The Interaction Between the DOCSIS 1.1/2.0 MAC Protocol and TCP Application Performance

Jim Martin
jim.martin@cs.clemson.edu
Department of Computer Science
Clemson University

Abstract- The deployment of data-over-cable broadband Internet access continues to unfold throughout the world. While there are competing technologies, the Data over Cable (DOCSIS) 1.1/2.0 effort is emerging as the single standard. There has been little research exploring the impact that the DOCSIS 1.1/2.0 MAC and physical layers has on the performance of Internet applications. We have developed a model of DOCSIS using the '*ns*' simulation package. In this paper we present the results of a performance analysis that we have conducted using the model. The contribution of our work is twofold. First, we provide insight into the interaction between the DOCSIS MAC protocol and web traffic. Our analysis suggests that DOCSIS does not efficiently support downstream web browsing. Second, we show how a DOCSIS system is vulnerable to a denial of service attack by a hacker who exploits the interaction between the DOCSIS MAC layer and TCP. We show that downstream rate control is not sufficient to avoid the vulnerability.

Keywords— DOCSIS, TCP performance, performance analysis, broadband access

Introduction

The DOCSIS Radio Frequency Interface specification defines the Media Access Control (MAC) layer as well as the physical communications layer [1] that is used to provide high speed data communication over a cable HFC infrastructure. DOCSIS 1.1, the current standard, provides a set of ATM-like services with equivalent quality of service mechanisms. The next generation DOCSIS standard (version 2.0) enhances the physical layer communication methods with higher upstream date rates and improved performance tolerance to bursts of noise. More importantly, DOCSIS 2.0 can provide symmetric data communications.

Figure 1 illustrates a simplified DOCSIS environment. A Cable Modem Termination System (CMTS) interfaces with hundreds or possibly thousands of Cable Modem's (CMs). The 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 6 Mhz (8Mhz in Europe) is shared by all CMs in a one-to-many bus configuration (i.e., the CMTS is the only sender). DOCSIS 1.1 supports a maximum downstream data rate of roughly 30.34Mbps. Upstream channels of 3.2Mhz offer maximum data rates up to 10.3.Mbps that is shared by all CMs using a TDMA based system. DOCSIS 2.0 increases upstream capacity to 30 Mbps through more advanced modulation techniques and by increasing the RF channel allocation to 6.4 Mhz.

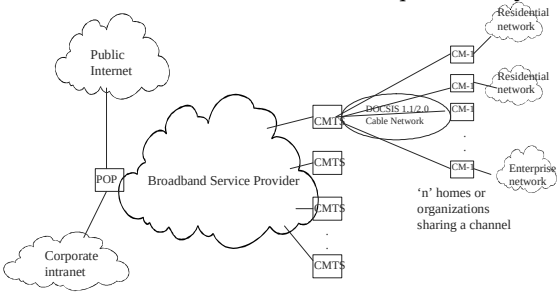

Figure 1. DOCSIS cable access environment

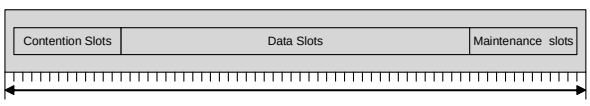

Figure 2. Example upstream MAP allocation

The CMTS makes upstream CM bandwidth allocations based on CM requests and QoS policy requirements. The upstream channel is divided into 'mini-slots' which, depending on system configuration, normally contain between 8 to 32 bytes of data. Figure 2 illustrates a possible MAP allocation that includes allocated slots for contention requests, user data and management data. A critical component of DOCSIS is the upstream bandwidth allocation algorithm. The DOCSIS specification purposely does not specify these algorithms so that vendors are able to develop their own solutions. DOSCIS does require 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)
- Best Effort Service (BE)

All DOCSIS scheduling algorithms will share a set of basic system parameters. These include the amount of time in the future that the scheduler considers when making allocation decisions (we refer to this parameter as the MAP_TIME), the frequency at which MAPs are issued, the frequency of contention slot offerings and the range of collision backoff times. The complex interactions between DOCSIS operating parameters and the subsequent impact on system and application performance is not well understood.

