A Tutorial on DOCSIS: Protocol and Performance Models

Neel Shah, Demetres Kouvatsos
University of Bradford, UK
N.P.Shah@Bradford.ac.uk, D.D.Kouvatsos@scm.brad.ac.uk

Jim Martin, Scott Moser
Department of Computer Science
Clemson University, USA
jim.martin,smoser@cs.clemson.edu

Abstract- Since the first community antenna television (CATV) system was deployed in 1948, cable technology has advanced at an astounding rate. Today, multiple service providers (MSOs) are competing with telephone companies to deliver the long sought 'triple play' of voice, video and data to residential and business premises. Upstream data rates have progressed from dial-up speeds to 10Mbps and to 100s of Mbps in the near future. While there are competing standards, the Data over Cable Service Interface Specification (DOCSIS) has emerged as the single MAC and physical layer standard. We have developed a model of DOCSIS using the '*ns'* simulation package. In this tutorial paper we provide a detailed presentation of the DOCSIS protocol. To provide a deeper understanding of DOCSIS, we present the results of a simulation analysis focusing on the impact that DOCSIS has on TCP applications. The objectives of this tutorial are: 1)to provide an overview of the DOCSIS protocol; 2)to present the '*ns'* simulation model; 3)to present preliminary upstream contention and downstream transmission queuing network models (QNMs).

Keywords— DOCSIS, HFC Networks, Broadband access, TCP performance, Performance analysis

1 Introduction

CATV systems were introduced as a way to deliver television content to households located in hilly terrain that could not receive broadcast television. Over the years CATV companies began offering Internet access, data and telephony services to their customers in addition to television channels as a means of increasing revenue. Initially cable operators deployed proprietary systems. To stay competitive with other access technologies such as DSL, it was decided to open the cable modem market by creating a single standard hoping to make cable modems commodity items. The industry converged on the Data Over Cable Service Interface Specification (DOCSIS) which defines the Media Access Control (MAC) layer and the physical layer that is used to provide high speed data communication over a cable Hybrid Fiber Coax (HFC) network [1]. By pushing fiber further to the subscriber, fewer amplifiers are needed, noise is less of a problem and two-way data communication is possible[1].

Figure 1 illustrates a simplified DOCSIS environment. A Cable Modem Termination System (CMTS) interfaces with hundreds or possibly thousands of cable modems (CMs). A Cable Operator allocates a portion of the RF spectrum for data usage and assigns a channel to a set of CMs. A downstream RF channel of 6MHz (8MHz in Europe) is shared by all CMs in a one-to-many bus configuration (i.e., the CMTS is the only sender). In DOCSIS version 1.0, only one QoS class was supported, that is, 'best effort', for data transmission in the upstream direction. Upstream data rates were limited to 5.12Mbps. DOCSIS 1.1 provides a set of ATM-like QoS mechanisms. In addition,

[1] A group of cable modems that share an RF channel connect to an Optical/Electrical (O/E) node with a coaxial cable using a branch-and-tree topology.

the physical layer supports an upstream data rate of up to 10Mbps. DOCSIS 2.0 further increases upstream capacity to 30Mbps through more advanced modulation techniques and by increasing the RF channel allocation to 6.4MHz. The next generation DOCSIS (version 3.0) will support hundreds of Mbps in both the upstream and downstream channels through channel bonding techniques.

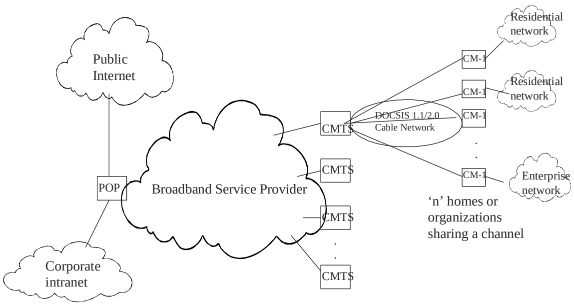


Figure 1. DOCSIS cable access environment

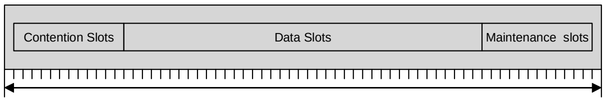


Figure 2. Example upstream MAP allocation

We have developed a model of the DOCSIS (version 1.1/2.0) MAC and physical layers using the '*ns*' simulation package [2]. In this tutorial, we will use the simulation model to demonstrate DOCSIS operation and behavior. Please refer to [3] for a detailed discussion of the validation of the model. This paper is organized as follows. First we present a brief summary of previous standards efforts and studies. The next section provides a detailed presentation of the DOCSIS protocol. We then present the '*ns*' DOCSIS model. To provide insight into the detailed behavior of DOCSIS, we document the results of experiments performed with the simulation model that focus on the impact of DOCSIS on TCP applications. We then present initial queuing network models. We end the paper with conclusions and with ideas for future work.

2 Related Work

After a brief discussion of past standardization efforts, this section describes examples of upstream bandwidth allocation algorithms, scheduling disciplines and bandwidth utilization enhancement mechanisms and gives an overview of research into the performance impact of TCP applications running on DOCSIS.

In the early 1990's, the cable industry developed a large number of schemes for supporting two-way data over cable. Several competing standards emerged.

IEEE 802.14: In 1994 the IEEE 802.14 working group was chartered to develop a MAC layer that would support both ATM and IP over HFC networks[4]. The upstream channel was TDMA with a slot size of 8 bytes. ATM's CBR, VBR, ABR and UBR services were supported over the HFC network. Primarily due to time constraints, the standard did not obtain vendor support.

Multimedia Cable Network System's (MCNS) DOCSIS: In response to competition from DSL, key multiple system operators (MSOs) in the early 1990s formed the MCNS to define a standard system for providing data and services over a CATV infrastructure. In 1997 they released version 1 of DOCSIS. The upstream channel was TDMA with a configurable slot size (referred to as a mini-slot). This standard was quickly endorsed by the cable industry. The DOCSIS standard is now managed by CableLabs, a non-profit research and development group funded by cable industry vendors and providers.

DAVIC/DVB: The non-profit Swiss organization Digital Audio Visual Council (DAVIC) was formed in 1994 to promote the success of digital audio-visual applications and services. The organization produced the DAVIC 1.2 and the very similar Digital Video Broadcast Return Channel for Cable (DVB-RCC) RF cable modem standards that defined the physical and MAC layers for bidirectional communications over CATV HFC networks. The DVB-RCC standard was popular in

Europe for several years. However, to benefit from the economies of scale, the European cable industry moved towards the EuroDOCSIS standard.

The operation of the IEEE 802.14 MAC layer is similar to that supported by DOCSIS. Therefore, prior 802.14 research is relevant. The authors in [5] found that TCP throughput over an 802.14 network is low primarily due to ACK compression. The authors propose two solutions: one involving piggybacking and a second involving TCP rate smoothing by controlling the ACK spacing. The authors found that piggybacking can help reduce the burstiness associated with the ACK stream in certain situations. However it is limited in its abilities to effectively match offered load over a range of operating conditions. The author's second solution is to control the TCP sending rate by measuring the available bandwidth and calculating an appropriate ACK rate and allowing the CM to request a periodic grant that provides sufficient upstream bandwidth to meet the required ACK rate.

The observation in [6] is that an HFC network presents difficulties for TCP due to the asymmetry between upstream and downstream bandwidths and due to high loss rates (the authors assume channel loss rates as high as 10-50%). Because of the known problems associated with TCP/Reno in these environments[7,8,9], the authors propose a 'faster than fast' retransmit operation where a TCP sender assumes that a packet is dropped when the first duplicate ACK is received (rather than the usual triple duplicate ACK indication).

The authors in [10] compare via simulation the request access delay (RAD) and throughput of the early versions of IEEE 802.14 (draft 2) and DOCSIS protocols. RAD is defined as the time from when a data job arrives at a CM to the time when the CM receives from the CMTS an acknowledgement of receipt of a BW request and it is therefore a measure of efficiency of the collision resolution algorithm. The benchmark that they used for fair comparison was examining the performance measures under the fine-tuned parameter settings of each technology and under the same mini-slot allocation strategy. Throughput was found to be very similar and the RAD was at most about 22% longer in DOCSIS than IEEE 802.14. The delay only differed in the load range of about 40-75%, for the rest of the load-range the delay was very similar. They also compare the performance of three upstream scheduling disciplines: Shortest Job First (SJF), Longest Job First (LJF) and Modified SJF. Here the short or long characteristic refers to the amount of bandwidth requested by the CM. The SJF discipline showed lower data transfer delay (DTD, defined as the time between receipt of BW request at CMTS and receipt of full data packets at CMTS and therefore a measure of efficiency of the upstream transmission scheduling algorithm) but poorer RAD. The authors state the reason for the larger RAD is that with small DTD, the CM queues are more likely to be empty and therefore CMs are less likely to piggyback requests. In order to reduce the RAD in the SJF discipline, the authors proposed an improvement, the Modified SJF, whereby allocated data grants to one CM are split and distributed over allocated mini-slots in smaller chunks and not granted altogether in one large burst in order to increase the probability for a station to piggyback its request. The Modified SJF exhibited the most balanced performance of the three disciplines.

