LETTER

A simple DOCSIS simulator

Logan VOLKERS, Neil BARAKAT, ***and*** **Thomas DARCIE,**

SUMMARY DOCSIS is the defacto industry standard for cable internet to the home. In this letter, we examine the delay characteristics of commercially deployed DOCSIS networks. We focus on four mechanisms of the DOCSIS MAC operation and develop a computationally simple simulator to reproduce the phenomena produced by these mechanisms. In reproducing these phenomena using our simulator, we demonstrate that the simulator properly encapsulates the core mechanisms of DOCSIS and effectively simulates the delay of packets.
key words: *DOCSIS, MAC protocol, broadband access, hybrid fiber coax.*

1. Introduction

Cable Access Television (CATV) networks have become an important medium for broadband internet access. These networks use the Data Over Cable Interface Specification (DOCSIS) to standardize their operation. DOCSIS is the North American *de facto* standard; whereas EURODOCSIS is the similar European *de facto* standard.

Real-time internet applications such as Voice Over IP (VOIP) and online games, specifically reaction games, rely on small delays between users at opposite ends of the network. The delay in DOCSIS networks needs to be understood so that these applications can reach their full potential. Understanding delay is particularly important in the context of DOCSIS because it has been shown in a previous study that a significant fraction of packets experience delays that are multiple times larger than the average delay[1].

Simulating the delay in DOCSIS networks has been the subject of previous research. To examine the behavior of Transfer Control Protocol (TCP) over DOCSIS, the authors in [2] developed a simulator and an analytic model based on a Markov chain. Their analytic and simulation results matched each other, but neither were compared to experimental results.

In [3], the authors used a simulation based on the '*ns*' software package to research improvements in throughput. In [4], the author used a highly parameterized simulator to evaluate the effectiveness of certain DOCSIS mechanisms. In both [3] and [4], only the mean or average delay was examined. None of the above works compared their results to the real-world delays of commercially deployed networks and none have modeled the multi-modal delay characteristic observed in [1].

Finally, a commercial DOCSIS simulator is available in the OPNET simulation package. OPNET simulates every node on the network – a process that, although accurate, is computationally intensive and therefore renders OPNET impractical for simple real-time situations.

In this letter, we present a novel simulator for the delay in DOCSIS networks. The aim of our simulator is to help real-time Internet applications become more robust by providing a simple tool to examine their performance. The novelty of our simulator is that we compute a single user's delay without explicitly simulating the traffic of other users in the system. As a result, our simulator is computationally and parametrically simple. Finally, our simulator is verified by comparing its results to those obtained experimentally from commercially deployed networks and made available for free download and unconstrained use.

2. Delay in DOCSIS Networks

A packet sent across the internet will accumulate delay at each step along the way to its destination. In [1], the authors observed that a majority of the delay and most of the jitter in a DOCSIS access network happen in the step between the Cable Modem (CM) and the Cable Modem Termination Server (CMTS). They concluded that the unique signature of the delay distribution was caused by the DOCSIS Medium Access Control (MAC) Protocol.

When a CM needs to send a packet up to the CMTS, it uses the upstream channel that is shared amongst several CMs. The MAC protocol controls access to the channel by forcing every CM to send a *bandwidth request* before sending any other data. The CMTS collects these *bandwidth requests* and allocates a *bandwidth grant* for each CM – a dedicated period of time for each CM to send their data in.

Bandwidth in DOCSIS is asked for (requested) before it is given (granted). This request-grant cycle accounts for a large amount of jitter in DOCSIS networks. There are several mechanisms related to this request-grant cycle: These are *contention*, *concatenation* and

Manuscript received January 1, 2008.
Manuscript revised January 1, 2008.
Final manuscript received January 1, 2008.
[†]The author is with the University of Victoria
DOI: 10.1587/trans.E0.??.1

piggybacking.

When a network is heavily loaded, there is a high probability that two or more requests arrive at the CMTS at the same time. When such *contention* occurs the CMTS doesn't allocate any bandwidth for these requests. When the CMs don't receive the bandwidth grants that they expected, they will both know that they have *contended*.

After a contention, a CM will try sending another bandwidth request, but it will do so only after a random wait called a *backoff*. This random wait aims to prevent a second contention by ensuring that the CMs do not again transmit at the same time.

Contention forces a CM to send multiple requests for a single packet; a process that causes delay. While one packet is waiting to be sent, other packets may arrive in the queue – causing a backlog. Backlogs due to contention could have a substantially detrimental effect on the delay if it were not for *concatenation* and *piggybacking*.

