The Impact of the DOCSIS 1.1/2.0 MAC Protocol on TCP

Jim Martin
Department of Computer Science
Clemson University
Clemson, SC 29634-0974
jim.martin@cs.clemson.edu

***Abstract*-- The number of broadband cable access subscribers in the United States is rapidly approaching 30 million. However there is very little research that has evaluated TCP/IP in modern cable environments. We have developed a model of the Data over Cable System Interface Specification (DOCSIS) 1.1/2.0 MAC and physical layers using the '*ns*' simulation package. In this paper we show that the interaction of the MAC layer on downstream TCP web traffic leads to poor network performance as the number of active users grow. We provide an intuitive explanation of this result and explore several possible improvements.**

Keywords— Last Mile Network Technologies, DOCSIS, TCP Performance

I. Introduction

The Data over Cable (DOCSIS) Service Interface Specification defines the Media Access Control (MAC) layer as well as the physical communications layer that is used in the majority of hybrid fiber coaxial cable networks that offer data services [1]. 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 original DOCSIS MAC interface (v1.0) provides a best effort service with simple prioritization capabilities. DOCSIS 1.1, which is currently being deployed, adds a set of ATM-like services along with the necessary QoS mechanisms. The follow on standard, version 2.0, enhances the physical layer communication methods with higher upstream data rates and improved tolerance to bursts of noise.

The CMTS makes upstream CM bandwidth allocations based on CM requests and QoS policy requirements. The upstream channel is divided into 'minislots' which, depending on system configuration, normally contain between 8 to 32 bytes of data. The CMTS periodically sends a 'MAP' message to all CMs on a downstream channel that indicates upstream bandwidth allocation over the next 'MAP time'. The MAP provides slot assignments for particular CMs in the form of data grants, provides opportunities for a CM to request bandwidth using a contention-based request process and identifies which slots are to be used for system overhead. Figure 2 illustrates an example MAP layout.

A critical component of the DOCSIS MAC layer is the upstream bandwidth allocation algorithm. The DOCSIS specification purposely does not specify these algorithms so that vendors can develop their own solutions. However, all upstream bandwidth management algorithm will share a set of basic system parameters such as 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 amount of upstream bandwidth allocated for contention-based bandwidth requests and the range of collision backoff times. These parameters are crucial for ensuring good performance at high load levels.

