Validating an *‘ns’* Simulation Model of the DOCSIS Protocol

Jim Martin, Mike Westall
Department of Computer Science
Clemson University
{jim.martin,westall}@cs.clemson.edu

Keywords: Simulation, broadband access, HFC cable networks, TCP performance

ABSTRACT

The number of households and businesses using HFC cable networks for Internet access is rapidly approaching 40million in the United States. The cable industry has standardized on a single MAC and physical layer standard, the Data Over Cable Service Interface Specification (DOCSIS). The emerging IEEE 802.16 broadband wireless access standard is based on the DOCSIS MAC layer. We have implemented a simulation model of the DOCSIS MAC using the *‘ns’* simulation package. In this paper we offer analytic and live network evidence that the simulation model accurately reflects the behavior of a DOCSIS network.

1. INTRODUCTION

The increasing use of broadband DSL and cable access networks is a major driver of the continuing robust growth of the Internet. According to a PEW survey, nearly 65% of adults in the United States access the Internet at least once each day [1]. The study also indicates that there are 66 million households equipped with broadband access, and that approximately 54% of these households use cable.

Beginning in 1998, the cable industry has developed a set of standards, collectively referred to as Data-Over-Cable Service Interface Specification (DOCSIS), for supporting data over HFC cable networks [2-3]. The DOCSIS standards cover the physical, MAC layers, security, operations system support (OSS), equipment interfaces, and equipment validation. The technology driving Hybrid Fiber Coaxial (HFC) cable networks continues to advance at breathtaking speed. Sophisticated modulation techniques along with channel bonding will increase data rates from the current tens of Mbps to hundreds of Mbps in both the upstream and downstream directions.

Our research in DOCSIS-based shared medium networks is motivated by the observation that physical layer capabilities are advancing at a faster pace than advances in the MAC protocol. The current DOCSIS protocol is an optimized version of slotted Aloha with reservations proposed by Roberts in 1973 [4]. Additional mechanisms such as Unsolicited Grant Service (UGS), piggybacking, and concatenation enhance performance and the ability to provide QoS guarantees. Our research agenda is to explore the continued evolution of DOCSIS to better support current and next generation shared medium networks. Aspects of this research are applicable to 802.16 (WiMAX) networks as its channel allocation mechanism is also DOCSIS-based, but the focus of this paper is HFC networks.

DOCSIS systems are extremely complex. The DOCSIS HFC cable specification is a 500 page document. Due to the complexity and cost, there are no open source DOCSIS platforms that are available to researchers. In contrast to the Internet community, where academic researchers can introduce new protocols or protocol enhancements through the IETF’s RFC process, in the industry-centric HFC cable and WiMAX communities, standards are developed in members-only industry consortiums. As a result, the evolution of DOCSIS is being directed by industry with little involvement of academia. To address this we have developed a simulation model of the DOCSIS MAC protocol for the popular *‘ns’* simulator tool [5].

In previous work, we presented a preliminary *‘ns’* DOCSIS simulation model and showed that certain system parameters can significantly impact performance and dynamics of a DOCSIS system [6-7]. The observation that research based solely on simulation lacks credibility [8] motivated our efforts to validate the simulation via analytic means and studies of live DOCSIS networks. Recent publications note that the wireless research is particularly sensitive to physical layer assumptions [9-12]. The authors of [10] indicate that the choice of appropriate experimental scenarios is just a crucial as the use of valid channel models. The validation presented in this paper addresses the concerns to some degree in HFC (but not wireless) environments by grounding the simulation model with data obtained from live cable network systems.

Validating a model of a DOCSIS system is difficult for a number of reasons. First the specification leaves significant room for implementation decisions. Second, the behavior of the system is very sensitive to parameter selection. Third, studies of the physical layer characteristics of HFC channels have not been published. In the validation presented in this paper, we address the first two issues and leave the physical model development and validation for future work. The focus of the validation is therefore on the operation of the MAC protocol, and on how our simulated DOCSIS MAC layer operates over a modified version of the existing *‘ns’* LAN physical link.

As a first step in the validation of the `ns’ simulation we developed a set of simple analytic models that confirm basic upstream and downstream behavior. To ensure that implementation decisions we made in the DOCSIS simulation model produced correct behavior under more complex (and thus analytically intractable) workloads, we compared the behavior predicted by the simulation with the behavior measured on two live DOCSIS networks: a private laboratory network that we operate; and a public network operated by Charter Communications.

The remainder of this paper is organized as follows. After a brief background on DOCSIS, we present the operation and features of the DOCSIS model. We summarize an analytic model that captures the upstream behavior of DOCSIS and use it to demonstrate the simulation model works as intended in a single-flow, always-on scenario. Next we describe the analysis involving live cable networks. We highlight related work and then end the paper with conclusions and future work.

2. BACKGROUND

The cable industry's research consortium, CableLabs [3], has developed the Data Over Cable Service Interface Specification (DOCSIS) which defines the physical layer and MAC layer protocols for data over HFC cable networks [2]. A cable operator operates the cable modem termination system (CMTS) units that interact with cable modems (CMs) deployed at subscribers' locations. A modern CMTS houses multiple 'blades' with each blade supporting one or more HFC domains (one downstream channel with four upstream channels). Six MHz (or greater) bandwidth is allocated from the 88-860 MHz spectrum for each downstream channel, and upstream channels are allocated from the 5 – 52 MHz frequency range.