We have developed a model of the Data over Cable (DOCSIS) 1.1/2.0 MAC and physical layers using the '*ns*' simulation package [2]. In previous work, we reported on the impact of several DOCSIS operating parameters on TCP/IP performance [3]. In this paper we extend those results by looking in greater detail at the impact that the MAC layer has on TCP performance when using the DOCSIS best effort service. We show that the interaction between DOCSIS and TCP exposes a denial of service vulnerability. By taking advantage of the inefficiency surrounding upstream transmissions, a hacker can severely impact network performance. This paper is organized as follows. The next section presents the operation and features of our DOCSIS model. We explain our experimental methodology and then discuss the results. We end the paper with a discussion of related work, present conclusions and identify future work.

Summary of the Model

All CMs receive periodic MAP messages from the CMTS over the downstream channel that identify future scheduling opportunities over the next MAP time. For best effort traffic, it is likely that bandwidth will be requested during contention transmission opportunities that are specified by the MAP. Once a request for bandwidth arrives from a CM, the CMTS responds with a data grant pending indication. This informs the CM that the contention-based request succeeded and to expect a data grant at some point in the future. The CONTENTION_SLOTS parameter determines the number of contention slots allocated in each MAP. DOCSIS allows the CM to combine multiple IP packets into a single DOCSIS frame by issuing a concatenated request. If a CM receives a grant for a smaller number of mini-slots than were requested (even for a concatenated request), the CM must fragment the data to fit into the assigned slots over several frames. To minimize the frequency of contention-based bandwidth requests, a CM can piggyback a request for bandwidth on an upstream data frame. To help us evaluate the benefits of these features, we have configuration parameters that enable or disable fragmentation, piggybacked requests and that limit the maximum number of packets that can be included in a concatenated MAC frame.

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 'lookahead'. 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. For the experiments described in this paper, the MAP_LOOKAHEAD was set to 255 and there were 80 slots in each MAP. Roughly 100 slots were required to transmit a frame containing a 1500 byte IP packet and 200 slots were required to send a concatenated frame containing two 1500 byte IP packets.

Methodology

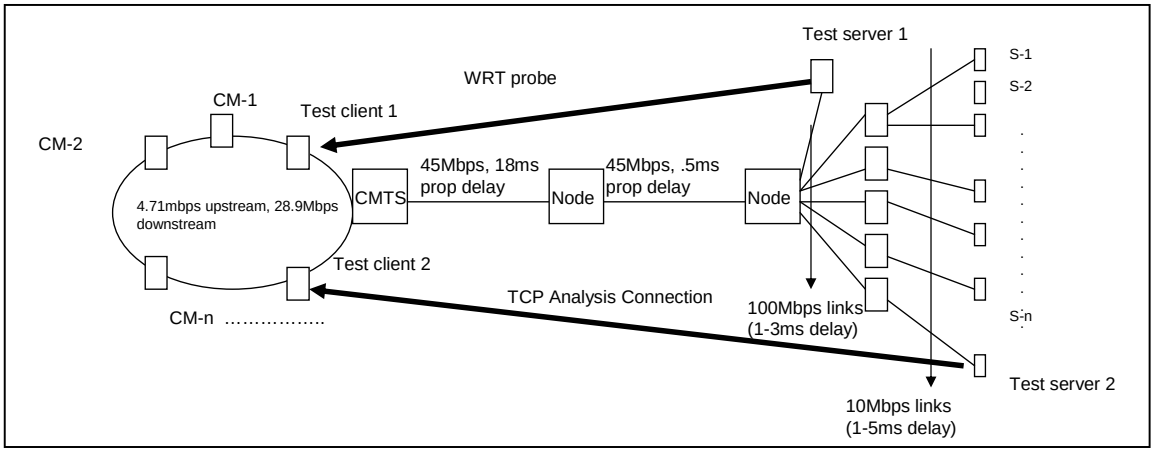

Figure 3. Simulation network

Our analysis used the simulation network shown in Figure 3. The DOCSIS parameters were based on optimal configuration parameters that we found in a previous study [3]. The results we report in this paper were based on two sets of experiments that utilized realistic traffic models consisting of a combination of web, P2P and streaming traffic. The network and web traffic models were based on the "flexbell" model defined in [4]. In addition to downstream web traffic, we configure 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 [5]) incorporates 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 the first experiment, referred to as the web congestion study, we varied two parameters, the MAP_TIME and the number of CMs. For a given MAP_TIME setting, we varied the number of CMs from 100 to 500[1]. We do this for six MAP_TIME settings ranging from .001 to .01 seconds. In the second experiment we used a MAP_TIME of .002 seconds and 100 CMs each configured with the traffic model described in Figure 4. We subjected the CMs in the network to a denial of service attack varying the number of CMs that were under attack. We refer to this experiment as the denial of service study.