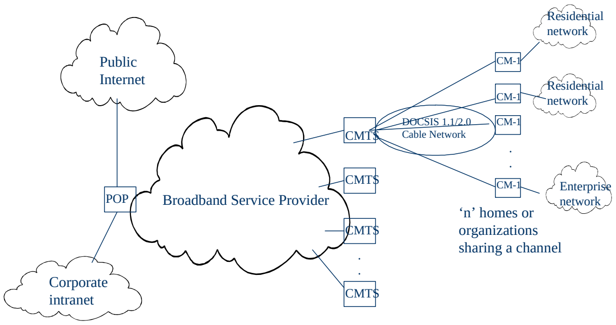

Figure 1. DOCSIS cable access environment

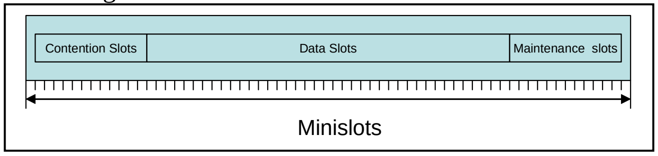

Figure 2. Example DOCSIS MAP layout

Prior analysis of the IEEE 802.14 standard, the predecessor to DOCSIS, found that TCP throughput over an hybrid fiber co-axial (HFC) cable network is low primarily due to ACK compression [2]. 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. In previous work, we presented an '*ns*' DOCSIS simulation model that we developed and showed that the MAP_TIME setting and the use of concatonation can induce significant levels of ACK compression in downstream TCP connections [3]. ACK compression occurs when a network causes TCP acknowledgement packets to 'bunch' at some point leading to higher loss rates and poor network utilization [5,6,7,8]. Also in previous work, we showed that for certain configurations, DOCSIS networks are vulnerable to a denial of service by an attacker who exploits the interaction between DOCSIS and TCP applications. In this paper we extend our past results and show that the interaction of the MAC layer with downstream TCP web traffic leads to poor network performance as the number of active users grow. We provide an intuitive explanation of this result along with possible improvements. The rest of this paper is organized as follows. Section II presents the operation and features of our DOCSIS model. In Section III, we present the results of a simulation analysis using our *'ns'* DOCSIS model involving hundreds of active CMs generating a mix of web, streaming and P2P traffic. Section IV ends the paper with a discussion of conclusions and future work.

II. SUMMARY OF THE MODEL

The model implements the DOCSIS architecture defined in [1] although with several limitations. A CM can only support a single default best effort service flow and any number of UGS services. Also, the model is limited to one upstream channel for each downstream channel. As in an actual DOCSIS implementation, packets sent over the downstream channel are broken into 188 byte MPEG frames each with 4 bytes of header and trailer. The model accounts for physical overhead including forward error correcting data. The downstream channel supports an optional token bucket backed service rate. Each SID service queue is treated in a first come first serve manner. Depending on traffic dynamics, queueing can occur at either the SID queue or the downstream transmission queue. The maximum size of either queue is a simulation parameter.

All CMs receive periodic MAP messages from the CMTS that identify future scheduling opportunities over the next MAP time. If provisioned with a periodic grant, the CM can send at its next data grant opportunity. For best effort traffic, the CM must request bandwidth from the CMTS using a contention-based transmission opportunity specified by the MAP. A CM can request bandwidth to combine multiple IP packets into a single DOCSIS frame by issuing a concatenated request. To minimize the frequency of contention-based bandwidth requests, a CM can piggyback a request for bandwidth on an upstream data frame. If a CM receives a grant for a smaller number of minislots than were requested, the CM must fragment the data to fit into the assigned slots.

Figure 3 illustrates the upstream transmission of a 1500 byte IP datagram from a TCP source directly connected to a CM to a sink connected to the CMTS. We show the upstream slot allocations for three MAP times based on the MAP layout illustrated in Figure 2. In Figure 3, time progresses in the downwards direction. In our example, we assume collisions do not occur. Although the nominal MAP size is 80 minislots (based on the system parameters listed in Figure 3), 96 slots are required to carry the entire packet. We assume that the CM will request 96 slots to carry the full IP packet[1]. The small dark square box positioned at the beginning each MAP time in the figure represents the transmission of the MAP message in the downstream direction. Our model sends the MAP at the beginning of each MAP time. Each MAP describes the slot assignments for the next MAP time. The IP packet arrives at the CM at time T-0. The CM sends the bandwidth request message at time T-1 and receives the data grant at time T-2. The grant is located in the third MAP time. The CM sends the frame at Time T-3 and is received by the CMTS at time T-4. The time between T-4 and T-0 is the access delay which represents the total time a packet is delayed over the DOCSIS access network.