In the downstream direction, a single sender (the CMTS) transmits to a set of CMs using a data rate ranging from 10 Mbps to 50 Mbps. IP packets sent downstream are divided into 188-byte MPEG frames. Each CM has a unique MAC address and will receive only frames that are addressed to its MAC address or to the broadcast address. Figure 1 illustrates the basic DOCSIS network model. Computers owned by the subscriber connect to the CM through an Ethernet or USB interface. In the upstream direction multiple senders (CMs) share a channel that supports data rates in the range of 5 Mbps to 10 Mbps. The upstream transmission model is shared access using time division multiple access (TDMA) with a random access contention-based reservation mechanism. IP packets that are sent upstream are encapsulated in a DOCSIS frame and transmitted during assigned slots. If a packet does not fit into the number of contiguous slots that were allocated it is fragmented into multiple frames.

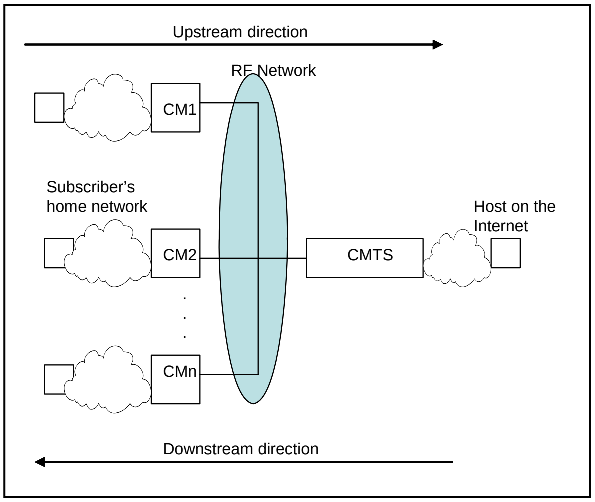


Figure 1. Network diagram

Traffic from CMs is classified by service flow. For example, a configuration that supports telephony and best effort data typically has 4 service flows: two for the upstream and downstream VoIP traffic and two for the upstream and downstream best effort traffic. DOCSIS maps service flows to one of several ATM-like services including best effort, unsolicited grant service (UGS), which is equivalent to ATM's constant bit rate service, and non-realtime polling (nrtPS), which is equivalent to ATM's non-realtime variable bit rate service. As in ATM, different performance guarantees are available for each service class. For UGS service the CMTS periodically provides unsolicited grants to the CM, for nrtPS the CMTS periodically asks the CM if it needs bandwidth, and for best effort bandwidth is allocated on-demand using a contention-based request mechanism.

All CMs receive periodic MAP messages from the CMTS over the downstream channel. The MAP message defines the assignment of upstream slots during the next time increment referred to as the MAP time. MAP messages must describe slots far enough in the future so that CMs have sufficient time to prepare for upstream transmission allocations. Typical values for the MAP time range from 1 to 10 milliseconds. Some implementations might keep the MAP time constant, other implementations might dynamically change the MAP time. Upstream slot categories include: data grants to specific CMs; management related slots; and contention slots in which CMs may request data grants for upstream traffic. When a CM needs an upstream grant, it selects a random number within a backoff window which is determined by a backoff range value maintained by the CM. After a CM transmits the request, if the next MAP does not contain a grant or a grant pending from the CMTS, the CM assumes a collision has occurred and doubles the size of the backoff window. The contention request cycle continues until it succeeds or it has tried a total of 16 times in which case the packet is dropped. A CM can request bandwidth sufficient to transport multiple IP packets in a single DOCSIS frame by issuing a concatenated bandwidth request. To further reduce contention, a CM is permitted to piggyback a request for bandwidth on an upstream data frame.

3. THE SIMULATION MODEL

The simulation model implements the DOCSIS architecture defined in [2] with the following restrictions: 1) CMs are limited to a single default best effort service flow and a single UGS or nrtPS 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.

Implementation aspects of the model are summarized as follows:

- To account for forward error correction (FEC) overhead in the physical layer, we reduce the upstream channel capacity by 8% and by 4.9% in the downstream channel. This approximation was suggested by Cisco [13].
- The contention backoff range that is sent to each CM in downstream MAP messages is statically set with a configuration parameter.
- Both concatenation and piggybacking can be enabled or disabled. If concatenation is enabled, a configuration parameter specifies the maximum number of IP packets that

can be inserted into a frame. If both piggybacking and concatenation are active, concatenation has priority when both are possible.

- The bandwidth scheduler runs at the CMTS node. It executes on every tick of a timer that is set to the MAP time frequency. The algorithm examines all existing requests for bandwidth and implements an earliest deadline first scheduling policy. All UGS service requests are scheduled first and remaining bandwidth is shared among best effort requests on a first-come-first-served basis. The scheduler supports dynamic MAP times by allowing a MAP to specify grants up to a configured maximum amount (known as the MAP lookahead). The scheduler will only do this if it can meet all QoS requirements. The scheduler by default allocates all unused slots for contention requests, but this behavior can be changed so that a fixed number of contention request slots are allocated each MAP time.