Concatenation is a mechanism to combine many bandwidth requests into one. The CM sends one request for many packets instead of one request per packet. Since subsequent bandwidth requests can't be sent until the previous request is dealt with, concatenation allows requests for backlogged packets to be sent much faster than in a complete request-grant cycle.

A CM can send a bandwidth request in one of two ways: either during the time dedicated for bandwidth requests or after one of its allocated bandwidth grants. In the latter situation, a bandwidth request for an additional packet is squeezed in after the data of the packet currently being serviced. Since a bandwidth request is small compared to a typical packet, this is called *piggybacking*. *Piggybacking* a bandwidth request protects it from the negative effects of *contention* that it could suffer if it was sent in the time dedicated for bandwidth grants. It may help to note that it is impossible for an initial bandwidth request to be "*piggybacked*", as there are no bandwidth grants to "*piggyback*" on.

3. The simulator

In order to simulate delay for a packet, we look at when it arrives and how big it is. If the packet arrives at the correct time, as per DOCSIS, it could be *piggybacked* or *concatenated* with other packets. The simulator keeps track of the previous packet to determine if the packet can be *piggybacked* or *concatenated*. We extracted variable parameters from the DOCSIS specification and simplified those that we could. We assumed that *contention* occurs with constant probability and that the CMTS bandwidth grant scheduler can be predicted by a uniform probability distribution. Both of these assumptions fit with experimental results and prove concise in producing realistic results.

3.1 Performance

Our simulator is coded using the Matlab software package. Matlab allows us to focus on what we simulate rather than how we simulate. We leverage its built-in random number generator for the randomness needed for simulating non-deterministic traffic.

The simulator simplifies the computation of network delay by abstracting the other clients and the server of the network using random variables. This level of abstraction allows our simulator to be very computationally simple. The processing time for each packet is therefore only a few milliseconds – small compared to the typical delay experienced in DOCSIS networks.

3.2 Utilization

To use the simulator presented in this letter, it is suggested that one has a knowledge of the Matlab programming language. Our simulator is modular and can thus be used for many applications provided that the end user is capable with Matlab. Instructions for downloading the simulator can be found in the appendix.

3.2.1 Traffic sources

The use the simulator, the first step is to determine the traffic source. The simulator includes a function for generating traffic that can be used as the traffic source. We used this function to generate the random traffic in the results shown later in this letter. Alternatively, if the traffic stream is pre-determined, the traffic generator can be bypassed and the individual packets can be fed into the simulator.

3.2.2 Trace generation

With a traffic source chosen, the simulator will generate and record the delays for each of the packets it has been input. This array of delays can then either be exported to a file for use with other application, or re-used within Matlab. For the figures in this letter, we transformed the array into the graphs shown here. The code to accomplish these graphs has also been included with the downloadable code.

3.2.3 Real-time simulation

The fast processing time of our simulator allows it to be used for real-time traffic simulation in Matlab. If the user needs to use another programming language for simulation, the translation to such a language such (like C++) is arbitrary.

Alternatively, our simulator can be used in conjunction with other software to simulate traffic in real time. To use such real-time simulation software such as [5] , a trace for some traffic pattern must first be generated by our simulator and exported to a file.

3.3 Parameters

The parameters of the simulator are broken into three groups: standard parameters, trace-derived parameters and specification parameters. The standard parameters are required but only one of the latter two should be used at a time. The trace-derived parameters reproduce the behaviour of a system using knowledge of its delay characteristic, while the specification parameters reproduce the behaviour of a system using knowledge of its configuration parameters.

3.4 MAP explained

The MAP parameter is an array with each entry representing a transmission time period (slot). These can be used for either data (1), request (0) or maintenance (-1).

During the simulation, the MAP is used to determine when a CM can transmit: The CM can only transmit data in data slots and requests in request slots (with the exception of piggybacking). For the trace-derived parameters and the specification parameters, the number of slots in a map must be specified. In both of these situations, the number of slots has little effect – it is only the position and relative percentage of the slot types that matters. A separate parameter, the duration of the map in seconds, also has an important effect on the resulting output.

3.5 initialWindowSize and maximumWindowSize explained

DOCSIS uses a backoff algorithm called *truncated exponential backoff* to ensure that subsequent requests do not overlap. The backoff window is the period of time over which a CM will randomly wait before transmitting a bandwidth request. Adjusting these backoff window parameters can have a detrimental effect when a backoff window is smaller than the number of request slots in a map described in the previous section.