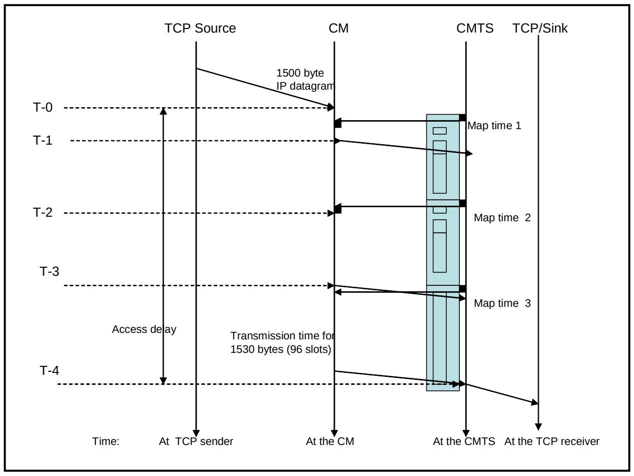

Figure 3. Upstream transmission

The maximum upstream TCP throughput achieved with a best effort service is $T_{\max-us} = \frac{BytesSent*8}{D_{total-access}}$, where *BytesSent* is the total data sent in an upstream transmission opportunity and $D_{total-access}$ is the access delay. The best case scenario illustrated in Figure 3 assumes no system delay and that the MAP_TIME can grow to allow the transmission of full size IP packets in a single frame. The minimum access delay (i.e., $D_{total-access}$) is roughly 2 MAP times plus the transmission time of the frame over the upstream channel. Assuming a MAP time of .002 seconds, no concatenation, physical layer overhead of 30 bytes and an upstream bandwidth of 5.12Mbps, the highest throughput to be expected is roughly 1.8Mbps. We have verified this with the simulation model with a single TCP connection using an unloaded network.

When a packet arrives at a CM to be sent in the upstream direction, it is unlikely to experience the ideal scenario depicted in Figure 3. Delay caused by the

[1] The bandwidth allocation algorithm allows the MAP_TIME to vary if conditions permit in an effort to improve performance.

contention-based bandwidth request process along with delay incurred while waiting for the CMTS to issue a grant can increase the upstream access delay significantly. When multiple packets arrive at a CM, the CM can piggyback a request for additional bandwidth on an upstream data frame. This can lead to significant benefits in a busy system as it avoids the contention process resulting in lower collision rates.

We provide a similar discussion for downstream. Figure 4 illustrates a downstream scenario involving the arrival of 4 back-to-back segments at the TCP sink. We assume that the sink generates two acknowledgement packets (i.e., one ACK for every other segment that arrives) which arrive at the CM separated only by the downstream transmission time of two back-to-back segments. Further, we assume that concatenation is enabled allowing the CM to request bandwidth for the transmission of up to two ACK packets in a single frame. In Figure 4, the concatonated request experiences collisions on the first two contention-based transmissions, and is successfully received sent on the third attempt. The figure also shows significant delay occuring at the CMTS before the CM is allocated a data grant. The CM sees it has been allocated the grant in the MAP that arrives at time T-2. In the best case, ACKs are received at the CMTS roughly within 2 MAP times from the time the ACKs arrive at the CM plus the transmission time of the data frame. Therefore, assuming a MAP_TIME of .002 seconds, we expect a maximum downstream throughput by any single CM to be limited to about 12Mbps corresponding to 4 segments metered out every 4 milliseconds. After setting the TCP window size very large and making sure the buffers at the CM were large, we did observe a maximum downstream TCP throughput of 12Mbps.

The maximum downstream data rate for the configuration depicted by Figure 4 is roughly 26Mbps after taking into account physical and link level overhead. The TCP throughput however is limited by the ACK rate which is the amount of data that is acknowledged per second (as observed by the TCP sender). The maximum downstream TCP throughput is given by $T_{\max-ds} = \frac{BytesACKed * 8}{D_{total-access}}$ where *BytesACKed* is the amount of data that is acknowledged in the bandwidth request and $D_{total-access}$ is the access delay from the time the ACK packet arrives at the CM until when the packet has been received by the CMTS. Increasing upstream or downstream channel capacities will not increase the single connection TCP throughput as the bottleneck is the packet rate in the upstream direction.