For both experiments, we obtained the following statistics.

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 packets transmissions attempted.

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

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. A similar Ping metric is also obtained but we found that a TCP response time metric is a much better indicator of TCP performance. 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 is a convenient, reproducible 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

Results and Analysis

Web Congestion Study
When the dominant application is web browsing the majority of data travels in the downstream direction. However, the system can become packet rate bound in the upstream direction which can limit downstream throughput due to a reduced acknowledgement rate. A CM requests upstream bandwidth to send an ACK by either piggybacking a request or using a contention based request. If more than one ACK is waiting to be send, the CM can concatenate multiple packets into the current transmission. For web traffic, piggybacking is of limited benefit since ACKs that arrive back-to-back are sent in a concatenated frame and the rest of the ACKs arrive at the CM with interarrival times that prevent piggybacking. Concatenation can be helpful although it drastically increases the level of ACK compression experienced by downstream TCP connections [3]. Further, the frequency of concatenation drops as the network load increased. For the experiments reported in this paper, piggybacking and concatenation were enabled however the maximum number of packets that could be concatenated into a

single upstream transmission was limited to 2. The majority of upstream transmissions in the experiment required contention-based bandwidth requests.

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%. What makes this result alarming is that the web traffic model accounts for the heavy tailed distribution associated with web user idle times. Consequently, the number of users actually competing for bandwidth at any given time is much less than 100. As the load increased, the collision rate approached 90-100% depending on the MAP_TIME setting.

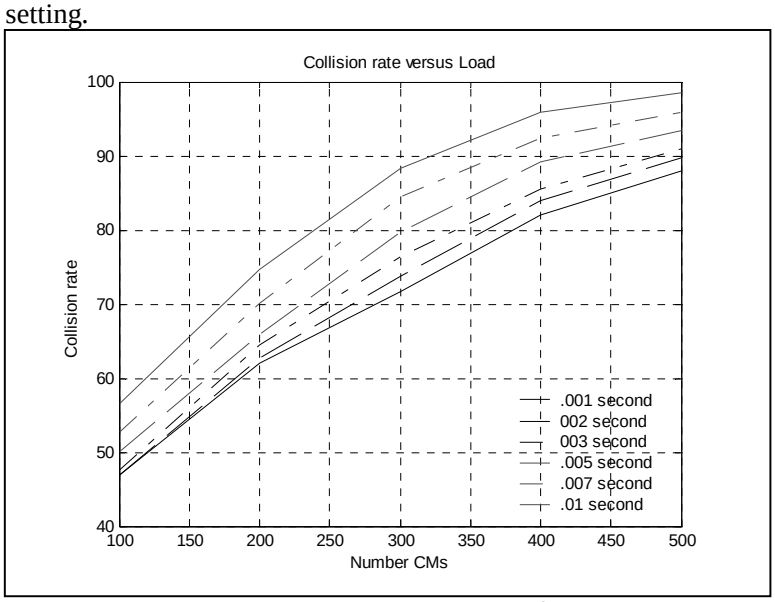

Figure 5. Upstream collision rates as the number of CMs increase

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. Our model uses 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.

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. In a separate experiment, we enabled a single high speed UDP CBR downstream flow with a rate of 40Mbps and achieved 100% downstream channel utilization. While it is possible to tweak system parameters to marginally improve downstream utilization, the invariant result is that a DOCSIS best effort service is not

able to make use of all available downstream bandwidth due to the interaction between the MAC layer and TCP. This result is true when a single upstream channel of any capacity is used. Downstream utilization can be improved by using multiple upstream channels. The focus of our study however was to analyze DOCSIS when using a single upstream channel.

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.