The authors in [11] carry out a simulation study on a DOCSIS upstream scheduling discipline using the CableLabs simulation tool, the Common Simulation Framework (CSF) version 12 [12]. They analyzed the prioritized FCFS discipline whereby CMs are identified by their priorities in addition to their SIDs. When a request for BW arrives at the CMTS, it is queued in its corresponding priority buffer. When generating the next MAP, the CMTS first schedules, in order of time, events until the maximum number of information elements is reached (this is a set value) then it schedules all grants for requests in order of their priority until all appropriate upstream mini-slots are used up. They found a maximum (offered load exceeds upstream channel capacity) upstream throughput efficiency without use of concatenation of about 77% with large packet sizes and only 61% for small packet sizes. They also found that small packet sizes exhibited high access delay, however the delay can be reduced with concatenation. For long packet sizes, they concluded that the user-perceived QoS is heavily dependent on the prioritization mechanism used as access delays were favorable.

[13] provides an industry-aware overview of the emergence of telephony in cable networks and briefly covers many issues surrounding its use such as billing, security and QoS.

In [14] the authors propose two predictive methods to improve contention mini-slot throughput (both first time and collision retrials), therefore increasing overall upstream mini-slot throughput. In order to obtain maximum mini-slot throughput the number of contention mini-slots must be equal to the number of requests to resolve and therefore the aim of using the predictive methods is to obtain an accurate number of requests. They prove this statement by equating to zero the maximum of the binomial probability distribution function representing a mini-slot out of *m* mini-slots being successful in transmitting a single request out of r requests for all r requests. In the case of first time requests they propose estimating the number of initial requests in the next contention cycle using the statistically valid observation that the number of requests in a cycle is proportional to its duration. This method assumes that access of requests arriving during the current cycle is not controlled. In the case of request retries i.e. after collision, they propose using a statistical Most Likely Number of Requests (MLR) Table, from where they take the modal value from a collection of possible numbers as the estimated number of requests. These possible numbers of requests are given in the table as the results of different combinations of successful and collision mini-slots for a given number of allocated mini-slots. The authors claim that these predictions help to resolve collisions more efficiently than the 3-ary scheme (used in IEEE 802.14) and that they can accommodate bursty arrivals to a greater extent. The first method however gives poor estimates when the loads are light.

In [15] the authors propose a new upstream scheduling service, UGS with piggybacking and claim, via simulation, using real video traces, that it improves both the overall upstream bandwidth utilization and delay experienced by real-time upstream VBR video packets when compared to the existing UGS (low delay, CBR allocation) and rtPS (good bandwidth utilization for both CBR and VBR but higher delay) service flow scheduling disciplines. It must be noted that piggybacking is currently not permitted with UGS nor are any other contention mechanisms and therefore the aim of this proposal is to highlight possible areas of improvement to the current DOCSIS standard. The application of the proposed scheduling service assumes that the real-time VBR traffic has been ‘smoothed’ to reduce burstiness. The authors reference works which state that compressed digital video and other types of video streams are Long Range Dependent (LRD) exhibiting burstiness over multiple time scales. They also describe several ‘smoothing’ techniques most of which result in video streams comprising a significant constant bit rate component and an additional bursty component which cannot be avoided. It is this constant bit rate component that is supported by the UGS part of the scheduling discipline and the piggyback requests accommodate the extra bandwidth required for the bursts, while maintaining better delay constraints than when using rtPS.

The authors in [35] propose an analytic model for upstream contention in DOCSIS. The authors model scheduling at the CMTS, under various simplifying assumptions, of two traffic flows: real-time CBR traffic and non-real-time data traffic under UGS and BE DOCSIS contention services respectively. They consider the real-time flow having higher priority over data and implement this prioritization specifically using the Pre-emptive Resume (PR) scheduling discipline. They derive formulae for average delay of both traffic flows and the network throughput (efficiency of bandwidth utilization) using queuing theoretic concepts. They define the UGS flow delay as the difference between actual grant time and nominal grant time and the BE flow delay is defined as the time from arrival of a BE request at the CMTS to the arrival of the last bit of the respective data packet at the CMTS (this is the same as DTD mentioned in [10]). Neither the analytic model nor formulae have been verified. The experiments carried out show that the use of the PR scheduling discipline in this specific context does enable the CBR real-time traffic to meet its stringent time constraints. In addition, further experiments confirm intuition, showing that as the load of real-time traffic increases, the lower priority non-real-time traffic suffers, delay-wise. Finally it is shown that larger packets exhibit better bandwidth utilization efficiency. This is attributed to the fact that larger packets use a relatively smaller physical overhead.

In [36] the authors assess TCP performance when upstream transmission of large (data) packets is deferred until smaller (acknowledgement) packets have been served. The scheme is implemented by introducing two queues at the CMTS, one for small jobs and the other for large ones, with the small job queue being processed first and both adhering to a FCFS policy. No changes are required at the CM. The proposed scheduling method results in an increased upstream acknowledgement-packet transmission rate than would otherwise be possible without the enhanced scheduling. This in turn increases downstream TCP throughput and application response times. This scheduling discipline is clearly restricted to asymmetric-bandwidth services and limited in its application to emerging symmetric services such as interactive video and gaming applications, video conferencing, remote storage and virtual DVD.

Prior work with TCP over wireless asymmetric paths is relevant[16,17,18,19]. A network exhibits asymmetry with respect to TCP performance if achieved throughput is not solely a function of the link and traffic characteristics of the forward direction but in fact depends on the impact of the reverse direction. Most of the prior work was focused on highly asymmetric paths with respect to bandwidth where the normalized asymmetry level (i.e., the ratio of raw bandwidths to the ratio of packet sizes in both directions) typically would be on the order of 2-4 [16]. In DOCSIS the upstream channel exhibits packet rate asymmetry due to low upstream packet rates with respect to downstream capacity. However the problem symptoms are similar. Various methods have been proposed to alleviate the TCP over asymmetric path problems including header compression and modified upstream queue policies (drop-from-front, ACK prioritization, ACK filtering). Some of these ideas can be applied to DOCSIS. For example, a CM that supports ACK filtering could drop 'redundant' ACKs that are queued. While this would increase the acknowledgement rate, it would also increase the level of ACK compression. ACK reconstruction could be implemented in the CMTS to prevent the increased level of ACK compression from affecting performance.

The cable industry is undergoing a period of rapid change. Fueled primarily by VoIP deployments, the Operating Support Systems (OSS) of MSOs are being upgraded. The academic community has focused primarily on video-on-demand architectures and related video transport issues [20,21,22,23]. Our work is motivated by the fact that the academic community has largely ignored the physical and MAC layers of HFC networks and has therefore not significantly contributed to the evolution of these systems.

3 The DOCSIS Protocol

We break the presentation of DOCSIS into the following four components: basic operation, QoS, Security and Performance.

3.1 Basic operation

Once powered on, the CM establishes a connection to the network and maintains this connection until the power to it is turned off. Registration of the CM onto the network involves acquiring upstream and downstream channels and encryption keys from the CMTS and an IP address from the ISP. The CM also determines propagation time from the CMTS in order to synchronize itself with the CMTS (and in effect the network) and finally logs in and provides its unique identifier over the secure channel. Due to the shared nature of these cable networks, transmissions are encrypted in both the upstream and downstream directions [24].

DOCSIS specifies an asymmetric data path with downstream and upstream data flows on two separate frequencies. The upstream and downstream carriers provide two shared channels for all CMs. On the downstream link the CMTS is a single data source and all CMs receive every transmission. On the upstream link all CMs may transmit and the CMTS is the single sink.

Packets sent over the downstream channel are broken into 188 byte MPEG frames each with 4 bytes of header and a 184 byte payload. Although capable of receiving all frames, a CM is typically configured to receive only frames addressed to its MAC address or frames addressed to the broadcast address. In addition to downstream user data, the CMTS will periodically send management frames. These frames include operations such as ranging, channel assignment, operational parameter download, CM registration, etc. Additionally, the CMTS periodically sends MAP messages over the downstream channel that identify future upstream TDMA slot assignments over the next MAP time. The CMTS makes these upstream CM bandwidth allocations based on CM requests and Quality of Service (QoS) policy requirements.

The upstream channel is divided into a stream of time division multiplexed 'mini-slots' which, depending on system configuration, normally contain from 8 to 32 bytes of data. The CMTS must generate the time reference to identify these mini-slots. Due to variations in propagation delays from the CMTS to the individual CMs, each CM must learn its distance from the CMTS and compensate accordingly such that all CMs will have a system wide time reference to allow them to accurately identify the proper location of the mini-slots. This is called ranging and is part of the CM initialization process.

Ranging involves a process of multiple handshakes between the CMTS and each CM. The CMTS periodically sends sync messages containing a timestamp. The CMTS also sends periodic bandwidth allocation MAPs. From the bandwidth allocation MAP the CM learns the ranging area from the starting mini-slot number and the ranging area length given in the message. The CM will then send a ranging request to the CMTS. The CMTS, after evaluating timing offsets and other parameters in the ranging request, will return to the CM a ranging response containing adjustment parameters. This process allows each CM to identify accurately the timing locations of each individual mini-slot.

In addition to generating a timing reference so that the CMs can accurately identify the mini-slot locations, the CMTS must also control access to the mini-slots by the CMs to avoid collisions during data packet transmissions. Figure 2 illustrates a possible allocation MAP that includes allocated slots for contention requests, user data and management data. For best effort traffic, CMs must request bandwidth for upstream transmissions. There are several mechanisms available: contention BW requests, piggybacked BW requests and concatenated BW requests.