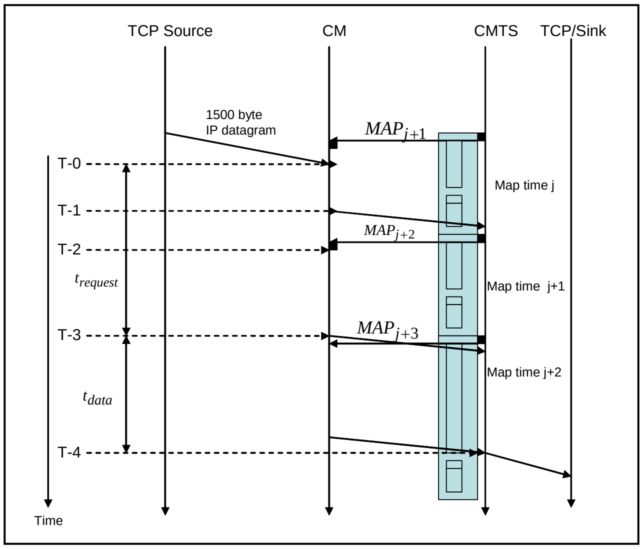


Figure 2. Upstream transmission scenario with variable MAP times

Figure 2 illustrates the upstream transmission of a 1500 byte IP datagram from a TCP source to a sink located outside the HFC network. Time progresses in the downwards direction. The small dark square box positioned at the beginning of each MAP time represents the transmission of the MAP message in the downstream direction. The CMTS (in our model) sends a MAP message at the beginning of each MAP time. Each MAP message describes the slot assignments for the next MAP time. The illustration assumes that the MAP time is allowed to increase dynamically to accommodate the transmission of the IP packet in a single frame. If this is not permitted then the CMTS would issue a partial grant. The CM would piggyback additional requests for bandwidth with each data fragment that is sent.

An IP packet arrives at the CM during the j-th MAP at time T-0. The CM sends a bandwidth request message at time T-1 (in a contention request slot) and receives the data grant at time T-2. The grant is located in the third MAP time. The CM begins to transmit the frame at time T-3 and the last bit of the frame is received by the CMTS at time T-4. The time between T-3 and T-0 is the access delay which represents the total time a packet is delayed over the DOCSIS network not including transmission and propagation time experienced by the data packet (we refer to this delay as $t_{request}$). Packets that arrive at a CM from the subscriber's network will be queued until their assigned transmission time. The size of the upstream CM queue is a configuration parameter. In prior work we showed that performance can deteriorate as the MAP time grows [6-7]. The contention request allocation strategy and the use of fragmentation, concatenation and piggybacking also have significant impact on system performance.

4. UPSTREAM ANALYTIC MODEL

The upstream behavior of DOCSIS is similar to slotted aloha with reservations[4,14-15]. Following the method presented in [16], we define the maximum upstream application throughput, $T_{\max us}$, to be $\frac{D_{perCycle}}{t_{data}+t_{request}}$ where $D_{perCycle}$ is the amount of user data sent upstream in one reservation request cycle, t_{data} is the upstream transmission and propagation time of the data, and $t_{request}$ is the delay associated with the request process. In this section, we consider an analytic model that computes the upstream throughput that could be obtained by a single application flow. The model does not consider loss or collision that might be caused by competing traffic. While the single-flow assumption is clearly unrealistic, it allows us to compare basic operation of the MAC analytic and simulation models to live DOCSIS networks under controlled conditions. For brevity, we only consider the backlogged case where a single CM always has 1500 byte IP packets to send. Figure 2 illustrates the $t_{request}$ and the t_{data} delays in an upstream operation. The $t_{request}$ represents the total delay experienced by the packet from its arrival at the CM until its first bit is transmitted upstream. The t_{data} is the transmission and the propagation time of the upstream data frame.

For brevity, we omit the details of the analytic model and focus upon its use in validating the simulation. Refer to [17] for a complete description of the analytic model. The model incorporates basic configuration information such as the number of slots in a MAP, the number of management slots and the number of contention request slots per MAP, the backoff range, and the stochastic nature of the backoff process. The model optionally accounts for piggybacking and concatenation. In this section we consider results from the analytic model with piggybacking and concatenation disabled.

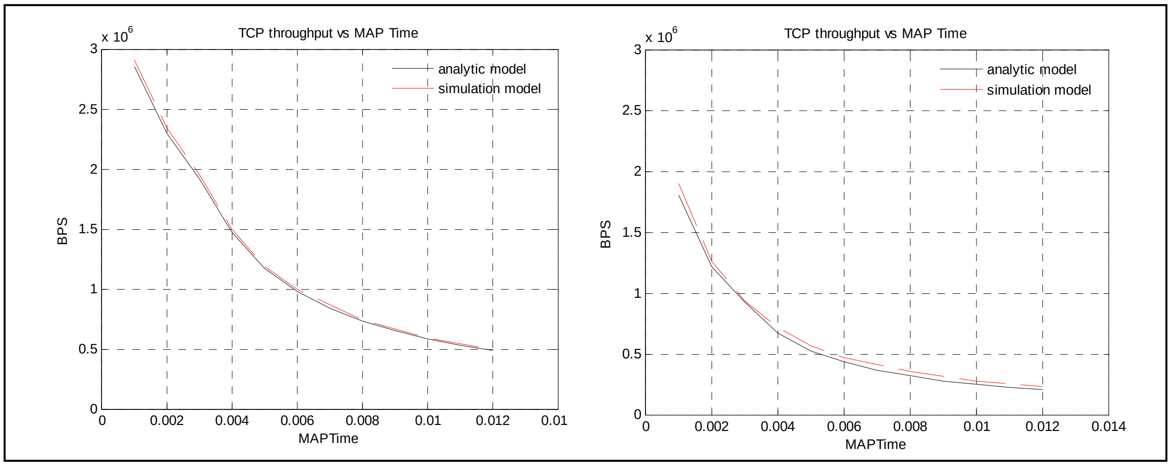
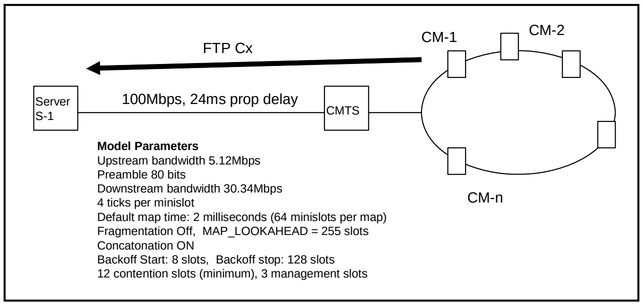