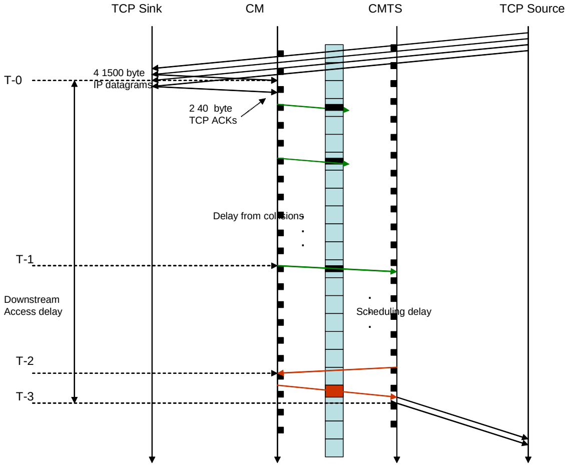

Figure 4. Downstream scenario

III. Results and Analysis

For the analysis presented in this paper, we rely on the simulation model illustrated in Figure 5. Up to ‘n’ cable modems (CMs) each with one simulated user generates upstream web requests to one or more servers (S-x nodes). The DOCSIS protocol handles transfering data between the CMs and the CMTS. Multiple point-to-point links define the path between the CMTS and the web servers. Figure 6 describes the DOCSIS configuration and the web traffic model parameters. The web traffic model is based on the model described in [9]. In addition to 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 [10]) 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.

We define an experiment to consist of 5 simulation runs with each run configured with a larger number of CMs (i.e., 100 to 500). We perform 6 experiments with each experiment configured with a different MAP_TIME (.001 to .01 seconds). We collect a set of statistics including the collision rate, channel utilizations and average TCP loss rates for each run.

Figure 7 shows that the collision rate got extremely high as the number of active CMs increased. When 100 users were active, the collision rate was about 50%. As the load increased, the collision rate approached 90-100% depending on the MAP_TIME setting.

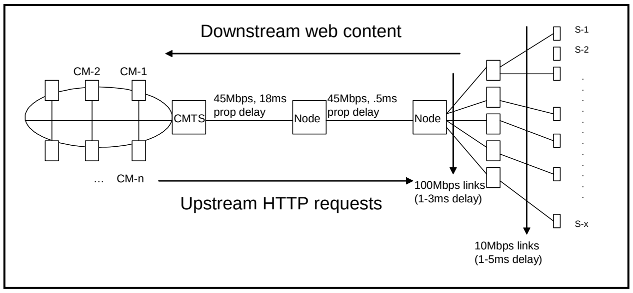

Figure 5. Simulation network model

Model Parameters

Upstream bandwidth 5.12Mbps, no service rate
Preamble 80 bits
Downstream bandwidth 30.34Mbps, no service rate
4 ticks per minislot
Default map time: 2 milliseconds (80 minislots per map)
Fragmentation Off
Concatonation ON
Backoff Start: 8 slots, Backoff stop: 128 slots
12 contention slots, 3 management slots per MAP
Simulation time: 500 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 6. Model parameters

