The Behavior of TCP Over DOCSIS-Based CATV Networks

Wanjiun Liao, *Senior Member, IEEE*

***Abstract*—This paper studies the impact of the data-over-cable service interface specification (DOCSIS) media-access control (MAC) protocol on the performance of the transmission control protocol (TCP) in hybrid fiber coax (HFC) broadband access networks. We find that the asymmetry ratio expressed in existing work cannot adequately explain the behavior of TCP in DOCSIS-based networks. To better capture the effect of DOCSIS on TCP, we express the asymmetry ratio (denoted by η) in another way (denoted by k), considering the time-division multiple-access-like MAC layer operation of DOCSIS. When $\eta > 1$, TCP behaves as in a symmetric network, and when $\eta \leqq 1$, the system acts as in an asymmetric network, and the performance of TCP degrades. We find that the number of simultaneous TCP transfers significantly affects the asymmetry ratio. When the number of active transfers is below two times the maximum number of pending requests in a transmission period, the value of η is larger than one, regardless of the value of k. However, when the number of active transfers becomes very large, the effect of DOCSIS on TCP becomes negligible, and the asymmetry ratio is determined by the bandwidth ratio of the channels times the length ratio of data and acknowledgement packets. Based on η, we develop the round-trip delay of sending a data packet for both one-way and two-way transfers, and discuss the buffer requirement at the head end. The accuracy of the analytical model is validated by ns-2 simulations. The analytical result can provide useful guidelines in the design of slot allocation or scheduling mechanisms for any DOCSIS-based broadband access networks, including the emerging IEEE 802.16 WiMAX networks.**

***Index Terms*—Broadband access networks, community antenna television (CATV), data-over-cable service interface specification (DOCSIS), transmission control protocol (TCP).**

I. Introduction

THIS PAPER studies the behavior of the transmission control protocol (TCP) over data-over-cable service interface specification (DOCSIS)-based hybrid fiber coax (HFC) networks [1], a popular choice for broadband access to the Internet provided by community antenna television (CATV) operators. Unlike legacy CATV networks built with coaxial cables only, HFC networks consist of both fiber and coaxial cables. Like legacy CATV networks, the HFC network is a tree-and-branch network, as shown in Fig. 1. The tree is rooted at the cable modem termination system (CMTS) at the head end, and is connected by fiber cables to fiber nodes (FNs), from where coaxial cables are used to connect to cable modems (CMs) at the households. This HFC architecture makes HFC systems more robust, and provides better signal quality and higher channel capacity than legacy CATV networks.

Paper approved by I. Andonovic, the Editor for Optical Networks and Devices of the IEEE Communications Society. Manuscript received March 1, 2004; revised September 12, 2005. This work was supported in part by the National Science Council of Taiwan, under a Center of Excellence Grant NSC94-2752-E-002-006-PAE and under Grant NSC94-2218-E-002-059.

The author is with the Department of Electrical Engineering and the Graduate Institute of Communication Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C. (e-mail: wjliao@ntu.edu.tw).

Digital Object Identifier 10.1109/TCOMM.2006.878833

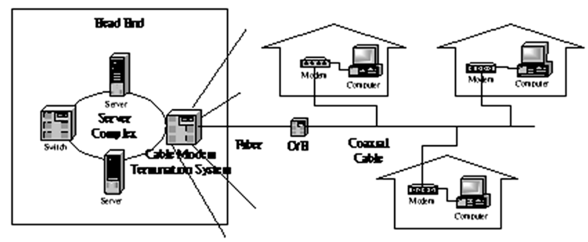

Fig. 1. Tree-and-branch topology of the HFC network.

In HFC networks, each branch is comprised of a downstream channel and an upstream channel. The downstream channel is a point-to-multipoint broadcast channel, and the upstream channel is a multipoint-to-point shared channel. The only transmitter on the downstream channel is the CMTS at the head end, and all the CMs connected to the channel are receivers. For the upstream channel, all the CMs are transmitters, and the CMTS is the only receiver. To arbitrate random access to the upstream channel, a media-access control (MAC) mechanism is required. The MAC mechanism in HFC may follow the DOCSIS of the Multimedia Cable Network System (MCNS) Partners [1] or IEEE 802.14 [2]. Since DOCSIS is the *de facto* standard in the cable industry, we will focus on DOCSIS in this paper.