Figure 3a. backOffDelay = 8 Figure 3b. backOffDelay=64

Figure 3. Model results with fixed number of contention slots

Figures 3a and 3b show TCP throughput predicted by the analytic model and throughput observed in a simple simulation experiment. The simulated network is illustrated in Figure 4. One FTP flow is active between CM-1 and the server S-1. The maximum TCP window configuration setting was 11 packets which was sufficient to keep the upstream transmission queue at CM-1 always filled with data but to avoid queue overflow. We disabled piggybacking and concatenation in the simulation model so that it matches the assumptions made in the analytic model. We performed 9 runs varying the MAP time parameter from a value of 0.001 seconds to a maximum value of 0.012 seconds. Each run lasted 1000 simulated seconds. For each run we obtained the mean TCP throughput over the lifetime of the simulated connection.



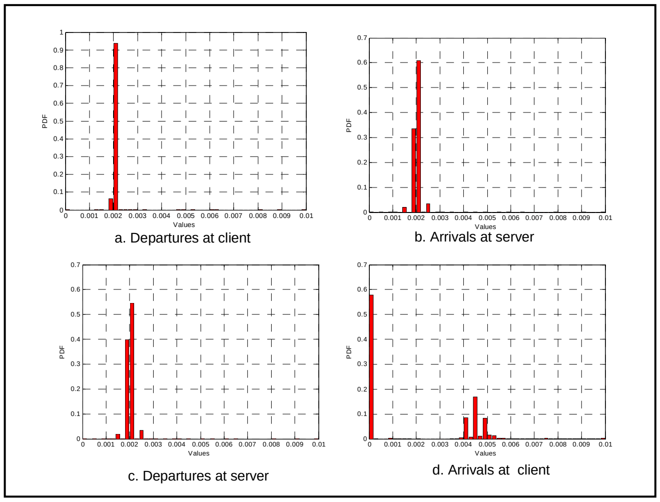
Model Parameters
Upstream bandwidth 5.12Mbps
Preamble 80 bits
Downstream bandwidth 30.34Mbps
4 ticks per minislot
Default map time: 2 milliseconds (64 minislots per map)
Fragmentation Off, MAP_LOOKAHEAD = 255 slots
Concatonation ON
Backoff Start: 8 slots, Backoff stop: 128 slots
12 contention slots (minimum), 3 management slots

Figure 4. Simulation network model and parameters

The first set of results, illustrated in Figure 3a, used a *backoffRange* of 8 slots. This was increased to 64 slots for the second set of results illustrated in Figure 3b. The throughput obtained by the simulation closely tracks the throughput predicted by the analytic model in both cases (the error never exceeded 10%). Two results are evident. First, Figure 3 shows that upstream TCP throughput deteriorates as the MAP time increases. Increasing the MAP time increases the $t_{request}$ delay which decreases TCP throughput. Second, the results show that increasing the *backoffRange* reduces throughput. Intuitively this makes sense since a higher *backoffRange* also increases the $t_{request}$ delay resulting in a lower throughput.

5. NETWORK MODEL VALIDATION

To further validate the simulation model, we compare simulation results with observations from two live cable networks. The first study was conducted on a private DOCSIS testbed located in our research lab. The second study employed a lab computer located at Clemson University and a home computer connected to the Internet via Charter Communication's public DOCSIS network.



a. Departures at client

b. Arrivals at server

c. Departures at server

d. Arrivals at client

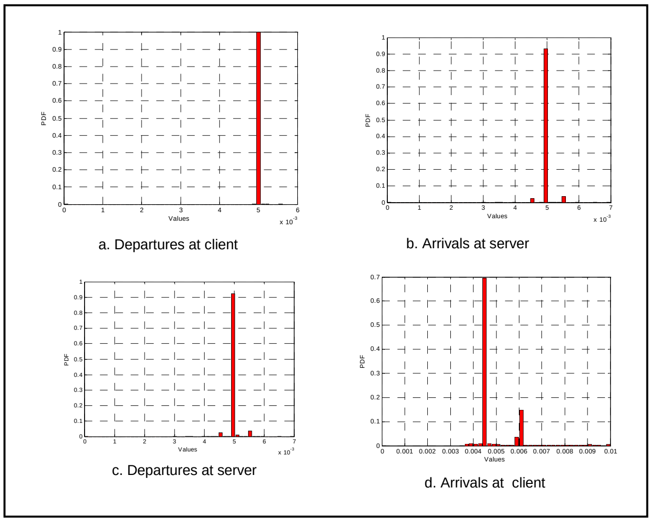
Figure 5. DOCSIS testbed observed departure and arrival distributions (2 ms packet spacing)

5.1 Network 1: DOCSIS testbed

The testbed consists of four CMs (two Cisco uBR905s and two Motorola SurfBoards), a Cisco uBR7110 CMTS, five Linux-based PCs, and a simple RF plant. The downstream service rates on the DOCSIS network were set to 10 Mbps and the upstream service rates were 1 Mbps. The CMTS is connected to a private 100 Mbps Ethernet which hosted the client system. We conducted experiments using a UDP echo application between a Linux computer (the server) connected to one of the CMs and a second Linux machine (the client) connected to the Fast Ethernet network at the CMTS. The client sent a periodic stream of small (64 byte) UDP packets downstream to the server which echoed the packets back. We obtained a *tcpdump* trace at both the client and the server [18].