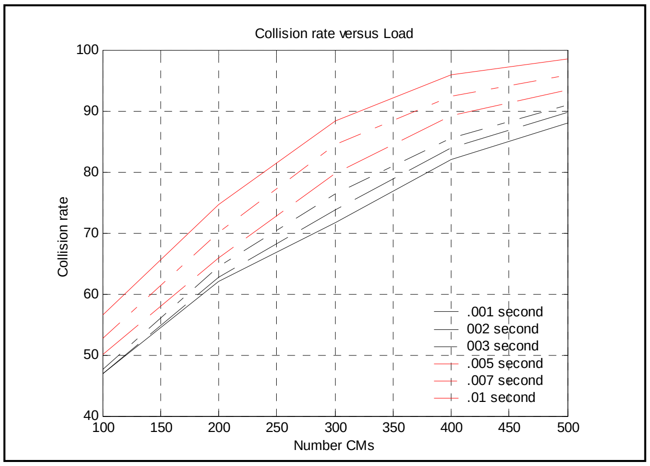

Figure 7. Upstream collision rates

The behavior of the system is greatly influenced by several MAC protocol parameters. First, the number of contention slots assigned per map directly impacts the collision rates at high loads. Our model uses a fixed number of contention slots per MAP that is controlled with the configuration parameter CONTENTION_SLOTS. For the results reported in this paper, we used 12 contention slots per MAP. If we increase the number of contention slots to 20, the curves in Figure 7 look similar, especially at higher loads which confirms our earlier finding that 12 contention slots per MAP is sufficient for a web traffic scenario [4]. As mentioned earlier, performance improves as the MAP_TIME gets smaller. Figure 7 illustrates that the collision rate at the smallest MAP_TIME setting (i.e., .001) is 10-15% lower than at the highest MAP_TIME setting. However, if the MAP_TIME becomes too small it consumes significant processing resources at the CMTS. In practice, a MAP_TIME in the range of .001 to .01 is typical.

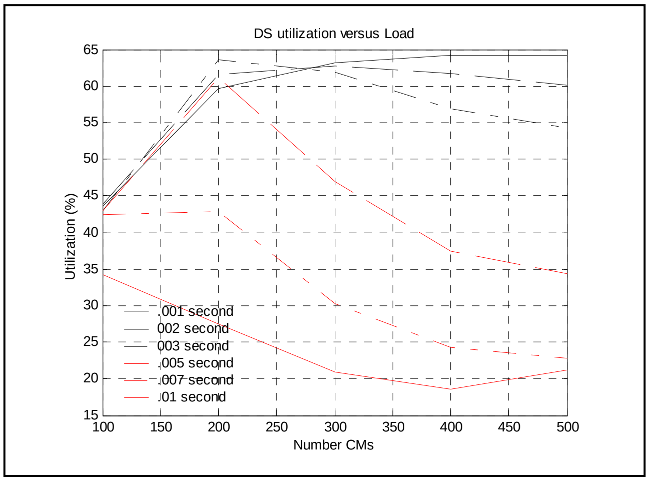

Figure 8. Downstream channel utilization

Figure 8 shows that even with a small MAP_TIME setting, the maximum downstream channel utilization is less than 65%. As the user load increases, ACK packets are not sent upstream fast enough which limits downstream throughput. The symptoms of this behavior are high collision rates and reduced downstream channel utilization. To assess the impact of the problem on application performance we used a simple web application performance metric by running a TCP client server application between CM-1 and the server S-1. Server S-1 periodically sent 20Kbytes of data to the client. At the end of the run we computed the mean of the response times. A Ping metric was also obtained but we found that a TCP response time metric was 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. Clearly this is not an accurate measure of end user quality of experience. It is however a convenient and reproducible performance reference. Figure 9 suggests that 300 active users is roughly the point at which the network becomes overloaded for reasonable MAP_TIME settings.

Higher channel capacities will not increase downstream channel utilization as the bottleneck is caused by low packet rates in the upstream direction. One approach to increasing the downstream channel utilization is to allocate multiple upstream channels to the group of CMs that share a downstream channel[2]. As long as the aggregate load is spread evenly between the upstream channels, the downstream channel can be driven to 100% utilization. Another approach that has been proposed for both DOCSIS and wireless networks is ACK filtering [11,12,13,14]. CableLabs is currently evaluating if they should add ACK filtering capabilities to the next version of DOCSIS [15].

We have implemented a simple ACK filtering algorithm that works as follows. When an ACK packet arrives at a CM to be sent upstream, any ACK that is currently in the queue is dropped if it is associated with the same connection as the ACK that arrived and if it acknowledges a sequence number lower than the sequence number of the ACK that arrived. We reran our experiment using ACK filtering. The downstream utilization actually dropped significantly as the small CMTS queue size (set to 50 packets) was not able to absorb the bursty TCP connection behavior caused by ACK compression. The maximum downstream utilization increased to 75% once we increased the buffer capacity of all nodes in the simulation to 300 packets. While the performance was comparable to the equivalent simulation but without ACK filtering, we confirmed that network dynamics were much more bursty when ACK filtering was enabled.