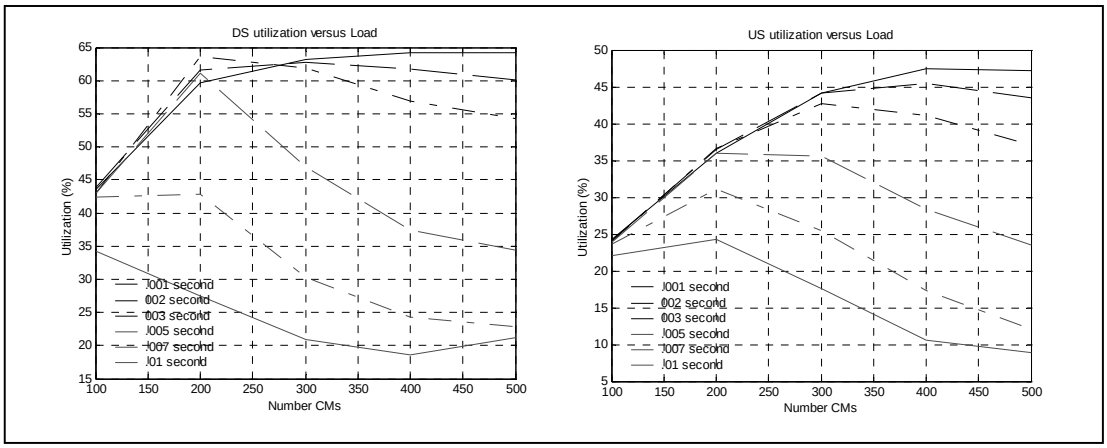

Figure 6a. Downstream channel utilizations

Figure 6b. Upstream channel utilizations

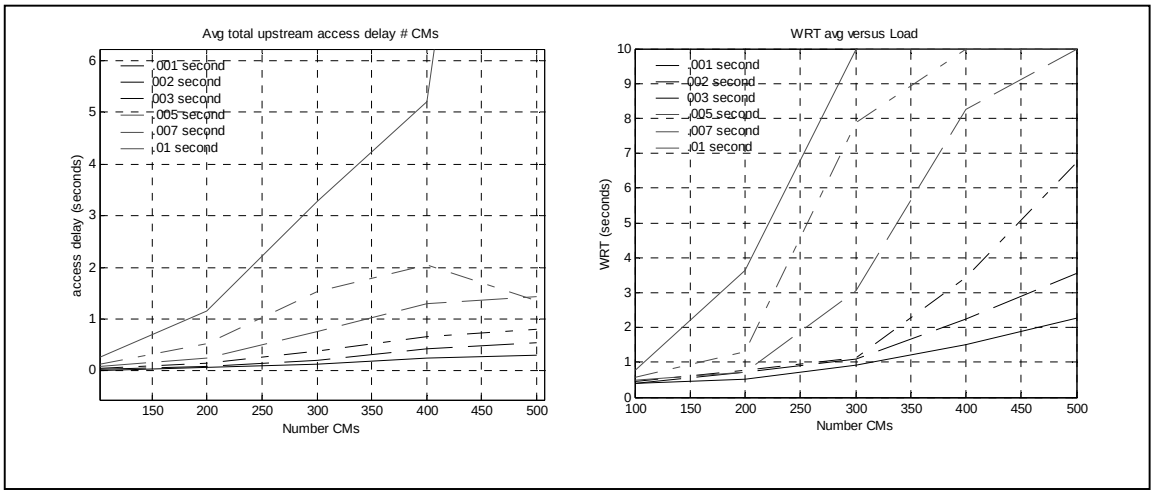

Figure 7a. Upstream access delay (no rate control)

Figure 7b. Web response time metric results

Cable service providers do not offer the full downstream bandwidth to best effort Internet access customers. Instead, providers configure DOCSIS to enforce downstream and upstream service rates. While 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. While a lower service rate, such as 512Kbps, would allow the network to support more users, the Internet access service would provide less value due to the reduced perceived quality at lower service rates