Figures 5a through 5d visualize both one-way streams at the respective sender and receiver sides when the client sends a packet every 0.002 seconds. The figures plot the distribution of the interpacket departure or arrival times. Figure 5a indicates that roughly 5% of the samples are in the 0.0018 to 0.002 second range. We assume timing inaccuracies at the client cause this behavior. Figure 5b shows the level of jitter increased as the packets traveled in the downstream direction. As there was no other traffic on either the wired or the RF networks, we assume the jitter was caused by processing delay either at the CMTS, the CM, or at the server.

Figure 5c illustrates that the UDP packet interdeparture times at the server (destined for the client over the upstream channel) ranged from .0015 seconds to .0025 seconds. The results suggest that about 55% of packets arrived at the CM when there was at least one packet already waiting for transmission. The CM sends these packets in a single, concatenated transmission. The remaining 45% of the UDP packets arrived at the CM when no other packets were queued. We conjecture that these packets arrived shortly after the CM has just sent. Therefore these packets suffer a lengthy delay of two MAP times plus a small amount of delay that depends on the accumulated jitter along with scheduling delays over the network. Based on the modes in the .004 through .005 second range of Figure 5c, we conjecture that the MAP time is 0.002 seconds. As we are the administrators of the CMTS, we know that the CMTS has been configured to use concatenation and that the MAP time is in fact 0.002 seconds. Figure 6 illustrates the interpacket departures and arrivals when the client sends a UDP packet every 0.005 seconds. Figure 6d shows the two modes at 0.0045 and 0.006 seconds which is what we would expect for a system using a MAP time of 0.002 seconds.



a. Departures at client

b. Arrivals at server

c. Departures at server

d. Arrivals at client

Figure 6 DOCSIS testbed observed departure and arrival distributions (5 ms packet spacing)

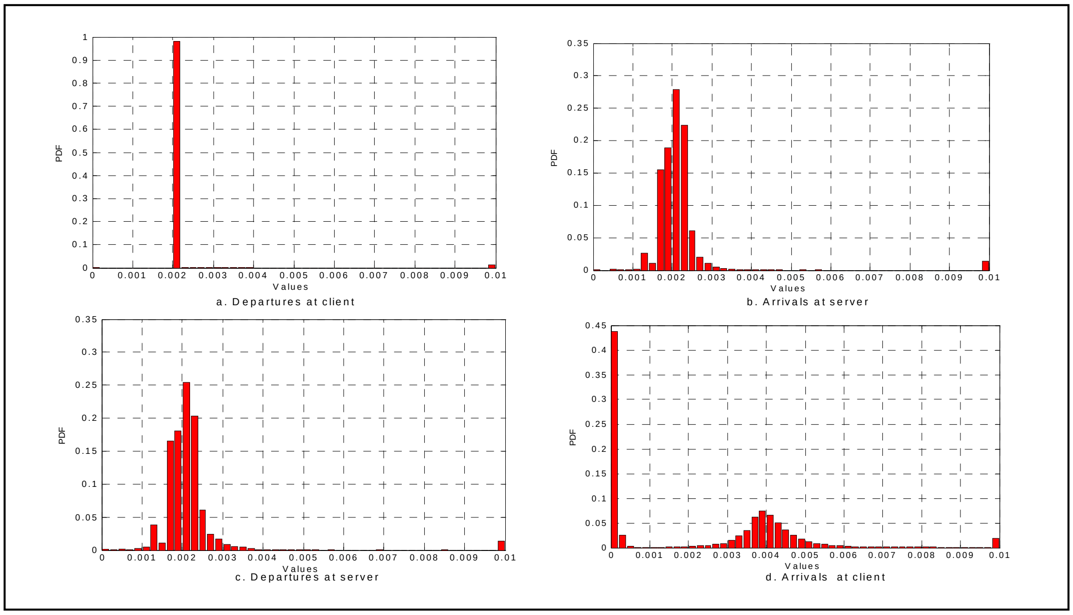
In previous work we concluded that the bottleneck in a DOCSIS system is the upstream channel and in particular its ability to transport packets at a high rate of speed. This upstream packet rate limitation impacts both downstream and upstream throughput. In the downstream direction, TCP throughput is limited by the rate at which TCP ACK packets can be sent over the upstream channel. For a MAP time of 0.002 seconds and a TCP/IP packet size of 1500 bytes, we found that the maximum application level downstream throughput has been shown to be limited to less than 6 Mbps. To confirm this we conducted a simple experiment using the testbed. We enabled a single TCP flow in the downstream direction and observed a maximum TCP throughput of 3.6 Mbps. We configured the identical network using our simulation model and also saw a maximum downstream throughput of 3.6 Mbps. Only by hand-tuning system parameters, in particular disabling piggybacking and forcing DOCSIS to concatenate large number of ACK packets in a single upstream frame, could we approach a downstream throughput of 10 Mbps. Unfortunately concatenation can significantly impact TCP dynamics by perturbing the TCP ACK spacing which has been shown to possibly lead to higher loss rates [19-20].