In DOCSIS, the upstream channel is modeled as frames of minislots. Each frame is comprised of contention minislots and data minislots, plus some management minislots, as shown in Fig. 2(a). The detailed minislot allocation of each frame is specified via a control message, called MAP, which is periodically transmitted by the CMTS on the downstream channel [Fig. 2(b)]. In each MAP, a variable number of information elements (IEs) are specified, each IE describing a range of minislots to be used by a CM in the next frame. The number of IEs, and the corresponding minislots, varies from MAP to MAP. Typically, each MAP contains a Request IE, some Data Grant IEs, a Null IE, and some management information. The Request IE specifies the contention interval of the next frame; each Data Grant IE describes the transmission interval for one CM, and the Null IE terminates the assignment list. Each MAP describes the details of the next frame to be transmitted upstream, and must be received by all CMs before the next frame starts, as in all reservation-based MAC mechanisms.

Each time a CM has a data packet to transmit, it requests the assignment of a Data Grant IE in a later MAP. If the CM is backlogged (i.e., a CM with a nonempty packet queue) and is

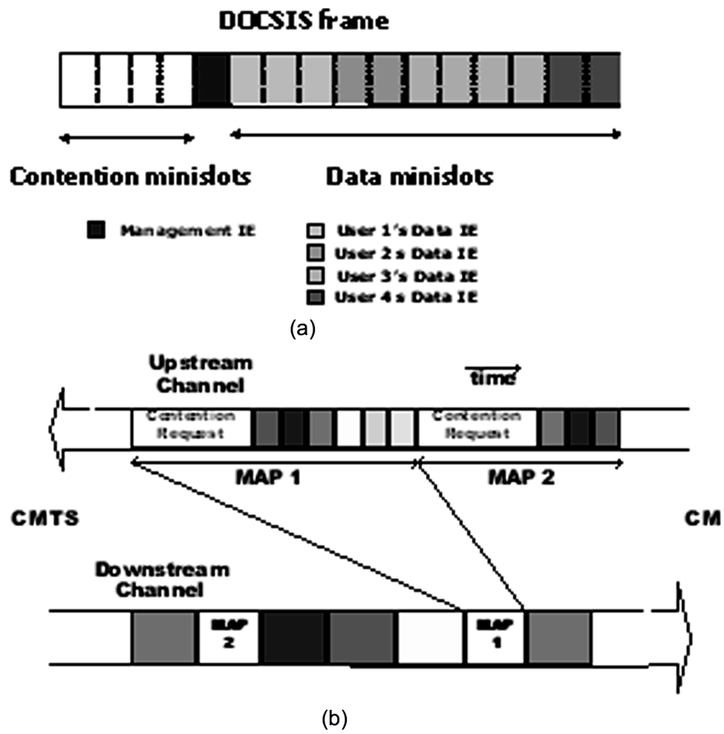

Fig. 2. Operation of DOSCIS MAC mechanism. (a) DOCSIS frame. (b) MAP control message.

assigned a Data Grant IE in the newly received MAP, its request is piggybacked on an outgoing packet using the assigned Data Grant IE in the data minislots; otherwise, the CM contends for the use of the upstream channel via contention minislots. Each contention request occupies one minislot. If several CMs compete for one minislot in the contention interval, all the requests are corrupted due to collisions. After sending a request, the CM waits for a Data Grant IE or a Data Pending IE in the next MAP. Once either is received, the contention resolution is complete. Each CM can use the Data Grant IE to transmit a data packet if assigned, or keep waiting for a Data Grant IE and send no further request if a Data Pending IE is received. The CM regards the contention request as lost if neither a Data Grant IE nor a Data Pending IE is received. The CM then increases the backoff window (i.e., contention window) by a factor of two, as long as the window size is less than the maximum backoff window. The CM randomly selects a value within its new backoff window, pauses until the timer expires, and then repeats the contention process. This retry process continues until the maximum number of retries is reached, at which time the packet must be discarded.

The design issues for HFC networks have been widely studied for years. An efficient MAC layer scheduling and allocation algorithm for the upstream channel of HFC networks is proposed in [3]–[6]. The performance of HFC networks is evaluated in [7]–[9]. The flow control mechanism of TCP is tuned in [10]–[13] to improve the performance of TCP over asymmetric networks. The performance of TCP in HFC networks is investigated in [14]–[17]. The impact of DOCSIS on the performance of TCP is studied in [18] and [19].