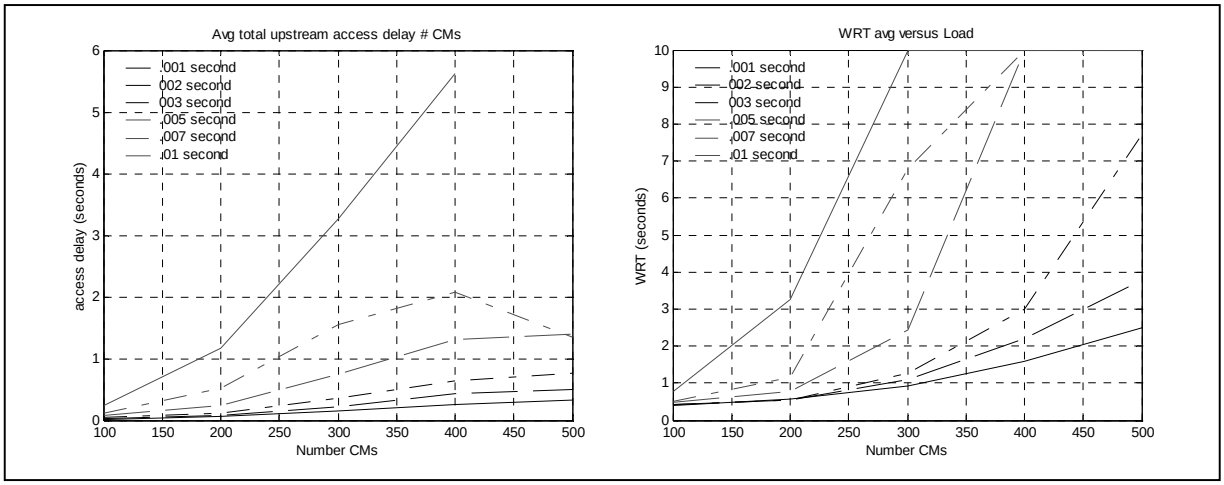

Figure 8a. Upstream access delay (with rate control)

Figure 8b. Web response time metric results

Denial of Service Study

The previous analysis showed that downstream TCP transfers are impacted by the DOCSIS MAC layer's upstream best effort transmission service. In this section we show that it is possible for a hacker to take advantage of this inefficiency by initiating a denial of service attack on CMs that can cause high levels of upstream collisions resulting in serious performance degradation. To accomplish the denial of service attack, a host located outside the network must learn the IP address of a number of CMs that share the same downstream and upstream channels. The attacker simply needs to ping or send a TCP SYN packet to the CMs at a frequency that is on the order of the MAP_TIME setting. The actual frequency, which might range from once per MAP_TIME to once every 5 MAP_TIMEs, is a parameter of the attack.

Figures 9a and 9b illustrate what happens to network performance as an increasing number of CMs come under attack. The configuration was identical to that described in Figure 4 with the MAP_TIME set to .002 second. There were 100 CMs active. Figure 9a shows that the collision rate increased from 48% to 68% as the number of CMs under attack increased from 0 to 100. The downstream utilization dropped from 45% to 10%. Figure 10a shows that the web response times increase by a factor of 2.5. The web response times were observed by a CM (i.e., Test Client 1) that was not under attack. In a separate experiment, we included the Test Client 1 CM in the attack and found that the CM was not able to complete a single response time sample. A CM under attack experiences high packet loss rates at the CM's upstream transmission queue along with loss caused by unsuccessful contention requests. The result is that CMs that are attacked can become unusable.