5.2 Network 2: Charter Communications

We repeated the experiments conducted in the DOCSIS testbed between a Linux computer (the client) located in a campus lab and a second Linux computer (the server) located at a residential network. The server connected to the Internet by Charter’s Internet access service which provided a 5 Mbps downstream rate and a 512 Kbps upstream rate. The interconnect between Charter and Clemson involved multiple backbone ISPs. The client sent a periodic stream of small (64 byte) UDP packets to the server which echoed the packets back. We obtained a tcpdump trace at both the client and the server.

Figures 7a through 7d show the distribution of the interpacket departure (or arrival) times of both one-way streams at the respective sender and receiver sides. The client sends a packet every 2 milliseconds. Figure 7a shows that approximately 97% of the interpacket departure times are within 200 microseconds (the bin size) of their expected value. The jitter is attributed to random delays that occur in the operating system. For the purposes of this experiment, the jitter associated with the stream generated by the client is acceptable. Figure 7b shows that packets traveling over the path were subject to delay adding significant jitter to the stream. It is not possible to determine where the packet delay actually occurs.

Figure 7c shows minor additional distortion caused by processing overhead at the Linux server. Figure 7d shows the impact of DOCSIS on the upstream UDP flow. The upstream bandwidth consumed was 275 Kbps (accounting for headers) which would not overload the upstream channel. We confirmed this by verifying that minimal loss occurs. Based on our understanding of DOCSIS (which we validated in the previous DOCSIS testbed experiments), Figure 7d suggests that Charter has configured its network for a MAP time of 2 milliseconds. The mode of 0.004 seconds represents the two MAP times that are required to send upstream data. This is true even if piggybacking is used. The large mode at 0 seconds indicates that about 50% of the echo packets are being sent back-to-back over the upstream channel in



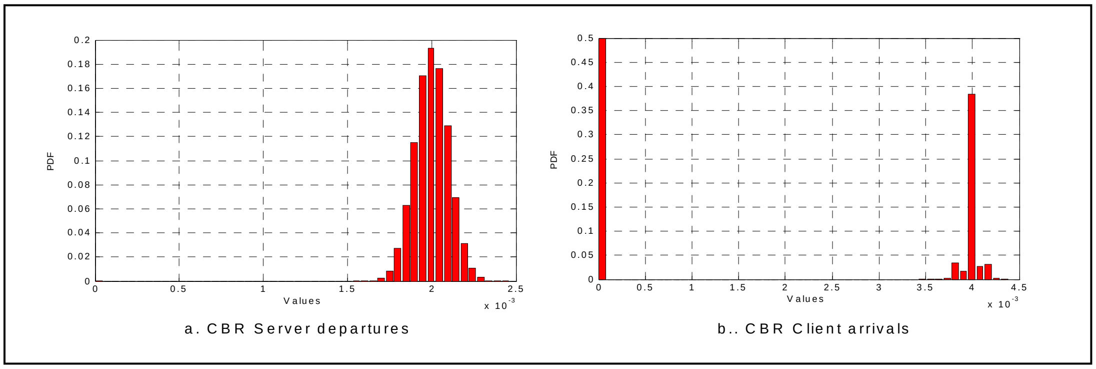
a. Departures at client

b. Arrivals at server

c. Departures at server

d. Arrivals at client

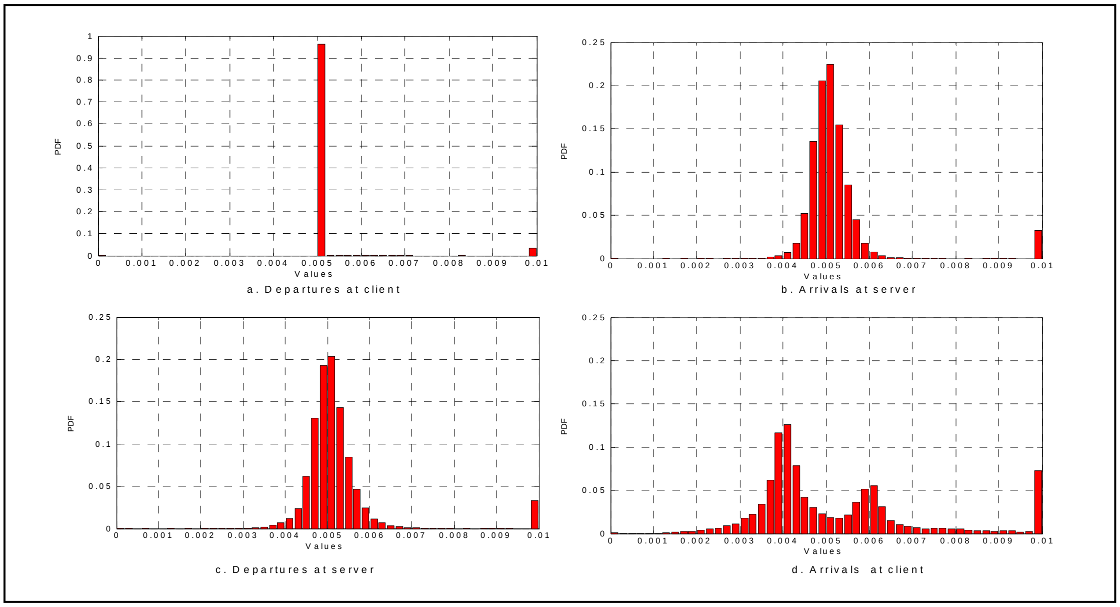
Figure 7. Cable network observed departure and arrival distributions (2 ms packet spacing)



a. CBR Server departures

b.. CBR Client arrivals

Figure 8. Simulation results (2 ms packet spacing)



a. Departures at client

b. Arrivals at server

c. Departures at server

d. Arrivals at client

Figure 9. Cable network observed departure and arrival distributions (5 ms packet spacing)

