

FAST REQUEST TRANSMISSION IN DOCSIS-BASED CATV NETWORKS

Hue-Jiun Ju and Wanjiun Liao

Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
Email: wjliao@cc.ee.ntu.edu.tw

ABSTRACT

This paper studies the effect of DOCSIS MAC layer on TCP performance in Hybrid Fiber Coax (HFC) networks. We discuss how DOCSIS MAC mechanism affects bandwidth asymmetry, analyze TCP behavior over DOCSIS MAC layer, and propose a new mechanism called "Fast Request Transmission (FRT)" to improve TCP performance in HFC networks. We have also conducted simulations using network simulator *ns-2* to compare the original control mechanism of DOCSIS with our mechanism. The results show that the proposed FRT mechanism has better performance in terms of aggregate downstream throughput, access delay, and buffer size.

1. INTRODUCTION

An HFC network is a tree-and-branch, point-to-multi-point access network in the downstream direction but a multipoint-to-point, bus access network in the upstream direction. It employs a contention-based reservation control mechanism to arbitrate random access to the upstream channel.

The Data-Over-Cable Service Interface Specifications (DOCSIS) standardizes the physical layer modulation and the MAC layer operation for HFC networks. In DOCSIS, the upstream channel is modeled as a stream of mini-slots. The CMTS controls the usage of the upstream channel. Each CM sends a request to the CMTS for the use of the upstream channel, and waits for bandwidth allocation from the CMTS. The CMTS regularly transmits downstream management messages called Upstream Bandwidth Allocation (denoted as MAP), which defines transmission intervals on the upstream channel. Each transmission interval defined by a MAP contains request mini-slots and data mini-slots. Request mini-slots are used by CMs to request upstream bandwidth; data mini-slots are used by CMs to transmit data frames. All CMs learn the assignment of bandwidth from the MAP. Thus, each MAP must be received by all the CMs before the start of the transmission period it describes.

Previous studies [1-4] have shown that TCP performance degrades when operating in HFC networks due to the bandwidth asymmetry of downstream and upstream channels. This paper investigates the TCP performance problem in HFC networks from a different angle: examine how DOCSIS MAC layer scheduling and allocation mechanisms affect TCP performance. We

discuss the effect of DOCSIS MAC mechanism on bandwidth asymmetry, analyze TCP behavior over DOCSIS MAC layer, and propose a new mechanism called "Fast Request Transmission" to significantly improve TCP performance in HFC networks.

The rest of the paper is organized as follows. Section 2 analyzes TCP behavior over the DOCSIS MAC layer. Section 3 presents the proposed mechanism called "Fast Request Transmission" to improve TCP performance in HFC networks. Section 4 shows the simulation results using *ns2* to compare the original DOCSIS MAC layer control mechanism with our mechanism. Finally, the concluding remark is included in Section 5.

2. TCP PERFORMANCE OVER THE DOCSIS MAC LAYER

2.1 The Effect of Bandwidth Asymmetry on TCP Performance

Asymmetric networks, such as HFC and xDSL, are defined as networks with different channel capacities in the downstream and upstream directions. The main effect of bandwidth asymmetry on TCP performance is that TCP ACK clocking may be disrupted. [1] defines a bandwidth asymmetric ratio, k , to better understand the behavior of TCP in asymmetric networks:

$$k = \frac{\text{forward channel bandwidth}}{\text{reverse channel bandwidth}} \times \frac{\text{ACK packet length}}{\text{data packet length}} \\ = \frac{C_d}{C_u} \times \frac{L_{ack}}{L_{data}} \quad (1)$$

TCP behaves normally when k is less than or equal to one. When bandwidth is asymmetric (i.e., $k > 1$), ACK packets arrive at the bottleneck link in the reverse direction at a rate faster than the bottleneck link can support. As a result, the sender clocks out data at a slower rate, thus slowing down the growth of the sender's congestion window. This in turn lowers the throughput in the downstream direction.

2.2 Problems with TCP Performance in HFC Networks

To examine the impact of the DOCSIS MAC layer control mechanism on TCP performance, we conducted

simulations using ns2 [5] with the simulation setup in Sec. 4.1 to observe TCP's behavior. We set the capacities of the downstream and upstream channels to 26.97 Mbps and 2.56 Mbps, respectively. Considering the Delayed ACK policy in TCP Reno (i.e., sending one ACK packet to acknowledge the receipt of d data packets), we modify the asymmetric ratio k as $k = \frac{C_d}{C_u} \times \frac{L_{ack}}{L_{data}} \times \frac{1}{d}$. With the packet lengths of ACK = 64 bytes and Data = 1024 bytes, and $d = 2$, $k = \frac{26.97}{2.56} \times \frac{64}{1024} \times \frac{1}{2} \cong 0.33$. Since $k \leq 1$, this network should be symmetric in bandwidth, and TCP is supposed to behave normally.