3.1.1Contention BW requests

The CMTS must periodically provide transmission opportunities for CMs to send a request for bandwidth to the CMTS. As in slotted Aloha networks [25], random access bandwidth request mechanisms are inefficient as collisions will occur if two (or more) CMs attempt to transmit a request during the same contention mini-slot. Most implementations will have a minimum number of contention mini-slots to be allocated per MAP time, and in addition, any unallocated mini-slot will be designated as a contention mini-slot.

When a packet arrives at the CM that requires upstream transmission, the CM prepares a contention-based BW request by computing the number of mini-slots that are required to send the packet including all framing overhead. The contention algorithm requires the CM to randomly select a number of contention mini-slots to skip before sending (an initial back-off). This number is drawn from a range between 0 and a value that is provided by the CMTS in each MAP. The values sent are assumed to be a power of 2, so that a 5 would indicate a range of 0 – 31. After transmission, if the CM does not receive an indication that the request was received, the CM must randomly select another number of contention mini-slots to skip before retrying the request. The CM is required to exponentially back-off the range with each collision with the maximum back-off specified by a maximum back-off range parameter contained in each MAP. The CM will drop the packet after it has attempted to send the request 16 times.

As an example of the operation of the truncated exponential back-off algorithm, assume that the CMTS has sent an initial back-off value of 4, indicating a range of 0 – 15, and a maximum back-off value of 10, indicating a range of 0 – 1023. The CM, having data to send and looking for a contention mini-slot to use to request bandwidth, will generate a random number within the initial back-off range. Assume that an 11 is randomly selected. The CM will wait until eleven available contention mini-slots have passed. If the next MAP contains 6 contention mini-slots, the CM will wait. If the following MAP contains 2 contention mini-slots, a total of 8, the CM will still continue to wait. If the next MAP contains 8 contention mini-slots the CM will wait until 3 contention mini-slots have passed, 11 total, and transmit it's request in the fourth contention mini-slot in that MAP.

The CM then looks for either a Data Grant from the CMTS or a Data Acknowledge. If neither is received, the CM assumes a collision has occurred. The current back-off range is then doubled, i.e. the current value is increased from 4 to 5 making the new back-off range 0 – 31, and the process is repeated. The CM selects a random value within this new range, waits the required number of contention mini-slots, and resends its request. The back-off value continues to be incremented, doubling the range, until it reaches the maximum back-off value, in this example 10, or a range of 0 – 1023. The current back-off range will then remain at this value for any subsequent iterations of the loop. The process is repeated until either the CM receives a Data Grant or Data Acknowledge from the CMTS, or the maximum number of 16 attempts is reached.

3.1.2 Piggybacked BW requests

To minimize the frequency of contention-based bandwidth requests, a CM can piggyback a request for bandwidth on an upstream data frame. For certain traffic dynamics, this can completely eliminate the need for contention-based bandwidth requests.

The MAC header has the capability of defining an Extended Header field. Extended Headers can be used to request bandwidth for additional upstream transmissions, during the current data transmission. This allows the request for bandwidth to be made outside of the contention process and thereby reduces the occurrence of collisions and consequently the access delay. This process will allow the transmission of data, without the possibility of collisions, when there are large packet flows to be passed upstream.

3.1.3 Concatenated BW requests

DOCSIS provides both Fragmentation MAC Headers, for splitting large packets into several smaller packets, and Concatenation MAC Headers, to allow multiple smaller packets to be combined and sent in a single MAC burst. Concatenation can also be used to reduce the occurrence of collisions by reducing the number of individual transmission opportunities needed. Concatenation is the only method for transmitting more than one packet in a single transmission opportunity. The CMTS, receiving the Concatenation MAC Header, must then 'unpack' the user data correctly. The Concatenation MAC Header precludes the use of the Extended Header field and therefore piggybacking of future requests can not be done in a concatenated frame.

3.2 QoS

DOCSIS manages bandwidth in terms of Service Flows that are specified with Service Flow IDs (referred to as a SID). Traffic arriving at either the CMTS or the CM for transmission over the DOCSIS network is mapped to an existing SID and treated based on the profile. A CM will have at least 2 SIDs allocated, one for downstream Best Effort Service (BE) traffic and a second for upstream BE traffic. The upstream SIDs at the CM are implemented as FIFO queues. Other types of traffic, such as VoIP, might be assigned to a different SID that supports a different scheduling service; e.g., Unsolicited Grant Service (UGS) for toll quality telephony. The DOCSIS specification purposely does not specify the upstream bandwidth allocation algorithms so that vendors are able to develop their own solutions. DOSCIS requires CMs to support the following set of scheduling services: Unsolicited Grant Service (UGS), Real-Time Polling Service (rtPS), Unsolicited Grant Service with Activity Detection (UGS-AD), Non-Real-Time Polling Service (nrtPS) and Best Effort Service (BE).

Unsolicited Grant Service (UGS); Designed to support real-time data flows generating fixed size packets on a periodic basis. For this service the CMTS provides fixed-size grants of bandwidth on a periodic basis. The CM is prohibited from using any contention requests. Piggybacking is prohibited. All CM upstream transmissions must use only the unsolicited data grants.

Real-Time Polling Service (rtPS); Designed to support real-time data flows generating variable size packets on a periodic basis. For this service the CMTS provides periodic unicast request opportunities regardless of network congestion. The CM is prohibited from using any contention requests. Piggybacking is prohibited. The CM is allowed to specify the size of the desired grant. These service flows effectively release their transmission opportunities to other service flows when inactive [1], demonstrating more efficient bandwidth utilization than UGS flows at the expense of delay, which is worse.

Unsolicited Grant Service with Activity Detection (UGS-AD); Designed to support UGS flows that may become inactive for periods of time. This service combines UGS and rtPS with only one being active at a time. UGS-AD provides Unsolicited Grants when the flow is active and reverts to rtPS when the flow is inactive.

Non-Real-Time Polling Service (nrtPS); Designed to support non real-time data flows generating variable size packets on a regular basis. For this service the CMTS provides timely unicast request opportunities regardless of network congestion. The CM is allowed to use contention request opportunities.

Best Effort Service (BE); Designed to provide efficient service to best effort traffic. The CM is allowed to use contention or piggyback requests for bandwidth.

In the downstream direction, arriving packets are classified into a known SID and treated based on the configured service definition. For best effort traffic, the service definition is limited to a configured service rate. For downstream traffic, the CMTS provides prioritization based on SID profiles, where each SID has its own queue. Management frames, in particular MAP frames, are given highest priority. Telephony and other real-time traffic would be given next priority. Best effort traffic would share the remaining available bandwidth. There is also a single downstream transmission queue. Queuing occurs at the SID queues only if downstream rate control is enabled. All downstream queues are FIFO with the exception that MAP messages are inserted at the head of the transmission queue. The maximum size of each queue is a modeling parameter.

3.3 Security

Historically, cable systems have had an image as being insecure. The 'always-on' capability attracts attacks on subscriber networks. Subscriber networks that have Microsoft Windows OS machines with improper security settings have caused significant problems[2]. The security of cable networks has also been questioned since, as in a bus-based Ethernet LAN, data is received by all CMs. By default, a CM is placed in non-promiscuous mode, however it is possible for a subscriber to change the configuration and to have the CM receive all data sent over the RF channel. Further, it is possible to

[2] The security vulnerability occurs when a subscriber configures his/her network with file or print sharing. There are many reports of how dangerous this can be, see for example http://cable-dsl.home.att.net/netbios.htm#Scour.

increase the provisioned service rates by modifying the configuration. To counter this, CableLabs has extended DOCSIS with the Baseline Privacy Interface (referred to as BPI+).

BPI+ addresses two areas of concern: securing the data as it travels across the network, and preventing the theft of service. BPI+ requires encryption of the frames, essentially forming a Virtual Private Network (VPN) for all transmissions between the CMTS and the CM to protect the customer's data as it traverses the coaxial cable. Triple-DES is used for encryption of a DOCSIS MAC Frame's packet data. Public key encryption is used by the CM to securely obtain the required keys from the CMTS. Each CM must contain a key pair for the purpose of obtaining these keys from the CMTS.

To prevent the theft of service BPI+ requires secure modem authentication procedures be used to verify the legitimacy of a particular CM. CMs download their firmware from the service provider each time they boot. BPI+ requires the CM to successfully boot only if the downloaded code file has a valid digital signature. When a CM makes an authorization request to the CMTS it must provide a unique X.509 digital certificate. After receiving a properly signed X.509 certificate and verifying the 1024 bit key pair the CMTS will encrypt an authorization key using the corresponding public key and send it to the CM. A trust chain is developed by using a three level certificate hierarchy. At the top level is the Root Certification Authority (CA) which belongs to CableLabs. The Root CA uses its certificate to sign a Manufacturer's CA certificate at the second level. The manufacturer CA certificates are then used to sign individual certificates for each CM produced by that particular manufacturer. This process insures that a given CM is legitimate and that the keys for encrypting the user's data are only distributed to trusted CMs.

Although DOCSIS specifies the use of these security procedures to protect both the service provider and the customer, like all security measures, if they are not used the system is vulnerable. Recent polls and press reports indicate that the majority of the cable network operators have not enabled the security methods required by DOCSIS. With security becoming of paramount importance it is imperative that the security measures required by the standard be employed and enabled.