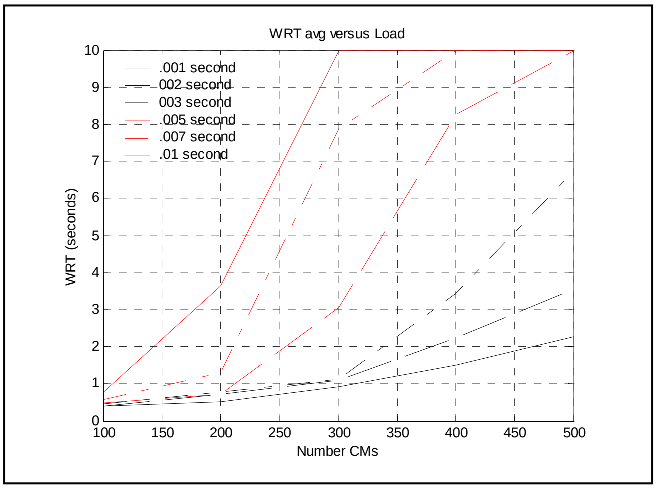

Figure 9. Downstream web response times

IV. Conclusions and Future Work

Based on simulation, we have shown how the DOCSIS MAC protocol impacts TCP performance. Downstream TCP connections might not be able to utilize all available bandwidth because the ACK stream could be limited by the packet rate in the upstream direction. This can lead to poor downstream channel utilization as the number of active users increases. While our analysis did not impose reduced service rates on users, the results would be identical if we had. This is because as the load increased the available bandwidth quickly becomes less than typical service rates. Smaller MAP_TIME settings will increase the upstream packet rate however at the expense of increased overhead. Concatenation or ACK filtering can also help however these approaches can significantly increase the burstiness associated with the TCP send behavior which can lead to higher packet loss rates. While it is possible for a CMTS to 'correct' the ACK stream using rate control [16,17], an open research question is if there are innovative packet scheduling algorithms to get around these problems. The urgency of this area of research will increase dramatically as the next version of DOCSIS provides access speeds of 200 Mbps downstream and 100 Mbps upstream. To address this we are developing a predictive data grant service that can efficiently support Internet applications over current and next generation DOCSIS 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]O. Elloumi, et. Al., "A Simulation-based Study of TCP Dynamics over HFC Networks", Computer Networks, Vol. 32, No. 3, pp 301-317, 2000.

[3]J. Martin, "Modeling the DOCSIS 1.1/2.0 MAC Protocol", ICCCN03, October 2003.

[4]J. Martin, "The Interaction Between the DOCSIS 1.1/2.0 MAC Protocol and TCP Application Performance", Performance Modelling and Evaluation of Heterogeneous Networks, July 2004.

[5]J. Mogul, "Observing TCP Dynamics in Real Networks", Technical Report, Digital Western Lab, April 1992.

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

[7]H. Balakrishnan, et. Al., "TCP Behavior of a Busy Internet Server: Analysis and Improvements", IEEE Infocom98, 1998.

[8]L. Zhang, S. Shenker, D. Clark, "Obserations on the Dynamics of a Congestion Control Algorithm: the Effects of 2-way Traffic", SIGCOMM 1991.

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

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

[11]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.

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

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

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

[15]R. Keenan, "CableLabs details DOCSIS Roadmap", Commsdesign magazine, May19, 2004, www.commsdesign.com.

[16]S. Karandikar, S.Kalyanaraaman, P. Bagal, B. Packer, "TCP Rate Control", ACM Computer Communications Review, 30(1):45-58, January 2000.

[17]M. Chan, R. Ramjee, "Improving TCP/IP Performance over Third Generation Wireless Networks", INFOCOM 2004.

[2] A common provisioning rule of thumb used by providers is to assign 2000 CMs to a downstream channel and 300-500 users per upstream channel.