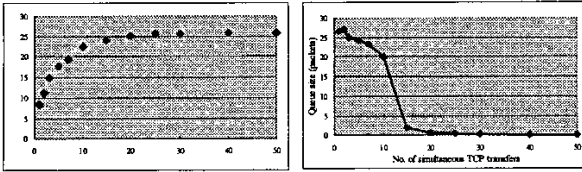


Figure 1. Aggregate throughput Figure 2. Queue size

To investigate the TCP behavior in such a network, we varied the number of CMs doing simultaneous TCP transfers, denoted as N_{dCM} , in the simulation. We observed the influence of varying N_{dCM} on the aggregate downstream throughput and the queue size (which buffers ACK packets) in the CM. Figs. 1 and 2 show the downstream throughput and the average buffer size in the CM, respectively, varying N_{dCM} from 1 to 200. Interestingly, we found that when N_{dCM} was less than 13, the network behaved like an asymmetric network, i.e., low downstream throughput and full upstream ACK queues. However, as N_{dCM} increased beyond 13, TCP started behaving as in a symmetric network. This implied that the network experienced bandwidth asymmetry ($k > 1$) when N_{dCM} was small, resulting in performance degradation in this region. Thus, we learned that k in eq. (1) alone could not adequately explain TCP behavior in DOCSIS-based HFC networks.

2.3 The Effect of DOCSIS MAC Layer on Bandwidth Asymmetry

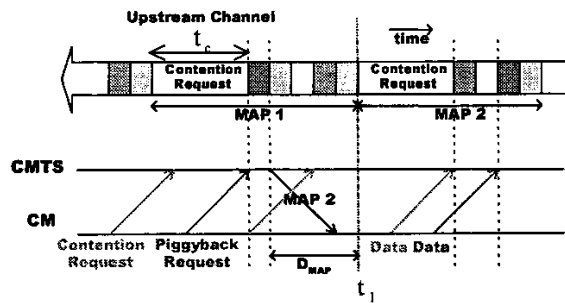


Figure 3. DOCSIS's MAC operation

In DOCSIS, the upstream channel is modeled as a stream of mini-slots, and a transmission starts only at the

beginning of any mini-slot. Let N_{u_ack} be the number of mini-slots used to transmit one ACK packet. Given L_{ack} , C_u , and t_{ms} , we can derive the number of mini-slots used to transmit an ACK packet as

$$N_{u_ack} = \left\lceil \frac{L_{ack}}{C_u} \times \frac{1}{t_{ms}} \right\rceil \quad (2)$$

where L_{ack} is the size of an ACK packet, t_{ms} is time period defined as one mini-slot on the upstream channel, and C_u is the upstream channel capacity.

The MAP describes the bandwidth allocation in a transmission period. It should be received by all the participating CMs before its effective time (i.e., t_1 in Fig. 3). Thus, each MAP may be transmitted before some requests, especially piggybacked ones, have arrived and been processed at the CMTS. The late requests be pending in the current transmission period and become backlogged requests in the next period. These pending requests plus new requests which arrive at the CMTS during the next transmission period will be waiting to be granted on the next MAP. Let D_{MAP} be the time difference between when a MAP is transmitted and when it goes into effect (see Fig. 3). Since the size of the ACK packet is fixed, the number of pending requests, N_{p_REQ} , (i.e., those arrive at the CMTS during a D_{MAP}) can be expressed as

$$N_{p_REQ} = \left\lceil \frac{D_{MAP}}{t_{ms}} \times \frac{1}{N_{u_ack}} \right\rceil \quad (3)$$

T_{usv} is defined as the average time between sending two consecutive ACK packets in the CM buffer. We determine T_{usv} for an ACK packet in two cases.

Case 1: $N_{dCM} \leq 2N_{p_REQ}$

If N_{dCM} is less than or equal to $2N_{p_REQ}$, each CM on the average receives a data grant in every other transmission period, i.e. there are N_{dCM} requests granted every two MAPs. Let t_c be the request contention period (see Fig. 3). Thus, T_{usv} is

$$T_{usv} = (2N_c + N_{dCM} N_{u_ack}) t_{ms} \quad (4)$$

where $N_c = \left\lceil \frac{t_c}{t_{ms}} \right\rceil$.