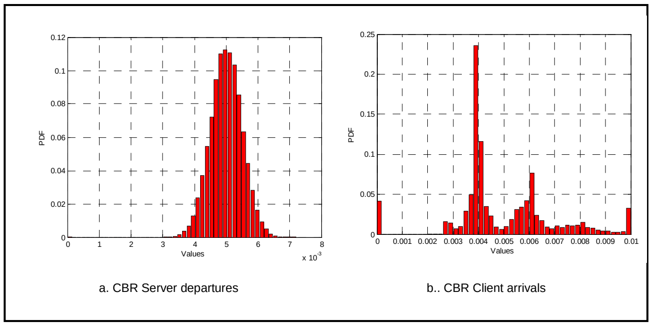
a concatenated frame. Concatenated packets arrive at the client separated by the transmission time of the bottleneck link over the path between the CMTS and the client (which we estimate to be 45 Mbps). Figures 8a and 8b illustrate comparable simulation results. We configured the CM-1 node (shown in Figure 4) with a CBR traffic source that sends a 64 byte packet every 0.002 seconds. Therefore we simulated only the echo reply stream between the Linux machine connected to the Charter network and the Linux machine located at Clemson University. We enabled piggybacking and concatenation. To model the random delay observed in Charter's network associated with UDP echo packets that arrive at the CM for upstream transmission (i.e., Figure 7c), we added an artificial delay to each packet before transmitting at the server. Based on the Anderson-Darling goodness-of-fit test the distribution illustrated in Figure 7c is neither normal nor Weibull (with 44000 samples and a 95% level of confidence the AD statistics was an order of magnitude greater than the critical value). As a coarse approximation of the distribution we used a delay that is based on a normal distribution with a mean of 0 and a standard deviation of 0.0001. Figure 8a plots the interpacket departure time distribution from the CBR source and Figure 8b plots the interpacket arrival distribution at the UDP sink (S-1). Figure 8b shows the same bimodal result observed in Figure 7d. From this, we deduce that the Charter network uses a MAP time of 0.002 seconds and that it does support concatenation.

We next subjected the server in the Charter network to a stream of periodic UDP packets sent every 5 milliseconds. Figures 9a through 9d show the packet arrival and departure time distributions for the echo request and reply streams. Figure 9d again suggests the network is configured with a MAP time setting of 0.002 as some number of UDP packets are sent 0.002 milliseconds after the main mode of 0.004 seconds. On an unloaded system, it takes one MAP time for the piggybacked request to arrive at the CMTS and then the subsequent grant to arrive at the CM. Figures 10a and 10b show the results of a comparable simulation run. The CBR source is configured to add an artificial jitter based on a normal distribution with a mean of 0 and a standard deviation of 0.0005. To better fit the distribution illustrated in Figure 9d, we added 200 competing CMs that generated realistic amounts of Web traffic (primarily in the downstream direction) to the simulation. Comparing Figure 10b with 9d shows that the simulation model behaves in a reasonable manner. Without the competing CM traffic the spread around the modes in Figure 10b was less than 200 microseconds.

To determine if Charter's network uses piggybacking, we repeated the simulation associated with Figures 10a and 10b but with piggybacking disabled. With piggybacking, as illustrated in Figure 10b, we see roughly 4% of packet interarrival times were back-to-back implying a small amount of concatenation was occurring. When we repeat the simulation experiment without piggybacking we saw the level of concatenation grow much larger (17%). We conclude that Charter's network does use piggybacking because if it did not we would see a higher level of concatenation.

6. RELATED WORK

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[21]. 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 cable industry supported the DOCSIS standard. However, because DOCSIS is based on 802.14, prior analysis of the IEEE 802.14 standard is relevent. In [22], the authors found that TCP throughput over an 802.14-based HFC network is low primarily due to ACK compression. While assumptions made by the authors (such as high loss rates in the upstream path) are no longer true, our recent results do confirm that DOCSIS induces ACK compression. Other early studies looked at TCP enhancements when operating over HFC networks suffering from high channel error rates and the 802.14 contention protocol [23-25]. More recent analysis has been performed using an Opnet model of DOCSIS[26]. Most of these studies were limited in scope or are now simply outdated. Further, the Opnet DOCSIS model source code is not freely available limiting its use to capacity planning rather than protocol evolution. In order for the research community to participate in the advancement of DOCSIS-based broadband access networks, a fully functional, validated and documented simulation model is required.



a. CBR Server departures

b.. CBR Client arrivals

Figure 10 Simulation results (5 ms packet spacing)

7. CONCLUSIONS AND FUTURE WORK

We have presented the design and validation of an *'ns'* DOCSIS simulation model. Using analytic models, we have provided evidence that the simulation model behaves as expected. The analytic model is based on our understanding of the DOCSIS protocol. Comparing results obtained from it to results obtained from the simulation model only validates that we have implemented the DOCSIS protocol as we understand it correctly. To validate the model for correctness we relied on the analysis of live DOCSIS systems. Based on measurement analysis involving a DOCSIS testbed and involving a home network connected to the Internet through Charter's Communications, we provided empirical evidence that our implementation, especially with respect to the MAP time duration and the usage of concatenation and piggybacking, is correct. The validation that we have presented is clearly not complete. The analytic model analysis was limited to single flow scenarios. The live network measurement served only to validate basic operation. In future work we will validate the simulation model's collision recovery and QoS support.