3.4 Performance

The following discussion summarizes the main issues surrounding DOCSIS performance. The bottleneck in a DOCSIS system is the upstream channel and in particular its ability to transport packets at a high rate of speed. This upstream packet rate limitation impacts both downstream and upstream throughput.

In the downstream direction, TCP throughput is limited by the rate at which TCP ACK packets can be sent over the upstream channel. For a sustained downstream TCP flow that is not limited by send or receive windows, the maximum throughput is:

$$T_{\max} = \max\left[\frac{2 * packetSize * 8}{2 * MapTime}, DownstreamServiceRate\right]$$

This is a simplification and depends on specific DOCSIS configuration parameters. The model assumes that the bottleneck in the path between the TCP sender and receiver is indeed the upstream channel. It assumes no piggybacking and no concatenation. Further, it assumes that the TCP receiver acknowledges every other data segment. If all of this holds, then 2 IP packets will be transmitted (in the downstream direction) each time an ACK is delivered. In this scenario, the upstream channel will deliver one ACK packet every 2 MAP times resulting in 2 TCP segments to be clocked out every 2 MAP times (assuming the connection is in congestion avoidance).

For a typical MAP time of .002 seconds and a TCP/IP packet size of 1500 bytes, the maximum downstream throughput is roughly 6Mbps not accounting for overhead. After accounting for protocol header, framing and FEC overhead, the application throughput is roughly 88% or 5.28Mbps.

Piggybacking will not help significantly in this scenario. Piggybacking increases efficiency for systems with backlogged upstream flows. However piggybacking is not effective for bursty streams carrying TCP ACK packets. Concatenation can significantly improve efficiency as it increases the rate at which ACK packets are sent upstream. In this scenario, it is possible for a single downstream TCP connection to consume very high bandwidths (if the service rates are high). For the above example, if we assume that up to 10 ACKs can be carried in a concatenated frame, the TCP session will consume 20 packets per two MAP times or 60Mbps (simulation experiments confirm this). Unfortunately concatenation can significantly impact TCP dynamics by perturbing the TCP ACK spacing. This has been shown to possibly lead to higher loss rate [26,27,28]. Several CM vendors have implemented ACK filtering as a further technique to increase efficiency. Our experiments show that ACK filtering provides the same benefit as concatenation (depending on the implementation ACK filtering can increase the burtiness of the TCP sending process increasing the loss rate) and is arguably not worth the additional complexity.

In the upstream direction, bulk TCP streams are also limited by the upstream packet rate. The maximum throughput is:

$$T_{\max} = \max\left[\frac{packetSize*8}{2*MapTime}, UpstreamServiceRate\right]$$

Without concatenation, only 1 packet can be delivered upstream every 2 MAP times. Using the example above, this translates to a maximum upstream throughput of 2.73Mbps. As in the downstream discussion, this is a maximum and will not be achievable for certain DOCSIS or network configurations. Piggybacking can be helpful to ensure that 1 packet does indeed get delivered every cycle by eliminating contention delays. Concatenation is of marginal help. If 2 full IP packets are concatenated, this effectively doubles the upstream packet rate and subsequently the throughput. However most networks will not allow this as it significantly increases the access delay experienced by packets sent by other CMs.

4 Quantitative Analysis

A DOCSIS network is a complex system [11]. There are many configuration parameters and it is difficult to know a priori how a particular combination of parameters will impact a traffic mix. To provide insight into the dynamics and impacts of DOCSIS on applications, we present a simulation implementation of DOCSIS for the '*ns*' simulation package[3][3]. In addition preliminary QNMs modeling contention and downstream transmission are presented in this section.

4.1 Simulation

[3] The DOCSIS 'ns' simulation model is publicly available at http://www.cs.clemson.edu/~jmarty/docsis.html.

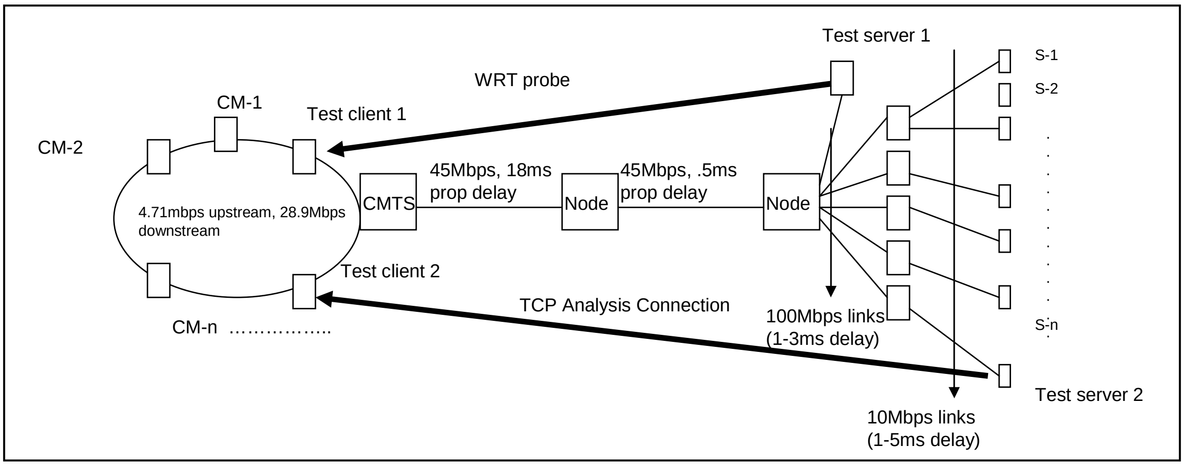


Figure 3. Simulation network

The simulation model implements the DOCSIS architecture defined in [1] with the following restrictions: 1)CMs are limited to a single default best effort service flow and a single UGS or rtPS flow; 2)the model is limited to one upstream channel for each downstream channel; 3)the model does not support dynamic service provisioning; 4)physical layer impairments are not modeled; 5)the model assumes that the CMTS and the CM clocks are synchronized.

The model accounts for MAC and physical layer overhead including forward error correcting (FEC) data in both upstream and downstream directions. For our simulations we assume a FEC overhead of 4.7% (8% in the upstream direction) and model this by reducing channel capacity accordingly[4]. The downstream and upstream channels support an optional service rate. Service rates are implemented using token buckets where the rate and maximum token bucket size are simulation parameters.

Traffic arriving at either the CMTS or the CM for transmission over the DOCSIS network is mapped to an existing SID and treated based on the profile. In our model, when a CM begins, it registers itself with the CMTS which establishes the default upstream and downstream SID. A CM has an upstream FIFO queue for each SID. In the downstream direction there are per SID queues as well as a single transmission queue. Queuing occurs at the SID queue only if downstream rate control is enabled. All downstream queues are FIFO with the exception that MAP messages are inserted at the head of the transmission queue. The maximum size of each queue is a simulation parameter.

The scheduler has a configured MAP time (i.e., a MAP_TIME parameter) which is the amount of time covered in a MAP message. The MAP_FREQUENCY parameter specifies how often the CMTS sends a MAP message. Usually these two parameters are set between 1 – 10 milliseconds. The scheduling algorithm supports dynamic MAP times through the use of a MAP_LOOKAHEAD parameter which specifies the maximum MAP time the scheduler can 'look ahead'. If this parameter is 0, MAP messages are limited to MAP_TIME amount of time in the future. If set to 255 the scheduler may allocate up to 255 slots in the future. This is only used on BE traffic and only if there are no conflicting periodic UGS or rtPS allocations.

The grant allocation algorithm (i.e., the scheduling algorithm) models requests as jobs of a non-preemptive soft real-time system[29]. There can be two types of the jobs in the system: periodic and aperiodic. Periodic jobs result in UGS periodic data grants and rtPS periodic unicast request grants.

[4] To account for FEC overhead we reduce the upstream channel capacity by 8%. This approximation was suggested in http://www.cisco.com/warp/public/109/data_thruput_docsis_world_19220.shtml. The DOCSIS framing overhead adds an additional 30 bytes to an IP packet. A system tick of 6.25 microseconds and an effective channel capacity of 4.71Mbps leads to 18 bytes of data per slot for a total of 85 slots required for a 1500 byte IP packet.

Aperiodic jobs are in response to rtPS and Best-effort requests for upstream bandwidth. Every job has a release time, a deadline and a period. The release-time denotes the time after which the job can be processed. The deadline denotes the time before which the job must be processed. For periodic jobs, the period is used to determine the next release time of the job.

The scheduler maintains four queues of jobs where a lower number queue has priority over a higher number queue. The first and second queues contain UGS and rtPS periodic jobs respectively. UGS jobs are unsolicited grants and rtPS jobs are unsolicited polls to CMs for bandwidth requests. The jobs in these queues are maintained in increasing order of relative deadlines. The third queue contains all the bandwidth requests that were in response to previous unicast request grants. Similarly, the fourth queue contains the bandwidth requests that arrived successfully from the contention request process. The latter two queues are serviced in a FIFO manner. The CMTS processes jobs from the four queues in strict priority order with no preemption.