The previous studies [10]–[13] have shown that the performance degradation of TCP over asymmetric networks is due to bandwidth asymmetry of downstream and upstream channels. This paper also investigates the performance degradation problem of TCP, but focuses on DOCSIS-based HFC networks, an asymmetric network which employs a MAC mechanism to arbitrate random access among multiple hosts. We discuss the impact of DOCSIS's MAC mechanism on the performance of TCP, and demonstrate that the result provided by existing work cannot adequately explain the behavior of TCP in such an environment. We then define an asymmetry ratio for DOCSIS systems and analyze the behavior of TCP over DOCSIS based on this ratio. We examine both one-way and two-way TCP transfers in the network, and analyze the throughput and delay performance of the system. The accuracy of the model is validated by ns-2 simulations. The simulation results support our analytical results. To the best of our knowledge, this is the first paper providing a thorough theoretical study on the performance of TCP over DOCSIS-based networks. The analytical result can provide useful guidelines in the design of scheduling mechanisms for DOCSIS-based systems [14]. The proposed analytical model can also be used to evaluate and compare the performance of resource management mechanisms designed for any DOCSIS-based broadband access networks, including the emerging IEEE 802.16 WiMAX networks.

The rest of the paper is organized as follows. Section II describes the performance problem of TCP over DOCSIS-based HFC networks and defines an asymmetry ratio for DOCSIS systems. Section III analyzes the behavior of one-way TCP transfers over the MAC layer of DOCSIS. Section IV analyzes two-way TCP transfers over the DOCSIS MAC layer. Section V shows simulation results to verify the analysis. Finally, the concluding remarks are given in Section VI.

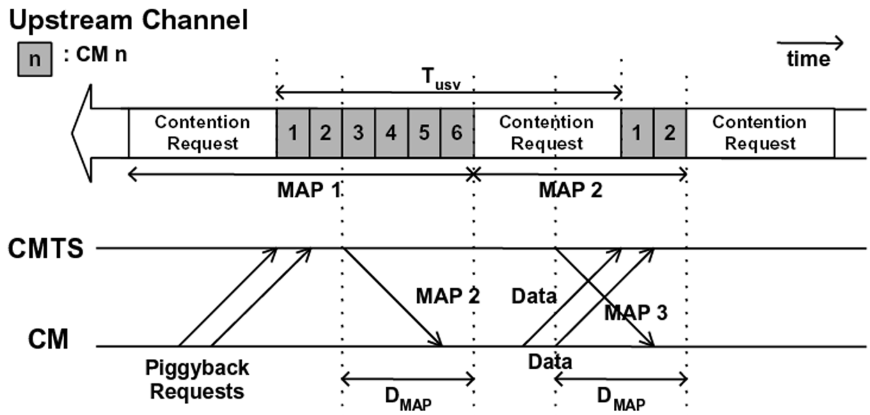

Fig. 3. Illustration of DOCSIS. The requests piggybacked on data IEs are sent by CMs; a MAP message is sent by the CMTS D_{MAP} seconds before the next frame starts. T_{usv} denotes the delay between sending two consecutive packets from the upstream buffer of each CM.

II. Problem Statement

In HFC networks, one CMTS controls multiple branches. Each branch is composed of one downstream channel and one upstream channel, and is typically connected to 200–500 CMs in the neighborhood. Since each branch works identically but independently, without loss of generality, we consider only one branch in the analysis. This analysis is based on TCP Reno, plus the delayed acknowledgement (ACK) mechanism, for congestion control.

A. Notation and Assumptions

Some terms are defined to facilitate explanation in the paper.

1) The delivery offset of the MAP (denoted by D_{MAP}) is the one-way propagation delay between the CMTS and CMs plus some extra delay, as shown in Fig. 3. This amount of time is spent for the CMTS to inform all CMs of the slot allocation details for the next frame. Thus, each MAP should be sent D_{MAP} seconds before the next frame starts.
2) Upstream user service time (denoted by T_{usv}) is the delay between sending two consecutive data packets from the upstream buffer of one CM, as shown in Fig. 3.
3) Access delay (denoted by T_{ad}) is defined as the time between when a packet is received by a CM and when the CM starts sending this packet out to the channel.

The notations used in the analysis are summarized as follows.

