Ten neurons were used in the hidden layer. The network also has three output neurons, since there are three targets (voice, video and data).

The data as seen in figure 5.13 was classified in only 17 ephocs with an accuracy of 10^{-10}. The *MATLAB Neural Network Toolbox* simulator was used for this simulation. The *"Levenberg-Marquardt"* algorithm was used in training the network. This is because the algorithm is faster in training feed forward networks [10].

The following is the set of variance vector p and the target vector t from the *MATLAB* command window that were used for the simulation. It can also be seen that after training, the network was able to classify the three traffic sources in only 17 ephocs with accuracy of 10^{-10}.

```
>> data


p =
```

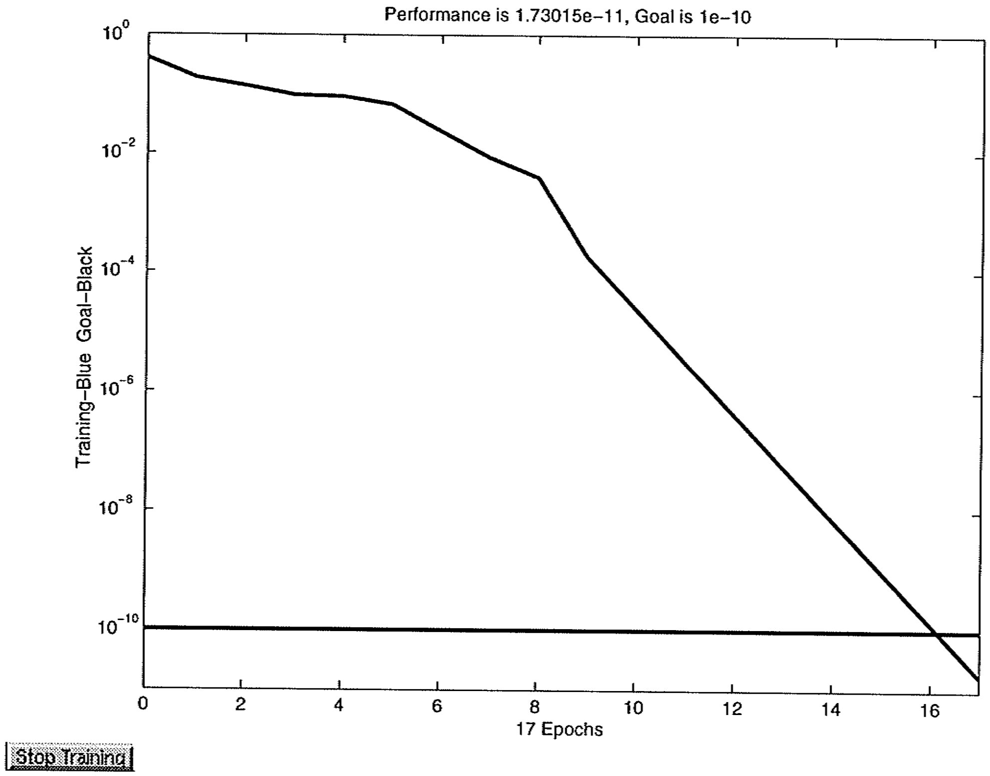

Figure 5.13: Classification Using Neural Networks.

Columns 1 through 7

0.3000	0.6000	0.8000	1.3000	1.5000	1.8000	2.2000
-0.9000	-1.0000	-1.0000	-1.0000	-1.0000	-1.1000	-1.1000

Columns 8 through 14

2.4000	2.7000	3.0000	3.3000	3.6000	3.8000	4.3000
-1.2000	-1.7000	-2.0000	-2.2000	-2.2000	-2.5000	-2.5000

Columns 15 through 21

4.5000	4.7000	5.2000	5.4000	5.7000	6.0000	6.3000
-2.5000	-2.5000	-2.5000	-2.5000	-2.9000	-3.0000	-3.5000

Columns 22 through 28

0.3000	0.6000	0.8000	1.3000	1.5000	1.8000	2.2000
-1.1000	-1.9000	-2.2000	-2.5000	-2.5000	-2.5000	-2.5000