The parameters associated with a UGS service flow include the grant size, the grant interval and the max-tolerated-jitter. When a CM registers a UGS flow with the CMTS, the CMTS releases a periodic job in the system with release time set to the current time and the deadline is set to the release time + max-tolerated-jitter. Finally, the period is set to the grant interval. After every period, a new instance of the job is released.

The same algorithm is used for rtPS except that the max-poll-jitter is used to determine the deadline. Requests for bandwidth allocations from best-effort contention or from rtPS polling are treated as aperiodic jobs. Periodic jobs with the earliest deadline are serviced first. Remaining bandwidth is then allocated to aperiodic jobs. The scheduler has an additional parameter (*proportion*) that is used to establish a relative priority between rtPS allocations and BE allocations.

In prior work we found that DOCSIS configuration parameters can significantly impact network performance[30,31]. To demonstrate the impact that DOCSIS has on TCP/IP applications, we provide the results of simulation experiments. We group the experiments into one of two sets. Both sets are based on the network depicted in Figure 3. The second set differs from the first set in several significant ways: 1)the scheduler allocates unused slots for contention requests; 2)the number of IP packets allowed in a concatenated frame is no longer limited to two; 3)the buffer size at the CMTS downstream queue is set to 300 packets rather than 50 packets; 4)the number of system ticks per slot was increased to 5 from 4 which decreased the number of slots per map from 80 to 64.

All experiments involved a variable number of CMs (i.e., CM1 through CM-n in Figure 3) that interact with a set of servers (S-1 through S-n). The RTT from the CMs to the servers was randomly selected in the range between 42 milliseconds and 54 milliseconds. The network and web traffic models were based on the “flexbell” model defined in [32]. In addition to downstream web traffic, we configured 5% of the CMs to generate downstream low speed UDP streaming traffic (i.e., a 56Kbps audio stream), 2% of the CMs to generate downstream high speed UDP streaming traffic (i.e., a 300Kbps video stream) and 5% of the CMs to generate downstream P2P traffic. The P2P model (based on [33]) incorporated an exponential on/off TCP traffic generator that periodically downloads on average 4Mbytes of data with an average idle time of 5 seconds between each download.

The simulation model parameters are shown in Figure 4. In both sets of experiments we varied the MAP_TIME and the number of CMs. For a given MAP_TIME setting, we varied the number of CMs from 100 to 500[5]. We do this for six MAP_TIME settings ranging from .001 to .01 seconds. For each experiment we obtained the following statistics:

[5] Many providers provision a downstream RF channel by assigning 2000 households per channel which makes our range of active CMs reasonable.

Collision rate: Each time a CM detects a collision it increments a counter. The collision rate is the ratio of the number of collisions to the total number of upstream packet transmissions attempted.
Downstream and upstream channel utilization: At the end of a run, the CMTS computes the ratio of the total bandwidth consumed to the configured raw channel bandwidth. The utilization value reflects the MAC and physical layer overhead including FEC bits.
Average upstream access delay: All CMs keep track of the delay from when an IP packet arrives at the CM in the upstream direction until when it actually gets transmitted. This statistic is the mean of all of the samples.
Web response time: a simple TCP client server application runs between Test Client 1 and the Test Server 1. Test Server 1 periodically sends 20Kbytes of data to Test Client 1. With each iteration, the client obtains a response time sample. The iteration delay is set at 2 seconds. At the end of the test, the mean of the response times is computed. The mean web response time (WRT) can be correlated to end user perceived quality by using a very coarse rule of thumb that says end users are bothered by lengthy download times when the mean WRT metric value exceeds 1 second. We do not claim this to be an accurate measure of end user quality of experience. Instead, it simply provides a convenient performance reference.

Model Parameters

Upstream bandwidth 5.12Mbps
Preamble 80 bits
Downstream bandwidth 30.34Mbps
4 ticks per minislot
Default map time: 2 milliseconds (80 minislots per map)
Fragmentation Off, MAP_LOOKAHEAD = 255 slots
Concatonation ON
Backoff Start: 8 slots, Backoff stop: 128 slots
12 contention slots, 3 management slots
Simulation time: 1000 seconds

Web Traffic Model Parameters

Inter-page: pareto model, mean 10 and shape 2
Objects/page: pareto model, mean 3 and shape 1.5
Inter-object: pareto model, mean .5 and shape 1.5
Object size: pareto model, mean 12 (segments) shape 1.2

Figure 4. Simulation parameters for set 1and set 2 experiments

Experiment Set 1
When the dominant application is web browsing the majority of data travels in the downstream direction. However, for certain configurations, the system can become packet rate bound in the upstream direction which can limit downstream throughput due to a reduced acknowledgement rate. For the set 1 experiments, piggybacking and concatenation were enabled however the maximum number of packets that could be concatenated into a single upstream transmission was limited to 2.

Figure 5 shows that the collision rates get extremely high as the number of active CMs increase. When only 100 users are active, the collision rate is about 50%. As the load increased, the collision rate approached 90-100% depending on the MAP_TIME setting. The behavior of the system is influenced by several MAC protocol parameters. First, the number of contention slots assigned per map (i.e., the CONTENTION_SLOTS) directly impacts the collision rates at high loads. This set of experiments used a fixed number of contention slots (12) per MAP which, as illustrated in Figure 5, is insufficient at high loads. The set of curves in Figure 5 illustrate the collision rate at different MAP_TIME settings. The collision rate is roughly 10 percent higher for the largest MAP_TIME than for the smallest MAP_TIME. This is a direct result of the MAP allocation algorithm which allocates a fixed number of contention slots each map time. As the MAP_TIME grows the bandwidth allocated for contention requests effectively is reduced. Another critical pair of parameters are the backoff start and stop which determine the average backoff delay a CM uses after it detects a collision. A large range is necessary to support many CMs but too large a range can unnecessarily increase the average upstream access delay.

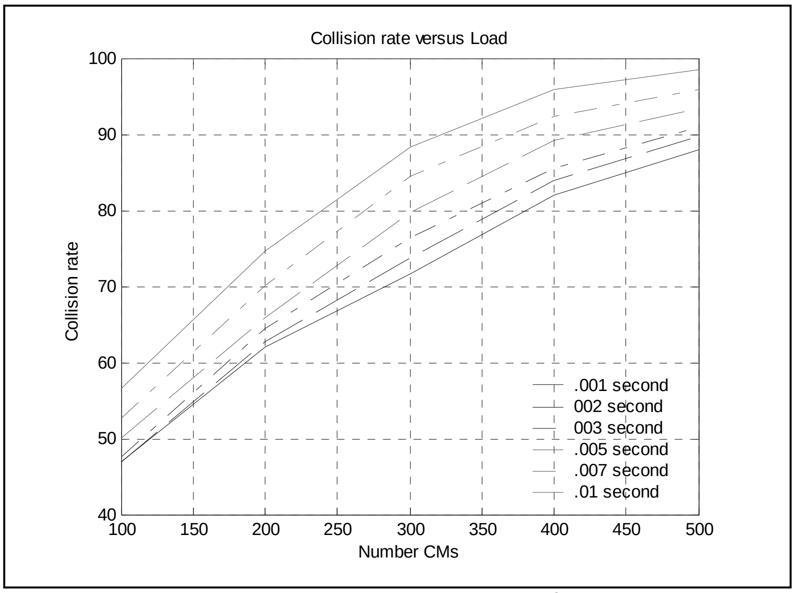


Figure 5. Upstream collision rates as the number of CMs increase

Figures 6a and 6b plot the channel utilization as the load increases. The downstream utilization reaches a maximum of about 64% with a MAP_TIME setting of .001 second. In this case, 12 contention slots per MAP is sufficient. For smaller MAP_TIME values, the downstream utilization ramps up to its maximum value and then decreases at varying rates as the load increases. As the collision rate grows, downstream TCP connection throughput decreases. Larger MAP_TIME values result in fewer contention request slots allocations leading to higher collision rates and reduced downstream utilization. Further illustrating this behavior, Figure 7a shows that the average upstream access delay becomes very large at high loads when configured with large MAP_TIME settings. Even for lower MAP_TIME values, the access delay was significant. For a MAP_TIME of .002 seconds, the access delay exceeded .5 seconds at the highest load level. To assess the impact of the cable network on end-to-end performance we monitored web response times. Using the rule of thumb described earlier, Figure 7b suggests that for MAP_TIME settings less than .005, up to 300 users can be active before performance becomes bothersome to end users.