Our DOCSIS testbed analysis, which had service rates twice that of the Charter network, illustrates that the performance observed over a DOCSIS network is likely to be determined by the behavior of the MAC protocol rather than on service rates. This motivates our continued research towards improving the MAC layer in shared medium broadband access networks. By making a validated simulation model available to the research community, we hope that others in academia also contribute to the evolution of DOCSIS broadband access networks.

8. REFERENCES

[1] J. Horrigan, "Broadband Adoption in the United States", PEW internet report, sept. 2005, Available at http://www.pewinternet.org/pdfs/PIP_Broadband.TPRC_Sept05.pdf

[2] Cable Television Labs Inc. , CableLabs, "Data-Over Cable Service Interface Specifications- Radio Frequency Interface Specification", SP-RFIv2.0, available at http://www.cable-modem.com/ specifications/ specifications20.html.

[3] CableLabs, www.cablelabs.com

[4] L. Roberts, "Dynamic Allocation of Satellite Capacity Through Packet Reservation", *Proc. National Computer Conference (NCC '73)*, June 1973.

[5] "The Network Simulator –ns2", UCB/LBNL/VINT http://www.isi.edu/nsnam/ns/

[6] J. Martin, "Modeling the DOCSIS 1.1/2.0 MAC Protocol", *Proceedings of the IEEE International Performance, Computing, and Communications Conference (ICCCN03),* Oct. 2003.

[7] J. Martin, "The Impact of the DOCSIS 1.1/2.0 MAC Protocol on TCP", *Proceedings of the IEEE Consumer Communications and Networking Conference,* Jan. 2005.

[8] K. Pawlikowski, H. Jeong, J. Lee, "On Credibility of Simulation Studies of Telecommunications Networks", *IEEE Communications*, Jan. 2002.

[9] S. Floyd, E. Kohler, "Internet Research Needs Better Models", *Proceedings of Workshop on Hot Topics in Networks (HotNets),* Oct. 2002.

[10] A. Gurtov, S. Floyd, "Modeling Wireless Links for Transport Protocols", ACM *SIGCOMM Computer Communication Review*, V. 34, No. 24, April 2004, pp.85-96.

[11] J. Heidemann, et al., "Effects of Detail in Wireless Network Simulation", *Proceedings of the SCS Multiconference on Distributed Simulation*, Jan. 2001, pp. 3-11.

[12] D. Kotz, C. Newport, C. Elliott, "The Mistaken Axioms of Wireless-Network Research", D. Kotz, C. Newport, and C. Elliott, Dartmouth College Computer Science, Tech. Rep. TR2003-467, July 2003.

[13] Cisco white paper, "Understanding Data Throughput in a DOCSIS World", available at http://www.cisco.com-/warp/public/109/data_thruput_docsis_world_19220.pdf

[14] N. Abramson, "The Aloha System – Another Alternative for Computer Communications", *Proceedings of the Fall Joint Computer Conference*, Jan. 1970.

[15] S. Lam, L. Kleinrock, "Packet Switching in Multiaccess Broadcast Channel: Dynamic Control Procedures", *IEEE Transactions on Communications*, 23(9), 891-904, Sept. 1975.

[16] D. Bertsekas, R. Gallager, *Data Networks*, Prentice-Hall, 1992.

[17] J. Martin, M. Westall, "Validating an *'ns'* Simulation Model of the DOCSIS Protocol – extended version", available at http://www.clemson.edu/~jmarty/docsis.html

[18] *tcpdump* program available at http://www.tcpdump.org

[19] H. Balakrishnan, et al., "TCP Behavior of a Busy Internet Server: Analysis and Improvements", *IEEE Infocom*, July 1998.

[20] V. Paxson, "Measurements and Analysis of end-to-end Internet Dynamics", Ph.D. dissertation, Univ. California, Berkeley, CA, 1997.

[21] Archive of IEEE802.14 working group material, http://home.knology.net/ieee80214/

[22] O. Elloumi, et al., "A Simulation-based Study of TCP Dynamics over HFC Networks", *Computer Networks*, Vol. 32, No. 3, pp 301-317, 2000.

[23] 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.

[24] N. Golmie, Y. Saintillan, D. Su, "A Review of Contention Resolution Algorithms for IEEE 802.14 Networks", *IEEE Communication Surveys*, 1Q99.

[25] D. Sala, J. Limb, S. Khaunte, "Adaptive Control Mechanism for Cable Modem MAC Protocols", *IEEE Infocom*, July 1998.

[26] G.Chandrasekaran, M. Hawa, D. Petr, "Preliminary Performance Evaluation of QoS in DOCSIS1.1", ITTC-FY2003-TR-22736-01, Information and Telecommunication Technology Center, University of Kansas, Jan. 2003.