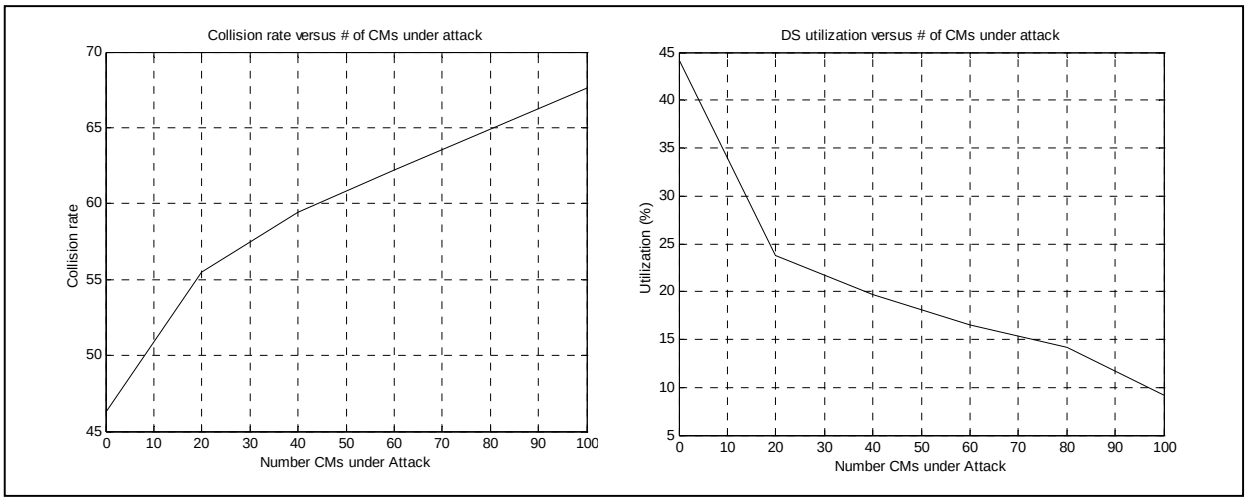

Figure 9a. Upstream collision rates (denial of service)

Figure 9b. Downstream channel utilization

We ran the denial of service experiment a second time with downstream service rates set to 2Mbps. The increase in the collision rates and the drop in downstream utilizations that was observed in the experiment without rate control was virtually identical. Figure 10b shows the average web response times from Test Client 1 which was not under attack. The results suggest that a 2Mbps service rate will not protect the network from the attack.

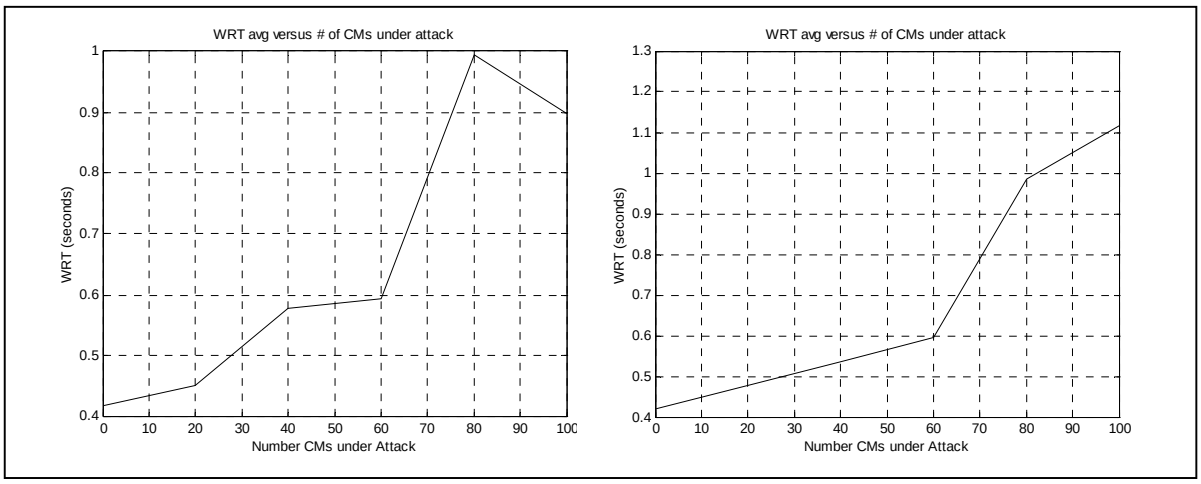

Figure 10. Web response time metric results versus the number of CMs under attack.

Figure 10a. Without rate control

Figure 10b. With 2Mbps downstream rate control

Related Work

While the intent of the 802.14 effort was to provide ATM services over a hybrid fiber coaxial (HFC) medium, the operation of the MAC layer is similar to that supported by DOCSIS. Therefore, prior 802.14 research is relevant. The work in [6] 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. We distinguish our work by focusing on the latest DOCSIS standards (1.1 and 2.0) and using more realistic traffic loads.

The observation in [7] is that an HFC network presents difficulties for TCP due to the asymmetry and due to high loss rates (possibly as high as 10-50%). Due to the problems of TCP/Reno in these environments[8,9,10], 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 motivations behind [7] are not relevant with the latest DOCSIS standards as DOCSIS 2.0 provides nearly symmetric access links with low packet loss rates as long as the plant is well engineered[2].