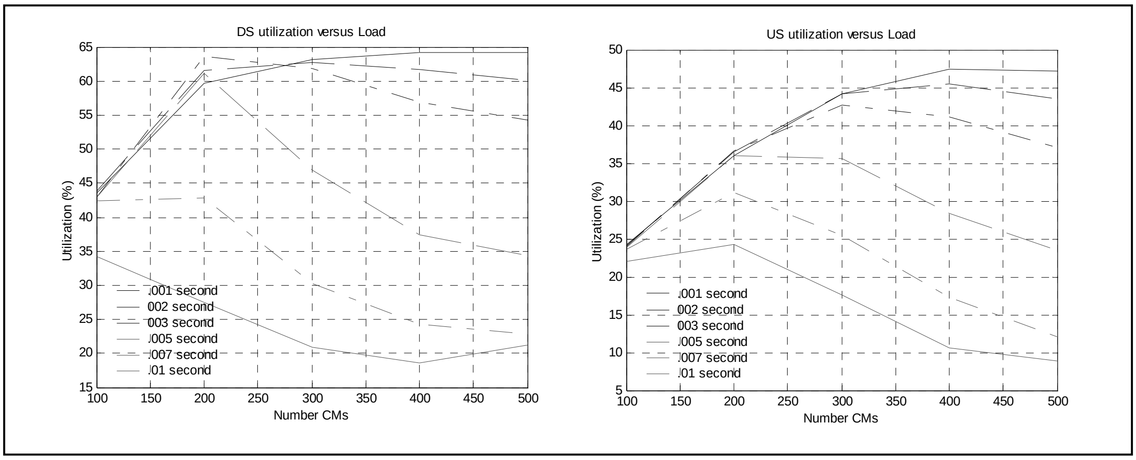
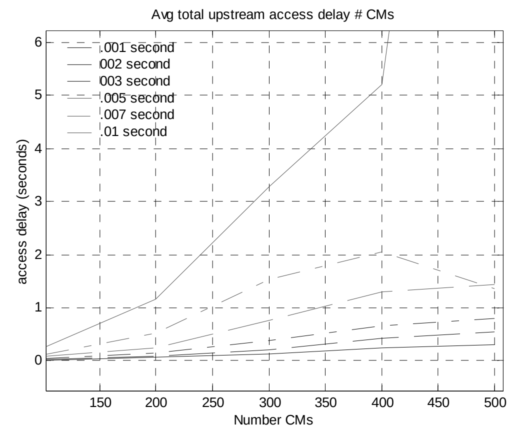


Figure 6a. Downstream channel utilizations

Figure 6b. Upstream channel utilizations



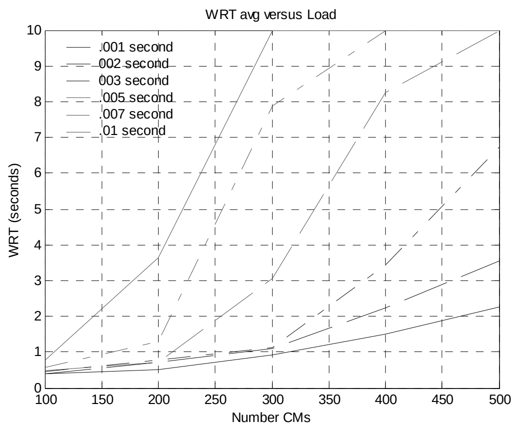
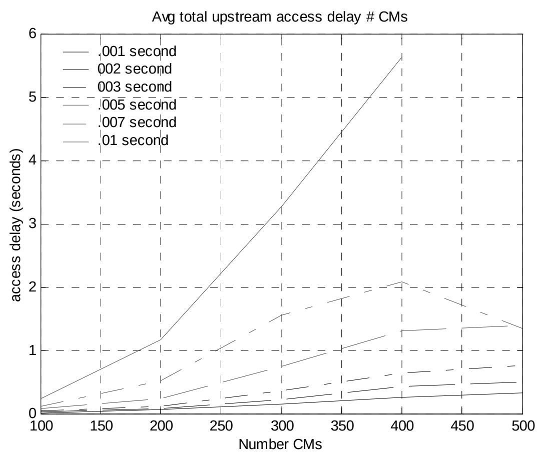


Figure 7a. Upstream access delay (no rate control)

Figure 7b. Web response time metric results

Rather than making the full channel capacity available to subscribers, MSOs typically offer different service plans where each plan is defined by a service rate. For example, Charter communications offers 3Mbps downstream rate and 512Kbps upstream rate[34]. While reduced service rates prevent customers from consuming more than their fair share of bandwidth at the expense of other customers, they offer little benefit when the network becomes congested. Figures 8a and 8b illustrate the results of an experiment that is identical to the web congestion scenario except that CMs are restricted to a 2Mbps downstream service rate. Figure 8a shows the average upstream access delay is almost identical to that observed in the scenario without rate control. The WRT results shown in Figure 8b further suggest that a 2Mbps downstream service rate is of little use.



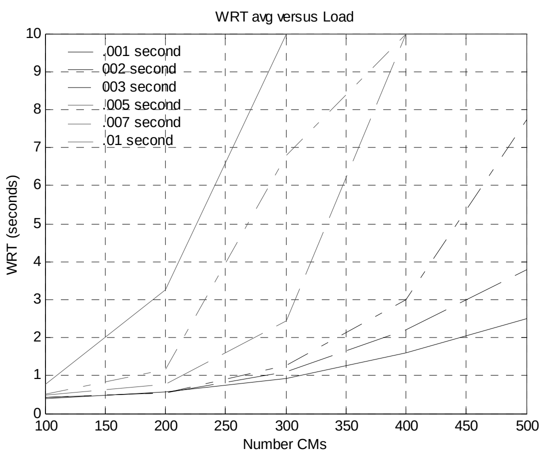


Figure 8a. Upstream access delay (with rate control)

Figure 8b. Web response time metric results

Experiment Set 2

In the set 2 experiments, the change that had the most impact was the increased bandwidth allocated for upstream contention requests. Figure 9 shows that collision rate ranged from 2% to 37%. Collision rates were lowest for the runs with smaller MAP times. As the system becomes busy the number of unused slots gets smaller which reduces the number of contention request slots. In other words, the bandwidth allocated for contention slots is greater for small MAP times. Figures 10 shows that the MAP time has little impact on channel utilizations. Piggybacking was highly effective in this scenario. Figure 11 illustrates that 50%-90% of all packets sent upstream used a piggyback bandwidth request. The runs with large MAP times were able to take advantage of piggybacking more than the runs with small MAP times because there is more time for packets to accumulate while waiting for a data grant. We reran the experiments with concatenation enabled and saw similar results with the exception that extreme levels of TCP ACK compression occurred. Since all nodes in the simulator were configured with adequate buffers, performance was not impacted by the bursty traffic dynamics caused by the ACK compression. However, it has been shown that ACK compression leads to higher loss rates and that it makes it difficult for protocols that estimate bottleneck bandwidths or that monitor packet delays to operate correctly [26,27,28].