Columns 29 through 35

2.4000	2.7000	3.0000	3.3000	3.6000	3.8000	4.3000
-2.6000	-2.6000	-2.6000	-3.2000	-3.5000	-3.8000	-3.8000

Columns 36 through 42

4.5000	4.7000	5.2000	5.4000	5.7000	6.0000	6.3000
-3.8000	-3.8000	-3.8000	-3.8000	-3.9000	-4.0000	-4.0000

Columns 43 through 49

0.3000	0.6000	0.8000	1.3000	1.5000	1.8000	2.2000
-2.1000	-2.5000	-2.9000	-3.0000	-3.2000	-3.8000	-4.2000

Columns 50 through 56

```
    2.4000    2.7000    3.0000    3.3000    3.6000    3.8000    4.3000
   -4.9000   -5.8000   -6.0000   -6.5000   -7.0000   -8.0000   -8.5000

  Columns 57 through 63

    4.5000    4.7000    5.2000    5.4000    5.7000    6.0000    6.3000
   -9.0000   -9.8000  -10.2000  -11.0000  -11.3000  -12.0000  -12.2000


t =

  Columns 1 through 11

     1     1     1     1     1     1     1     1     1     1     1
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0

  Columns 12 through 22

     1     1     1     1     1     1     1     1     1     1     0
     0     0     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     0     0     0     0     0

  Columns 23 through 33

     0     0     0     0     0     0     0     0     0     0     0
```

```
  1     1     1     1     1     1     1     1     1     1     1
  0     0     0     0     0     0     0     0     0     0     0

 Columns 34 through 44

  0     0     0     0     0     0     0     0     0     0     0
  1     1     1     1     1     1     1     1     1     0     0
  0     0     0     0     0     0     0     0     0     1     1

 Columns 45 through 55

  0     0     0     0     0     0     0     0     0     0     0
  0     0     0     0     0     0     0     0     0     0     0
  1     1     1     1     1     1     1     1     1     1     1

 Columns 56 through 63

  0     0     0     0     0     0     0     0
  0     0     0     0     0     0     0     0
  1     1     1     1     1     1     1     1

>> net = newff(minmax(p), [10, 3], {'tansig','logsig'}, 'trainlm');
>> net.trainParam.show  = 5;
>> net.trainParam.ephochs=50;
>> net.trainParam.goal   =1e-10;
>> [net, tr] = train(net, p, t);
```

```
TRAINLM, Epoch 0/100, MSE 0.364237/1e-10, Gradient 19.6129/1e-10

TRAINLM, Epoch 5/100, MSE 0.0645928/1e-10, Gradient 12.5053/1e-10

TRAINLM, Epoch 10/100, MSE 0.000184764/1e-10, Gradient 0.146557/1e-10

TRAINLM, Epoch 15/100, MSE 5.33172e-09/1e-10, Gradient 4.43058e-06/1e-10

TRAINLM, Epoch 17/100, MSE 9.44799e-11/1e-10, Gradient 7.89496e-08/1e-10

TRAINLM, Performance goal met.


>>
```

Chapter 6

CONCLUSIONS

The Multimedia Cable Network Systems (MCNS) Data Over Cable Service Interface Specifications (DOCSIS) for the provision of bi-directional broadband digital data transfer have been widely adopted by manufacturers of Community Antenna Television (CATV) data networking systems.

The specifications define the Medium Access Control (MAC) and physical layer interfaces and protocols for the transfer of data upstream and downstream between subscriber's cable modem and the service provider's data headend.

The downstream capacity was set as $26.97 Mbps$ and the upstream channel capacity as $5.12 Mbps$. The MAC protocol operates on a contention based reservation system with the CMTS confirming, or otherwise, bandwidth allocation via a MAC management message (MAP). The streams generated by the cable modems are assigned service priority (eight levels) and so the service provision is determined with respect to this priority. The quality of service is provided by the scheduling algorithms implemented in the CMTS.