C_d	Capacity of the downstream channel, in b/s.
C_u	Capacity of the upstream channel, in b/s.
N_{data}	Data packet length, in bits.
N_{ack}	ACK packet length, in bits.
B_{CM}	Buffer size (i.e., backlog) at a CM, in terms of number of packets.
T	One-way propagation delay between the CMTS and CMs, in seconds.
$N_{d\mathrm{CM}}$	Number of CMs performing TCP transfers in the downstream direction.
$N_{u\mathrm{CM}}$	Number of CMs performing TCP transfers in the upstream direction.
t_{ms}	One minislot time, in seconds.
N_{frame}	Total number of minislots granted in a MAP for a frame.
N_c	Number of contention minislots described in a MAP.
N_{u_ack}	Number of minislots for one ACK packet.
N_{u_data}	Number of minislots for one TCP data packet.
T_{ad}	Access delay, in seconds.
T_{usv}	Upstream user service time, in seconds.
D_{MAP}	Delivery offset of a MAP, in seconds.
N_{p_REQ}	Maximum number of pending requests received during a D_{MAP}.

We make the following assumptions.

1) We consider TCP traffic only in the analysis. The result can serve as the lower bound of the performance when user datagram protocol (UDP) and TCP traffic coexists on the channel.
2) The propagation delay between the CMTS and all CMs are assumed the same.
3) In DOCSIS, multiple outstanding MAPs are allowed. Here we restrict the number of outstanding MAPs to one, so as to simplify the analysis. Nevertheless, the analytical approach is applicable to the case with multiple MAPs outstanding.
4) The number of contention minislots in each DOCSIS frame, i.e., N_c, is a constant.
5) The contention minislots in each frame is assumed larger than the delivery offset of the MAP, i.e., $N_c \times t_{\mathrm{ms}} > D_{\mathrm{MAP}}$.

B. Effect of Bandwidth Asymmetry on Performance of TCP

Asymmetric networks, such as HFC and xDSL, are networks with different channel bandwidths in the downstream and upstream directions. The main effect of bandwidth asymmetry on the performance of TCP is that the ACK clocking may be disrupted [10]–[13]. To better understand the behavior of TCP in asymmetric networks, an asymmetry ratio k [10] is defined by

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

TABLE I
PARAMETER SETTINGS

Parameter	Value
C_d	26.97 Mbps
C_u	2.56 Mbps
T	0.5 ms
D_{MAP}	2 ms
t_{ms}	50 μs
N_c	50 (mini-slots)
MAP limit	2048 mini-slots and 240 IEs
L_{data}	1024 bytes
L_{ack}	64 bytes
B_u	20 (packets)
d	2

The physical meaning of this ratio is explained as follows. TCP behaves normally when k is less than or equal to one. When this ratio exceeds one, the network is asymmetric. Operating TCP over an asymmetric network makes ACK packets arrive on the bottleneck link in the reverse direction at a rate faster than the bottleneck link can support, and leads to two consequences. First, the time spacing between consecutive ACKs received by the sender is larger than that in the symmetric case. As a result, the sender clocks out data at a slower rate, and slows down the growth of the congestion window. This, in turn, degrades the throughput in the downstream direction. Second, the buffer to the reverse bottleneck link is rapidly filled up with ACK packets. A full buffer may cause serious ACK drops. This may lead to a bursty sender, because one ACK may acknowledge several TCP data packets in an unpredictable way. In addition, fewer ACK packets received may slow down the growth of the sender's congestion window, because TCP increases its congestion window by counting the number of ACKs received.

C. Asymmetry Ratio for DOCSIS Systems

To examine the adequacy of the expression k in (1) for describing the behavior of TCP over DOCSIS-based HFC networks, we estimate the performance of TCP using the parameters listed in Table I. The bandwidths of the downstream and upstream channels are set to 26.97 and 2.56 Mb/s, respectively. Considering the delayed ACK policy of TCP Reno (i.e., sending one ACK packet to acknowledge the receipt of d data packets), the expression of k is modified as

$$k = \frac{C_d}{C_u} \times \frac{L_{\text{ack}}}{L_{\text{data}}} \times \frac{1}{d}.$$

Using the packet lengths of ACK $= 64$ B and Data $= 1024$ B, and $d = 2$, we obtain

$$k = \frac{26.97}{2.56} \times \frac{64}{1024} \times \frac{1}{2} \cong 0.33.$$

Since $k \leqq 1$, this network is supposed to be symmetric for TCP, i.e., TCP should behave normally.