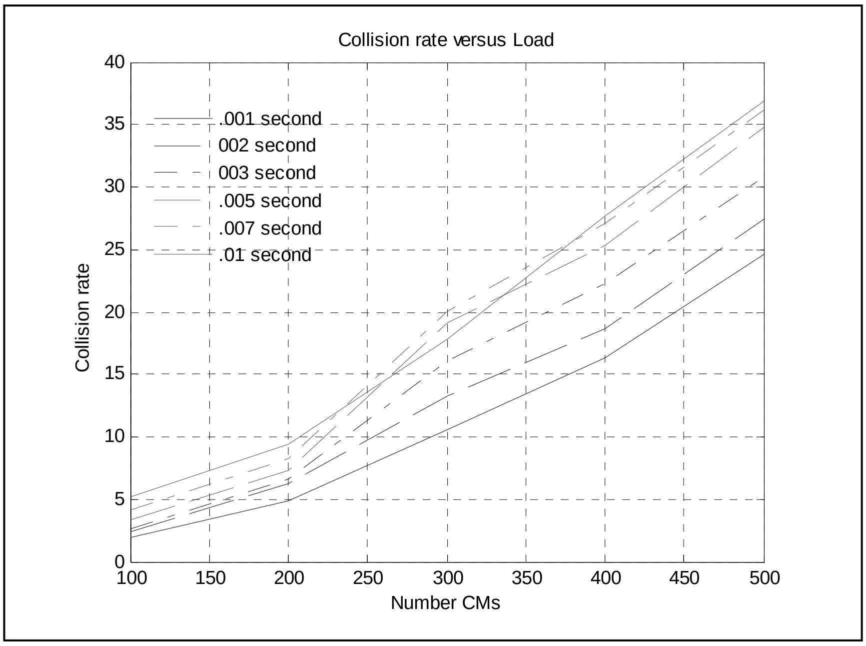


Figure 9. Upstream collision rates

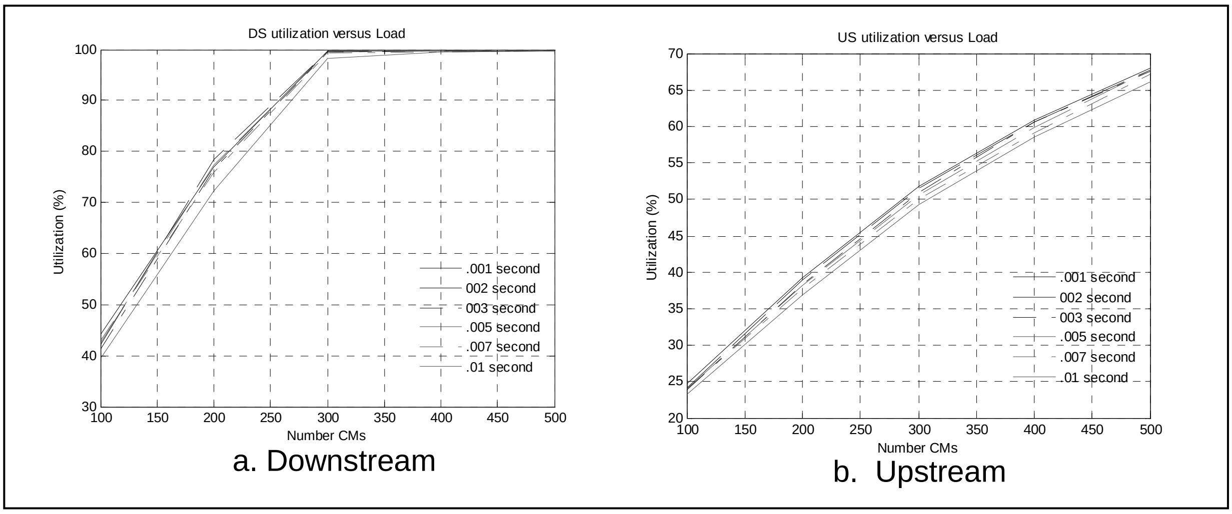
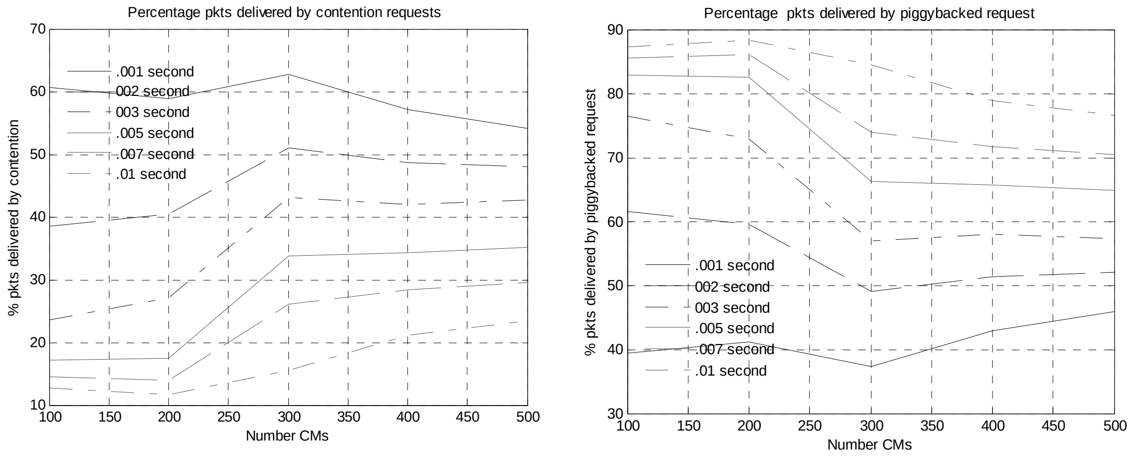
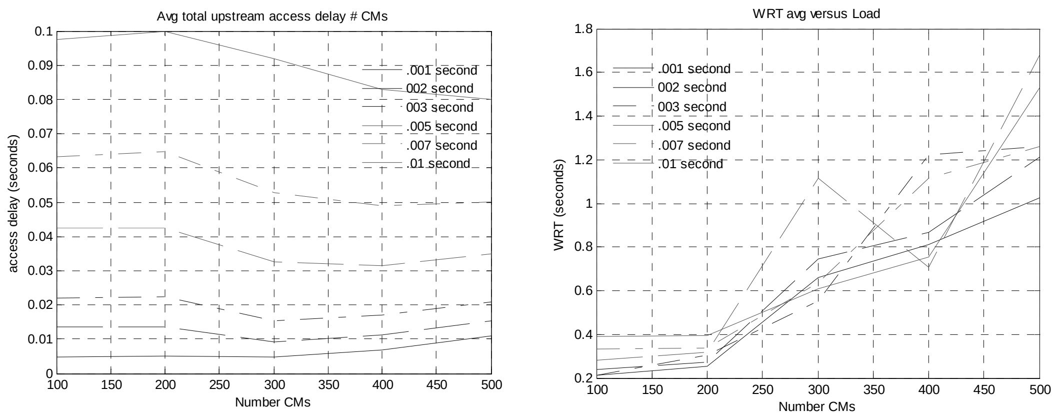


Figure 10. Channel utilizations



a. Percentage of concatenation b. Percentage of piggybacking

Figure 11. Type of upstream bandwidth request



a. Upstream access delay b. Web response times

Figure 12. Upstream performance measures

Comparing Figures 5,6 and 7 with Figures 9,10 and 12 shows dramatic differences in performance. We summarize the observed difference of the set 2 experiments relative to set 1.

- They exhibited much lower collision rate due primarily to higher levels of bandwidth allocated for contention requests.
- They exhibited a DS utilization of 100% because loss is not occurring.
- The US utilization is higher as a side effect of the increased DS efficiency.
- The access delay is more than an order of magnitude lower because of the reduced collision rate.
- The results are reflected in the web application level metric.

4.2 Analytic Model Framework

This section presents preliminary open queuing network models (QNMs) of the upstream contention and downstream DOCSIS transmission queues and their descriptions. It is intended to solve and verify these networks against the simulation program, initially using simplifying assumptions. Assumptions include exponential traffic, ignoring concatenation and piggybacking, modeling only BE service and assuming fixed length IP packets. In future work we plan on extending the QNMs to better capture the behavior of DOCSIS when subject to realistic Internet traffic.

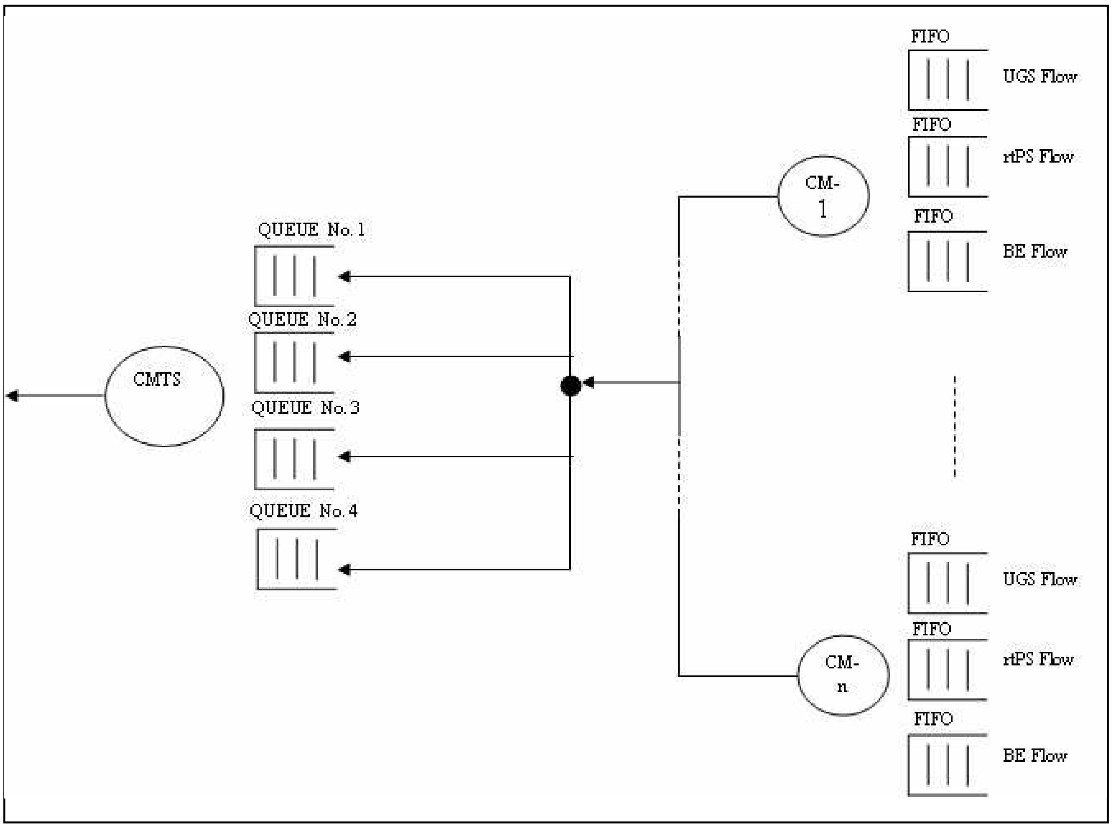


Figure 13. Preliminary Upstream QNM

Based on the upstream scheduling design and its underlying assumptions described in Section 4.1, Figure 13 shows a preliminary QNM of a DOCSIS upstream contention scenario. There are *n* CMs soliciting transmission from one CMTS. Each CM has 3 SID queues, one each for UGS, rtPS and BE flows, with priority decreasing from the UGS to BE. These queues are scheduled in order of priority.

In relation to the design and its assumptions given in Section 4.1, the CMTS has four queues. The first queue contains UGS bandwidth grants (virtual requests) and the second queue rtPS unsolicited polls, both operating under the EDF policy. These are priority queues and may each be modeled using the HOL queue arrangement where the priority parameter is the deadline of the job. rtPS bandwidth requests go to the third queue and BE bandwidth requests to the fourth and these latter two queues operate a FIFO policy. Like at the CMs, these queues are scheduled according to their priority with the first queue having greatest priority and the fourth the least. The priority scheduling mechanism at the CMTS does not carry out pre-emption.