The simulation results show that for an upstream channel capacity of $5.12Mbps$ the maximum throughput cannot exceed $3000Kbps$ without packet classification, and with classification it can be as high as $4000Kbps$. The difference between the maximum throughput and the channel capacity is caused by the MAC and physical layer packet overheads, unused capacity and the multiple access scheme's MAP structure.

The CBR streams would be used to provide compressed audio, video and voice applications. The performance was analyzed in terms of number of supported subscribers and packet length used for the delivery of the services. The results showed that the number of streams that can be supported is almost proportional to the stream intensity,480(625), 250(312), 125(156), 62(78) and 31(40) streams of 8, 16, 32, 64, and $128Kbps$ respectively. This number was found to be substantially lower than the theoretical maximum due to collisions, protocol overheads the lagging effect of the transmission cycle.

It was shown that, with existing HFC MAC protocol and CMTS load scheduling algorithm, it is possible to improve the capacity utilization to as high as 84% by optimizing the upstream traffic.

Variation in packet size was found to have a significant affect on the system throughput. Mean packet delay in all cases was increased insignificantly in respect to modem population, up to a threshold point which is effectively the number of streams that the system can support as mentioned earlier. Beyond this point the mean delay increases assymtotically and the system becomes unstable. Therefore, when planning for the delivery of interactive and delay sensitive applications the number of stations allowed to simultaneously participate should not exceed the supported limit.

It was also shown that packet classification has significant impact on the overall network performance. It was also shown that the CMTS can classify the upstream traffic by monitoring the packet interarrival time information from the traffic sources, and thereby calculating the variance vectors. The classification can be carried out by calculating the *Hurst-parameter* from the variance vectors or using a neural network approach.

APPENDIX

VBR TRAFFIC MODEL SOURCE CODE

```
##########################################################################
##########################################################################
                  General Process Description:


This process model "catv_vbr_src" represents the root process for
vbr source. This process spawns a child process "catv_src_type1"
based on the number of sources of this type specified after a delay
that represents the starting time ofthis source.


                  Attribute: arrival_rate


This attribute specifies the arrival rate in msg/sec of this source
type. The size of this message is selected from the distribution
specified by the user. The default specification is "catv_src1_pk-
size_pdf" which represents a batch poisson process generating
bursty traffic.


                  Attribute: pksize_pdf


This attribute specifies the distribution from which message sizes
for this source are selected. The default specification "catv_src1
_pksize_pdf" represents a batch poisson process generating bursty
traffic.
```

Header Block

```
#include        "csf.h"

int global_src1_count = 0;

#define SRC_PRO_TYPE1 "docsis_vbr_src"
```

Temporary Variable Block

```
int i;

int num_sources;

Prohandle child_process;

double start_time;
```

Enter Execs for the forced state "create_child"

```
/* Determine the number of sources this module represents. */

op_ima_obj_attr_get (op_id_self (), "number_of_sources", &num_sources);


/* Now create an instance of the child process model */

/* for each of these sources and invoke it. */


for (i = 0; i < num_sources; i++)

{

child_process = op_pro_create (SRC_PRO_TYPE1, OPC_NIL);

op_pro_invoke (child_process, OPC_NIL);
```

```
}

          Enter Execs for the unforced state "Start"

int sim_active_nodes;

/* Get the start time for this generator and schedule a self inter
rupt */
/* to start at that time. */

op_ima_sim_attr_get (OPC_IMA_INTEGER, "data_nodes_active", &sim_active_nodes);

if (global_src1_count++ < sim_active_nodes)
{
op_ima_sim_attr_get (OPC_IMA_DOUBLE,
"gen_start_time_type1", &start_time);
op_intrpt_schedule_self (start_time, 0);
}
```

CBR TRAFFIC MODEL SOURCE CODE

```
################################################################################
                      Process Model Report:  docsis_cbr_src
################################################################################
                   General Process Description:

This process model "catv_src_root3" represents the root process for the
cbr source. This process spawns a child process "catv_src_type3" based on
the number of sources of this type specified after a delay that represents
the starting time of this source.

                   Attribute: bit_rate

This attribute specifies the bit rate of the cbr Source in Kbits/sec.
Source. It generates 48 byte messages at a constant rate.

                   Attribute: segment_size

This attribute specifies the size of messages generated by this traffic
source. The default value is set to 48 bytes.

                   Header Block

#include "csf.h"
```

```
#define SRC_PRO_TYPE3 "docsis_cbr_src"
```