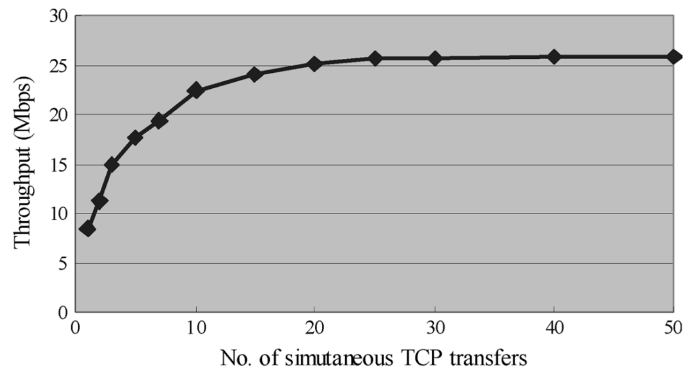

Fig. 4. Throughput versus number of TCP transfers.

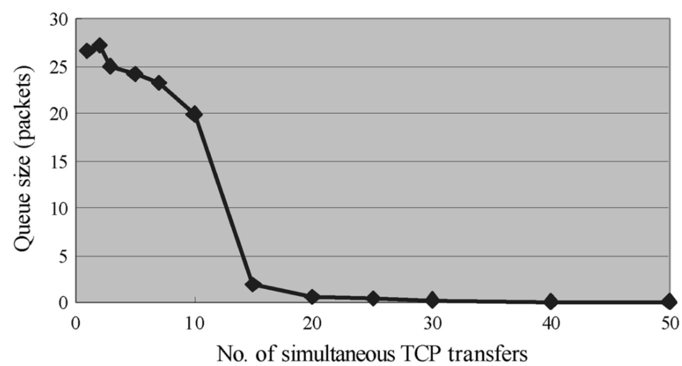

Fig. 5. Backlog at the CM versus number of TCP transfers.

We then verify the behavior of TCP in such a network by simulation using ns-2 [18], again based on the parameters listed in Table I. We observe the influence of different number of downloading CMs (i.e., $N_{d\text{CM}}$) on the aggregate downstream throughput and the backlog in the upstream buffer (which buffers ACK packets) at the CM. Fig. 4 plots the aggregate downstream throughput of the network, varying $N_{d\text{CM}}$ from 1 to 50, and Fig. 5 shows the respective buffer size (i.e., backlog).

Interestingly, we find that when $N_{d\text{CM}}$ is small (e.g., less than 13 in this example), the network behaves like an asymmetric network, i.e., low downstream throughput and full upstream ACK queues. However, as $N_{d\text{CM}}$ becomes large (e.g., exceeds 13 in this example), TCP starts behaving normally. This implies that the network experiences bandwidth asymmetry (i.e., the asymmetry ratio is greater than one) when $N_{d\text{CM}}$ is small, and the performance degrades in this region. However, according to the estimation based on k in (1), the network should operate symmetrically. Thus, we learn that while k in (1) captures well the behavior of TCP on point-to-point asymmetric links, it cannot adequately explain the behavior of TCP operating over DOCSIS-based HFC networks, an asymmetric network which needs a MAC mechanism to arbitrate random access among end hosts. To better capture the abnormal behavior of TCP over DOCSIS-based HFC networks, we need to consider the minislot reservation-based random access in the upstream direction of DOCSIS systems.

We then revise the asymmetry ratio k in (1) as in (2) for DOCSIS networks, considering the MAC layer operation of DOCSIS.

$$\eta = \frac{C_d \times T_{\text{usv}}}{d \times L_{\text{data}} \times N_{d\text{CM}}}. \tag{2}$$

This new ratio is obtained as follows. Considering the upstream channel in HFC networks as a time-division multiple-access (TDMA)-like channel, $(L_{\text{ack}})/(C_u)$ in (1) is replaced with T_{usv}, where T_{usv} is the duration that a CM should wait for its scheduled time. For downstream transmissions, $(C_d)/(L_{\text{data}})$ in (1) is shared by all simultaneous TCP transfers, and should be replaced by $(C_d)/(L_{\text{data}} \times N_{d\text{CM}})$. We also consider the delayed ACK policy. Thus, we multiply $1/d$ by $(C_d \times T_{\text{usv}})/(L_{\text{data}} \times N_{d\text{CM}})$ to obtain η.