Case 2: $N_{dCM} > 2N_{p_REQ}$

If N_{dCM} is two times larger than N_{p_REQ} , the mean upstream buffer service time is given by

$$T_{usv} = \frac{[N_c + (N_{dCM} - N_{p_REQ}) N_{u_ack}] t_{ms} N_{dCM}}{N_{dCM} - N_{p_REQ}} \quad (5)$$

Considering the MAC layer operation of DOCSIS 1.1, we modify the asymmetric parameter k as follows:

$$k = \frac{C_d \times T_{usv}}{d \times L_{data} \times N_{dCM}} \quad (6)$$

We replace $\frac{L_{ack}}{C_u}$ with T_{usv} , the service time of packets in

the upstream queue in a CM in the expression for k , taking into the consideration that the upstream HFC networks is a TDMA channel. Usually, a cable modem should wait until the scheduled time to transmit a packet. The upstream buffer service time is the time interval a CM should wait. For downstream transmissions, we consider multiple TCP connections rather than just one connection. Since the downstream bandwidth is shared

by all TCP connections, R_d (i.e., $R_d = \frac{C_d}{L_{data}}$) should be

replaced with $\frac{R_d}{N_{dCM}}$. We also consider the delayed

ACK policy. Thus, we multiply $\frac{1}{d}$ by the original definition of the asymmetric parameter, k , in eq. (1) to obtain eq. (6).

Substituting eqs. (4) and (5) into eq. (6), we can derive asymmetric parameter k for both cases.

3. FAST REQUEST TRANSMISSION

Based on the analysis in Sec. 2, we learned that asymmetric ratio k is high and TCP round trip delay is large when the number of simultaneous transfers is small. This phenomenon is due mainly to the control mechanism of DOCSIS. In DOCSIS, a MAP should always be sent before all requests have arrived at the CMTS. As shown in eq. (4), each CM can acquire one data grant only every other MAP period. That results in low bandwidth utilization. To solve this problem, we propose a new mechanism called "Fast Request Transmission" to keep access delay short and bandwidth efficiency high while having a fixed contention period.

A CM learns when to send a packet in the buffer from periodic MAPs. The CM cannot transmit a packet and must be waiting until its scheduled time, when a packet is really sent out, and may be piggybacked a request on the outgoing packet's header if needed. According to the analytical results, some piggybacked requests may not be granted on the next MAP. If we skip the piggyback mechanism and force the requests to the contention period directly, the requests might arrive at the CMTS earlier, at the expense of a higher collision probability. The proposed "Fast Request Transmission (FRT)" mechanism preserves the advantage of piggybacked requests while having higher bandwidth utilization.

3.1 FRT Operation

The FRT mechanism should be supported both by the CMTS and CMs, with a minor modification to the

original DOCSIS control mechanism. It works as follows.

A) CMTS operation:

The CMTS knows the difference between the transmission time and the effective time of a MAP. Thus, before a MAP is sent, the CMTS knows which data grants' piggybacked requests cannot arrive by the departure time of the next MAP. Let the number of these "late" data grants be n , a value less than or equal to N_{p_REQ} . The CMTS reserves n mini-slots in the beginning of the request contention period and describes them by unicast request IEs for each SID. The request contention period is truncated by n mini-slots instead of adding n more mini-slots. This avoids too many mini-slots for requests, which may reduce the bandwidth available for data transmission.

B) CM operation:

In DOCSIS, the priority to send a request is as follows. A CM first considers piggybacking the request on the Data Grant IE assigned on a newly received MAP, and then the unicast REQ IE, and finally the request contention broadcast REQ IE. In our mechanism, a CM also uses the Data Grant IE to piggyback the request first. If its data grant is in the end region of the MAP and the effective time of the MAP has not yet started, the CM goes to the second choice directly, which is the unicast request IE. If neither is the case, the CM will use the multicast/broadcast request IE to send the request in contention.

3.2 Analysis on FRT

In our mechanism, the CMTS can successfully receive most piggybacked requests before sending the immediate subsequent MAP. With non-zero upstream buffers, every CM can get a data grant on every MAP. Thus, no matter what the number of simultaneous transfers is, the upstream buffer service time can be given by

$$T_{usv} = (N_c + N_{dCM} N_{u_ack}) t_{ms} \quad (7)$$

Substituting eq. (7) into eq. (6), we can obtain asymmetric ratio k for FRT.

3.3 Numerical Examples and Discussion

This section used the same simulation setup as Sec. 4.1 to compare the performance of FRT and the original DOCSIS mechanism using the analytical results presented in previous sections.