Temporary Variable Block

```
int i;
int num_sources;
Prohandle child_process;
double start_time;
```

Enter Execs for the forced state "create_child"

```
/* Determine the number of sources this module represents. */
op_ima_obj_attr_get (op_id_self (), "number_of_sources", &num_sources);

/* Now create an instance of the child process model */
/* for each of these sources and invoke it. */
for (i = 0; i < num_sources; i++)
{
child_process = op_pro_create (SRC_PRO_TYPE3, OPC_NIL);
op_pro_invoke (child_process, OPC_NIL);
}
```

Enter Execs for the unforced state "Start"

```
/* Get the start time for this generator and schedule a self interrupt */
/* to start at that time. */
```

```
op_ima_sim_attr_get (OPC_IMA_DOUBLE, "gen_start_time_type3", &start_time);
op_intrpt_schedule_self (start_time, 0);
```

Bibliography

[1] ETSI 300 800. Digital Video Brodcasting Group (DVB) ; Interaction Channel foe Cable TV Distribution System (CAT)v. 1998.

[2] R. Addie and M. Frater. Loss forecasting by means of gaussian model of video traffic. In *ATRS Melbourne Australia*, 1993.

[3] R. Addie and M. Zukerman. A gaussian traffic model for a B-ISDN statistical multiplexer. In *IEEE Globcom*, 1992.

[4] R. Gittaet. al. Phase 2 Simulation Results for Adaptive Random Access Protocol. In *IEEE 802.14 working group*, 1996.

[5] Azzan Albert. High Speed Cable Modems. In *McGraw-Hill*, pages 100–500, 1997.

[6] C. Blsdikian. Performance Analysis of Multi-slot n-ary Stack Random Access Algorithm. In *IEEE 802.14 working group*, 1996.

[7] G. Box and G. Jenkins. Time Series analysis: Forcasting and Control. In *Englewood Cliffs NJ: Prentice-Hall*, 1976.

[8] Mark C. and J. Liebeherr. Contention resolution schemes for 802.14 HFC Networks: A comparative evaluation. In *Dept of Electrical Eingineering and Computer Science University of Virginia Charlottesville VA22903*, Sept 1996.

[9] Ran F. C., Reuven C., Michael C., and Bob H. The MXL MAC protocol for HFC networks. In *HP Video Communication Division Cupertino CA*.

[10] H. Demuth and M. Beale. Neural Network Toolbox for use with MATLAB Computation, Visualization, Programming. In *The MathWorks Inc Natick MA 01760*, pages 4–03 – 5–55, June 1998.

[11] Data Over Cable Service Interface Specification (DOCSIS). Cable modem to customer premises equipment interface specification SP-CMCI. In *Cable Television Laboratories Inc*, Jan 2002.

[12] N. Golmie, S. Masson, C. Pieris, and D. Su. A MAC Protocol for HFC Networks: Design Issues and Performance Evaluation. In *National Institute of Standards and Technology, Maryland 20899.*

[13] R. Grunenfolder, J. Cosmas, S. Manthorpe, and A. O. Okafor. Characterization of video codecs as autoregressive moving average process and related queuing performance. In *IEEE J. Selected Areas on communication vol. 9 no. 3*, pages 284–293, 1991.

[14] A. Hiramatsu. Training techniques for neural network applications in ATM. In *IEEE Commun. Mag.*, pages 58–67, Oct. 1995.

[15] IEEE. IEEE Project 802.14 Cable-TV Functional Requirements and Evaluation Criteria. In *IEEE 802.14/ 94-002R2.*

[16] IEEE. Project Authorization Request (PAR), Local and Metropolitan Area Network Standards Committee. 1994.

[17] Cable Television Laboratories Inc. Radio frequency interface specfications SP-RFIv1.1. In *Data Over Cable Service Interface Specification (DOCSIS 1.1)*, 2000.