The performance of TCP over asymmetric paths has been thoroughly studied [11,12,13]. 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 [11]. In DOCSIS, depending on the service rate configuration, the level of bandwidth asymmetry is small (or nonexistent). Instead, DOCSIS 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) [11,12,13,14]. 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. We plan on addressing this in the future.

Cablelabs has developed a model of DOCSIS 1.1 using the Opnet simulator [15]. Their intent was to make the model available to CableLabs members. Since the model source code is not generally available, it is of limited use to the Internet research community. While several relevant studies based on the Opnet model have been performed [16,17], it is difficult to directly compare results due to differences in objectives and in simulation details.

Conclusions and Future Work

The number of broadband cable Internet access users is approaching 80 million world wide[18]. In spite of these large numbers, there have been very few studies of DOCSIS. Using simulation we have provided insight in how the DOCSIS MAC protocol impacts TCP connections. When subject to low levels of downstream web traffic, a correctly configured DOCSIS network experiences collision rates greater than 50% and achieves downstream utilizations less than 65%. The network becomes even more inefficient as the load increases. The difficulties lie in the overhead of the upstream transmission mechanism which can limit the delivery of ACKs in the upstream direction. Increasing the number of contention slots can improve downstream utilization but if too many contention slots are allocated, not enough upstream bandwidth is available to transmit the ACK packets. We are currently working on a dynamic allocation algorithm that tries to find the optimal number of contention slots based on observed conditions. If the set of users assigned to a downstream channel are divided into groups with each group assigned a separate upstream channel, it is possible to improve downstream utilizations. While we have

[2] This is based on private communication with an RF engineer at Cablelabs.

focused on a single upstream channel, we plan to develop and evaluate an algorithm that dynamically switches CMs to different upstream channels based on network conditions.

We have identified a denial of service vulnerability in DOCSIS. Taking advantage of the inefficiency of upstream packet transmissions, a hacker can negatively impact network performance by repeatedly pinging a number of CMs at a frequency that is on the order of the MAP_TIME setting. Network performance deteriorates as the number of CMs under attack grow. The CMs that are under attack can suffer extremely high loss rates effectively making the access network unavailable to users connected to the CM. While the denial of service method that we have described is easy to detect, it is possible for a hacker to disguise the attack but with the same impact. This result further motivates the need for further research in the area of intelligent bandwidth management of shared medium access 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, 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.

[4]. A. Feldmann, et. Al., "Dynamics of IP Traffic: A study of the role of variability and the impact of control", SIGCOM99.

[5] S. Saroiu, P. Gummadi, S. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems", Multimedia Computing and Networking (MMCN), Jan 2002.

[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., "A Simulation-based Study of TCP Dynamics over HFC Networks", Computer Networks, Vol. 32, No. 3, pp 301-317, 2000.

[8] O. Elloumi, et. Al., "Improving Congestion Avoidance Algorithms in Asymmetric Networks", IEEE ICC 97, June 1997.

[9]. K. Fall, S. Floyd, "Simulation-based Comparisons of Tahoe, Reno and SACK TCP", CCR, Vol 26, No. 3, July 1996.

[10]. J. Hoe, "Improving the Startup Behavior of a Congestion Control Scheme for TCP", SIGCOMM 96, August 1996.

[11].H. Balakrishnan, et. Al., "The Effects of Asymmetry on TCP Performance", ", ACM/IEEE International Conference on Mobile Computing and Networking, Sept. 1997.

[12] 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.

[13] V Jacobson, "Compressing TCP/IP Headers for Low-Speed Serial Links", Feb 1990, RFC 1144.

[14] L. kalampoukas, A Varma, K. Ramakrishnan, "Improving TCP Throughput over Two-Way Asymmetric Links: Analysis and Solutions", SIGMETRICS 98, June 1998.

[15]. Opnet simulation model, http://www.opnet.com/services

[16]. 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.

[17].S. Cho, et. Al., "Performance Evaluation of the DOCSIS 1.1 MAC Protocol According to the Structure of a MAP Message", unpublished, available at : http://viplab.hanyang.ac.kr/pdf/5-20.pdf.

[18]. DOCSIS Overview, CableLabs, June 2004, available at http://www.cablemodem.com/downloads/slideshow.ppt.