Table 1 Standard parameters

netLoad	The current network load
lookAheadWait	The number of maps between a request and a grant
initialWindowSize	The initial backoff window size
maximumWindowSize	The maximum backoff window size. Larger backoffs are truncated to this value.
maxGrantSize	The maximum upstream grant size (in bytes). Packets larger than this will be fragmented.
fragmentationOn	Is fragmentation enabled?
concatenationOn	Is concatenation enabled?
piggybackingOn	Is piggybacking enabled?

Table 2 Specification parameters

secondsPerMap	The duration of a MAP in seconds
MAP	The slots of a MAP. (see below)

Table 3 Trace-derived parameters

minPingDelay	The minimum delay from the trace. Used to find the secondsPerMap.
numDataSlots	The number of data transmission slots. Used to generate a MAP.
numRequestSlots	The number of request slots. Used to generate a MAP.
numMaintenSlots	The number of maintenance slots. Used to generate a MAP.

4. Results

In this section, we compare experimental results from commercially deployed DOCSIS networks against those obtained through simulation. A ping program generated the traffic for the experimental results and a traffic generator created the traffic for our simulation results. The two parameters that characterize the traffic and are the key difference between the following results are the inter-packet arrival times and the packet sizes.

4.1 The Effect of Contention Requests

Figure 1 shows the distribution of the round-trip times of 32 byte pings sent every second. The important feature to note in this figure is the multi-modal behavior. There are several "humps" in the delay density function, each one representing a group of packets.

Packets that sent a bandwidth request and received a bandwidth grant immediately or without contention are shown in the first hump on the left. The next hump to the right contains the packets that suffered a contention on their first request and had to retry before a bandwidth grant was received. The third hump contains the packets that contended twice before they received a bandwidth grant, and so on.

4.2 The Effect of Concatenation and Piggybacking

Figure 2 results from 32 byte pings sent every 5 ms and shows the effect of concatenation and piggybacking when a backlog occurs. When one packet experiences a large delay, the others are forced to queue up behind it, causing a backlog.

In order to clear this backlog, the packets that waited during the backlog are *concatenated* together into a single bandwidth request – that is, a concatenated bandwidth request for the combination of the backlogged packets is sent.

It is important to note that the CM is not allowed to send the combined bandwidth request of the backlog as a concatenated request right away. It must wait until the end of the bandwidth grant and send a "piggybacked" request. This creates the flat-tops on the peaks shown in figure 2 where one would otherwise expect them to be sharp.

Also of note is that when requests for several packets are concatenated into the same bandwidth request, the data of the concatenated packets can often fit in the same bandwidth grant. This causes the linear slope seen in Figure 2 after a backlog. Since the transmission bit rate is high, all of the concatenated packets are sent almost instantaneously in the following bandwidth grant and the delay on each of the backlogged packets is equal to the time that they had to wait in the backlog.

It can be seen from both the cumulative distributions and predicted backlog dynamics that our simulator is capable of reproducing the rather complex dynamics of deployed DOCSIS networks.

5. Conclusion

In this letter, we examined the delay characteristics of commercially deployed DOCSIS networks. We focused on three mechanisms: contention, concatenation and piggybacking, and developed a computationally simple simulator to reproduce the phenomena produced by these mechanisms. In reproducing these phenomena using our simulator, we demonstrate that the simulator properly encapsulates the core mechanisms of DOCSIS and effectively simulates the delay of packets.

This work provides a means for researchers to simulate the complex variability of delay in DOCSIS networks without extensive knowledge of its operation.

References

[1] N. Barakat and T.E. Darcie, "Delay characterization of cable access networks," IEEE Communications Letters, vol.11, no.4, pp.357–359, April 2007.
[2] K.C. Chang and W. Liao, "The contention behavior of docsis in catv networks," IEEE Transactions on Broadcasting, vol.53, no.3, pp.660–669, September 2007.
[3] J. Martin and J. Westall, "A simulation model of the docsis protocol," Simulation, vol.83, no.2, pp.139–155, 2007.
[4] R. Murphy, "A simulation study of docsis upstream channel bandwidth allocation strategies for minimal user response time," master of science in computer science, The University of Texas at San Antonio, December 2004.
[5] R. Baumann and U. Fiedler, "A tool for emulating real network dynamics on a pc.," TIK Report 231, ETH Zurich, 2005.

Fig. 1 Round-trip delay probability density function – demonstrates contention

Fig. 2 Delays of 32 byte pings sent in 5ms intervals

Appendix: Downloading the Simulator

The simulator is available for download from http://www.ece.uvic.ca/ tdarcie/publications.php. The simulator comes in a package that contains Matlab code and utilization instructions. In order to use the simulator, one must have a licensed version of Matlab installed on one's computer.