[18] Cable Television Laboratories Inc. Cable modem termination system network side interface specifications SP-CMTS-NST. In *Data Over Cable Service Interface Specification (DOCSIS 1.1)*, Feb 1996.

[19] Cable Television Laboratories Inc. Baseline privacy plus interface specification SP-BPI+. In *Data Over Cable Service Interface Specification (DOCSIS)*, Jan 2002.

[20] Cable Television Laboratories Inc. Operation support system interface specification SP-OSSIv1.1. In *Data Over Cable Service Interface Specification (DOCSIS 1.1)*, Jan 2002.

[21] Cable Television Laboratories Inc. Baseline privacy interface specification SP-BPI. In *Data Over Cable Service Interface Specification (DOCSIS 1.1)*, June 2001.

[22] MIL 3 Inc. Modeling Framework, OPNET Modeling Manual, Volume 1,. In *OPNET Doc. Viewer, Release 7.0.B.*

[23] Suming J. and Guangguo B. The utilization of redundancy bandwidth of upstream channel in HFC network. In *IEEE Trans. on broadcasting vol. 44 no. 2*, pages 216–225, June 1998.

[24] Aleksandar K. and Stephen W. Flexible bandwidth allocation in hybrid fiber coax distribution networks. In *C and C Research Laboratories NEC USA Inc. Princeton*, pages 983–987, 1995.

[25] Ying D. L., Chen Y. H., and Wei M. Y. Allocation and scheduling algorithm for IEEE 802.14 MCNS in Hybrid Fiber Coaxial Networks. In *IEEE Trans. on broadcasting vol. 44 no. 4*, pages 427–435, Dec 1998.

[26] M. Lee and D. Ahn. Cell loss analysis and design tradeoffs of nonblocking ATM switches with nonuniform traffic. In *IEEE/ACM Trans. on Networking vol. 3 no. 2*, pages 199–210, 1995.

[27] Robert C. Lehr and Jon. W. Mark. Traffic classification using neural networks. In *ECE Department University of Waterloo, Waterloo, Ontario, Canada*, 1997.

[28] R. Lipman. Pattern classification using neural networks. In *IEEE Commun. Mag.*, pages 47–64, Nov. 1989.

[29] M. Nomura, T. Fujii, and N. Ohta. Basic characteristics of variable rate video coding in ATM environment. In *IEEE J. on select. Area commun vol. 7 no 5*, 1989.

[30] E. Posner. Neural networks in communication. In *IEEE Trans. on Neural Networks vol. 1 no. 1*, pages 145–147, March 1990.

[31] D. Rumelhart, G. Hinton, and R. Williams. In parallel distributed processing, computational models of cognotion and perception, chapter 8. the MIT press, Cambridge, Massachusetts. In *Learning Internal Representations by Error Propagation*, pages 318–362, 1986.

[32] Jeremy S. S. Signal classification in digital telephone networks. In *MSC Thesis: University of Alberta.*, 1996.

[33] YUFEI SHI. ATM traffic generation and call admission control. In *MSC Thesis: University of Manitoba.*, 1998.

[34] K. Sriram. HFC MAC Protocol with Dynamically Variable vs Fixed Number of Request Mini-Slots: Performance and Capacity Ccomparisons. In *IEEE 802.14-96/120*, 1996.

[35] P. Tzerefos, V. Sdralia, C. Smythe, and S. Cvetkovic. Delivery of low bit rate isochronous stream over the DOCSIS 1.0 cable television protocol. In *IEEE Trans. on broadcasting vol. 45 no. 2*, pages 206–214, June 1999.

[36] P. Tzerefos, V. Sdralia, C. Smythe, and S. Cvetkovic. Performance characterization of the MCNS DOCSIS 1.0 CATV protocol with prioritised first come first served load scheduling. In *IEEE Trans. on broadcasting vol. 45 no. 2*, pages 196–205, June 1999.

[37] F. Yegeneglu, B. Jabbari, and Y. Zhang. Motion classified autoregressive modelling of variable bit rate video. In *IEEE Trans. on Circ. Syst. Video Tech vol. 3(1)*, 1993.