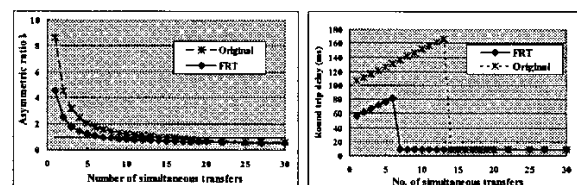


Figure 4. Asymmetric ratio k Figure 5. TCP round trip delay

Fig. 4 shows that asymmetric ratio k for both mechanisms. FRT falls below one when the number of simultaneous transfers is larger than 6. It shifts the original asymmetric boundary from 13 to 6 as compared to the original mechanism. The advantage of moving the asymmetric boundary forward is that it can have shorter TCP round trip delay as shown in Fig. 5. Thus, TCP transfers in FRT can achieve minimum delay when the number of simultaneous transfers is larger than 6, while the original DOCSIS achieves the minimum delay when the number of simultaneous transfers is 13. Meanwhile, round trip time determines the growth of the TCP sender's congestion window, which in turn determines the downstream throughput. The longer the round trip time, the lower the throughput. Thus, FRT can significant improve the downstream throughput.

4. SIMULATION

This section presents the simulation results using ns2 for the FRT mechanism and the original DOCSIS control mechanism.

4.1 Simulation Environment

The simulation parameters are described as follows. Data rate: downstream channel: 26970350 bps, upstream channel: 2560000 bps. The propagation delay from the CMTS to all cable modems is a fixed value of 0.5 ms. One mini-slot time of upstream is 50 us in size (8 times of 6.25 us). The contention period in one MAP is fixed to 50 mini-slots. A MAP cannot describe more than 2048 mini-slots or more than 240 IEs. D_{MAP} is set to 2 ms. The downstream buffer is set to 50 packets. The upstream buffer is set to 20 packets. Each CM can have one TCP transfer only. TCP data packets are of 1000 bytes and ACK packets are of 40 bytes.

4.2 Simulation Results

This experiment evaluates the performance of TCP transfers for the original DOCSIS and the FRT mechanism.

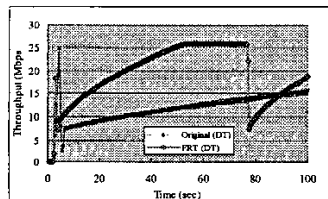


Figure 6. Downstream throughput

Fig. 6 shows the downstream throughput of a single one-way TCP transfer as a function of time. Our mechanism grows the TCP congestion window faster, thanks to having shorter upstream buffer service time to provide better downstream throughput. Fig. 7 shows the average access delays on the upstream channel

experienced by the original DOCSIS and our mechanisms. Fig. 8 shows their respective average upstream buffer sizes. Thus, FRT has smaller access delay and buffer size when the number of simultaneous transfers is small.

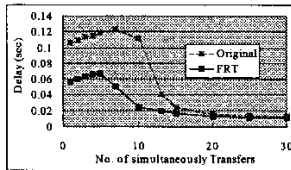


Figure 7. Access delay

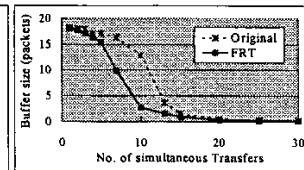


Figure 8. Upstream buffer size

5. CONCLUSION

We have studied the effect of DOCSIS MAC layer on TCP performance in CATV networks. We learned that the definition of $k = \frac{C_d}{C_u} \times \frac{L_{ack}}{L_{data}}$ in eq. (1) alone

could not adequately explain TCP behavior in DOCSIS-based HFC networks. We then modified k to be

$$k = \frac{C_d \times T_{usv}}{d \times L_{data} \times N_{dcm}}$$

considering the MAC layer operation of DOCSIS v1.1. Based the modified k , we analyzed the impact of the number of simultaneous transfers on TCP behavior, and proposed a new mechanism called "Fast Request Transmission" to improve TCP performance in HFC networks. We have also conducted simulations using network simulator ns-2 to compare the original control mechanism of DOCSIS with our mechanism. The results show that the proposed FRT mechanism has better performance in terms of aggregate downstream throughput, access delay, and buffer size.

6. ACKNOWLEDGEMENT

This work is supported by the National Science Council, Taiwan, under Grant Number NSC90-2213-E-002-119.

7. REFERENCES

- [1]. H. Balakrishnan, V. N. Padmanabhan and Randy H. Katz, "The effects of asymmetry on TCP performance," Mobile Networks and Applications 4, p219-p241, 1999.
- [2]. Yi-Der Li and Wanjiun Liao, "Improving TCP Performance over Asymmetric Networks," Proc. IEEE ICC 2001, Helsinki, Finland, June 2001.
- [3]. S. Varma, "Performance and Buffering Requirements of TCP Applications in Asymmetric Networks," Proc. IEEE INFOCOM '99.
- [4]. T. V. Lakshman, U. Madhow and Bernhard Suter, "Window-based Error Recovery and Flow Control with A Slow Acknowledgement Channel: A Study of TCP/IP Performance," IEEE INFOCOM '97.
- [5]. UCBLBNL/ISI/VINT Network Simulator - ns, Version 2, <http://www.isi.edu/nsnam/ns/>