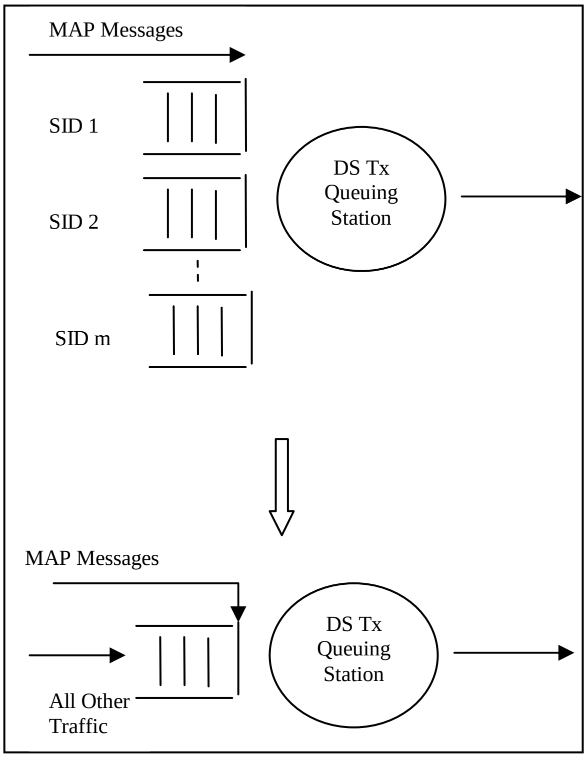


Figure 14. Preliminary Downstream QNM

Figure 14 shows a preliminary QNM of downstream transmission in DOCSIS (not subject to simplifying assumptions). Each SID flow is queued separately and is scheduled according to priority, where SID 1 has greatest priority from all m SID flows. MAP messages are treated with the highest priority from all downstream traffic. This may effectively be modeled as a HOL queuing station with ($m+1$) classes of traffic. MAP messages have greatest priority, followed sequentially by SID 1 up to SID m. In the case where downstream rate control is de-activated, the system may be modeled as a HOL queuing station with 2 classes of traffic: MAP messages having the greater priority and all other downstream traffic considered as the second traffic class, with each service flow treated with equal priority (implemented as FIFO).

5 Conclusions and Future Work

In this paper, we have presented the DOCSIS protocol, summarized basic performance, illustrated the behavior of DOCSIS using our '*ns*' simulation model and finally presented an initial queuing network model of a DOCSIS system. The simulation analysis presented in this paper shows that a DOCSIS system is complex. Finding an optimal set of configuration parameters is difficult. More fundamentally, reliably assessing DOCSIS performance is difficult as there are no standard industry performance benchmarks or methodologies.

The first step in our project has been to develop and validate a set of tools (simulation and analytic) that can be used for algorithm evaluation. By disseminating basic knowledge of the protocols as well as making our tools available, we hope to spark additional research in an effort to better engage the academic community in the rapid evolution of HFC networks. In future work we plan on developing dynamic algorithms that adapt system parameters towards optimal settings depending on system load and performance objectives. Further, we plan on continuing with the development of the queuing network models in an effort to provide a framework for analyzing complex cable networks.

References

[1]Cable Television Labs Inc. , CableLabs, "Data-Over Cable Service Interface Specifications- Radio Frequency Interface Specification", SP-RFIv2.0, available at http://www.cablemodem.com/specifications/specifications20.html.
[2]The Network Simulator. Available at : http://www-mash.cs.Berkeley.EDU/ns/.
[3]J. Martin, M. Westall, "Validating an 'ns' Simulation Model of the DOCSIS Protocol", under submission. Available at http://www.cs.clemson.edu/~jmarty/docsis.html .
[4]Archive of IEEE802.14 working group material, http://home.knology.net/ieee80214/
[5]O. Elloumi, et. Al., "A Simulation-based Study of TCP Dynamics over HFC Networks", Computer Networks, Vol. 32, No. 3, pp 301-317, 2000.
[6]R. Cohen, S. Ramanathan, "TCP for High Performance in Hybrid Fiber Coaxial Broad-band Access Networks", IEEE/ACM Transactions on Networking, Vol. 6, No. 1, February 1998.
[7]O. Elloumi, et. Al., "Improving Congestion Avoidance Algorithms in Asymmetric Networks", IEEE ICC 97, June 1997.
[8]K. Fall, S. Floyd, "Simulation-based Comparisons of Tahoe, Reno and SACK TCP", CCR, Vol 26, No. 3, July 1996.
[9]J. Hoe, "Improving the Startup Behavior of a Congestion Control Scheme for TCP", SIGCOMM 96, August 1996.
[10]Y Lin, C Huang, W Yin, 'Allocation and Scheduling Algorithms for IEEE 802.14 and MCNS in Hybrid Fiber Coaxial Networks', IEEE Transactions on Broadcasting, Vol. 44, No. 4, December 1998.
[11]V. Sdralia, et. Al., "Performance Characterisation of the MCNS DOCSIS 1.0 CATV Protocol with Prioritized First Come First Served Scheduling", IEEE Transactions on Broadcasting, Vol. 45, No. 2, June 1999, pp.196-205.
[12]Opnet simulation model, http://www.opnet.com/services
[13]E Morgan, D Greenstreet, "Voice over Cable (VoCable)", White Paper, Telogy Networks Inc., February 2000, Version 1.6, available at http://www.neural.uom.gr/Documents/Networks/Tech/Voice%20Over%20Cable.pdf
[14]W Yin, Y Lin, "Statistically Optimised Minislot Allocation for Initial and Collision Resolution in Hybrid Fiber Coaxial Networks", IEEE Journal on Selected Areas in Communications, Vol. 18, No. 9, September 2000.
[15]D Bushmitch, S Mukherjee, S Narayanan, M Ratty & Q Shi, "Supporting MPEG Video Transport on DOCSIS-Compliant Cable Networks", IEEE Journal on Selected Areas in Communications, Vol. 18, No. 9, September 2000.
[16]H. Balakrishnan, et. Al., "The Effects of Asymmetry on TCP Performance", ACM/IEEE International Conference on Mobile Computing and Networking, Sept. 1997.
[17]T. Lakshman, U. Madhow, B. Suter, "Window-based error recovery and flow control with a slow acknowledgement channel: a study of TCP/IP performance", INFOCOM97, April 1997.
[18]V Jacobson, "Compressing TCP/IP Headers for Low-Speed Serial Links", Feb 1990, RFC 1144.
[19]L. kalampoukas, A Varma, K. Ramakrishnan, "Improving TCP Throughput over Two-Way Asymmetric Links: Analysis and Solutions", SIGMETRICS 98, June 1998.
[20]M Tantaoui, K. Hua, T. Do, "BroadCatch: a Periodic Broadcast Technique for Heterogeneous Video-on-Demand", IEEE Transactions on Broadcasting, Sept. 2004.
[21]G. Muntean, P. Perry, L. Murphy, "A New Adaptive Multimedia Streaming System for All-IP Mutli-Service Networks", IEEE Transactions on Broadcasting, March 2004.
[22]H. Radha, M. van der Schaar, Y. Chen, "The MPEG-4 Fine-Grained Scalable Video Coding Method for Mutlimedia Streaming Over IP, IEEE Transactions on Multimedia, March 2001.
[23]L. Cheng, W. Zhang, L. Chen, "Rate-distortion Optimized Unequal Loss Protection for FGS Compressed Video", IEEE Transactions on Broadcasting, June 2004.
[24]A S Tanenbaum, "Computer Networks", Fourth Edition, Prentice Hall PTR, 2003.
[25]N. Abramson, "The Aloha System – Another Alternative for Computer Communications", Proceedings of the Fall Joint Computer Conference, Jan 1970.
[26]H. Balakrishnan, et. Al., "TCP Behavior of a Busy Internet Server: Analysis and Improvements", IEEE INFOCOM98, 1998.
[27]V. Paxson, "Measurements and Analysis of end-to-end Internet Dynamics", Ph.D. dissertation, Univ. California, Berkeley, CA, 1997.
[28]J. Martin, A. Nilsson, I. Rhee, "Delay-based Congestion Avoidance for TCP", IEEE/ACM Transactions on Networking, June 2003.
[29]J. Stankovic, M. Spuri, K. Ramamritham, G.Buttazzo, "Deadline Scheduling for Real-time Systems – EDF and Related Algorithms", Kluwer Academic Publishers, 1998.

[30]J. Martin, N. Shrivastav, “Modeling the DOCSIS 1.1/2.0 MAC Protocol”, Proceedings of the 2003 International Conference on Computer Communications and Networks”, Dallas TX, October 2003.
[31]J.Martin, “The Interaction Between the DOCISS 1.1/2.0 MAC Protocol and TCP Application Performance”, Proceedings of the International Working Conference on Performance Modeling and Evaluation of Heterogeneous Networks, (Ikley, UK, July, 2004), pp. P57/1-10.
[32]A. Feldmann, et. Al., “Dynamics of IP Traffic: A study of the role of variability and the impact of control”, SIGCOM99.
[33]S. Saroiu, P. Gummadi, S. Gribble, “A Measurement Study of Peer-to-Peer File Sharing Systems”, Multimedia Computing and Networking (MMCN), Jan 2002.
[34]Charter Communications, http://www.charter.com/products/highspeed/highspeed.aspx
[35]L. Zhenglin, X. Chongyang, “An Analytical Model for the Performance of the DOCSIS CATV Network”, The Computer Journal, Vol45, No3, 2002, pp.278-284.
[36]H Ju, W Liao, “Adaptive Scheduling in DOCSIS-based CATV Networks”, IEEE, Computer Communications and Networks, 2002. Conference Proceedings. Page(s):543 – 547.