In the next two sections, the impact of the DOCSIS MAC mechanism on the performance of TCP will be studied with respect to the round-trip delay (RTT) of each TCP transfer. This RTT, in turn, determines the throughput of each TCP connection in DOCSIS networks.

III. One-Way TCP Transfers

We first examine one-way TCP transfers, i.e., all CMs download TCP traffic on the downstream channel. Thus, all data packets go on the downstream channel, and all ACK packets go on the upstream channel.

A. *Effect of DOCSIS MAC on Bandwidth Asymmetry*

In DOCSIS, the MAP for the next frame should be received by all CMs before the frame starts. This may cause some requests, especially piggybacked ones, in the current frame to arrive and be processed at the CMTS after the MAP for the next frame has been sent. The late requests will be pending and become backlogs for the second frame. These pending requests plus new requests which arrive at the CMTS during the next frame will be waiting to be granted in the next MAP. Each MAP can describe at most 2048 minislots. Compared with the MAP limitation of 2048 minislots, the ACK packet length is relatively small. To simplify the analysis, we assume that the minislot limitation of each MAP will not be exceeded when the CMTS grants all the waiting requests. In fact, even if the limitation is exceeded, the analytical approach is still applicable, because the global ordering of data grants placed in frames remains intact. Recall that the contention minislot interval is assumed larger than the delivery offset of each MAP. Thus, the pending requests will be granted in the next MAP, not in later MAPs. Since the packet length of each ACK is fixed, the maximum number of pending requests, N_{p_REQ} (i.e., those arriving at the CMTS during a D_{MAP}), in terms of number of minislots, can be expressed by

$$N_{p\_\text{REQ}} = \left\lfloor \frac{D_{\text{MAP}}}{t_{\text{ms}}} \times \frac{1}{N_{u\_\text{ack}}} \right\rfloor.$$

The value of T_{usv} can be calculated in two mutually exclusive, collectively exhaustive cases.

Case 1: $N_{d\text{CM}} \le 2N_{p\_\text{REQ}}$

When $N_{d\text{CM}} \le N_{p\_\text{REQ}}$, no piggybacked requests are granted in the next MAP. On average, each CM receives one data grant in every other frame, i.e., there are $N_{d\text{CM}}$ requests granted every two frames. Thus, T_{usv} can be expressed by

$$T_{\text{usv}} = (2N_c + N_{d\text{CM}} N_{u\_\text{ack}})t_{\text{ms}} \tag{3}$$

where $N_{u\_\text{ack}} = \lceil (L_{\text{ack}})/(C_u) \times (1)/(t_{\text{ms}}) \rceil$.

Case 2: $N_{d\text{CM}} > 2N_{p\_\text{REQ}}$

If $N_{d\text{CM}} > 2N_{p\_\text{REQ}}$, some CMs may obtain their data grants immediately in the next frame. When piggybacked requests arrive at the CMTS before the next MAP has been sent, these requests will be scheduled after those pending requests arriving in the current frame. These "lucky" piggybacked requests are scheduled behind the pending requests in the next frame. Those requests placed in the last N_{p_REQ} data grants of the next frame become the pending requests of the second frame. As such, all CMs take turns as the "lucky" ones in each frame, and the average data grants each CM can obtain in every other frame is more than one. Thus, T_{usv} can be expressed by

$$T_{\text{usv}} = \frac{[N_c + (N_{d\text{CM}} - N_{p\_\text{REQ}})N_{u\_\text{ack}}]t_{\text{ms}} N_{d\text{CM}}}{N_{d\text{CM}} - N_{p\_\text{REQ}}}. \tag{4}$$

Substituting (3) and (4) into (2), we can obtain the asymmetry ratio η as follows:

1) $N_{d\text{CM}} \le 2N_{p\_\text{REQ}}$

$$\eta = \frac{C_d}{d \times L_{\text{data}}} \times \frac{(2N_c + N_{u\_\text{ack}} N_{d\text{CM}})t_{\text{ms}}}{N_{d\text{CM}}} \tag{5}$$

2) $N_{d\text{CM}} > 2N_{p\_\text{REQ}}$

$$\eta = \frac{C_d}{d \times L_{\text{data}}} \frac{[N_c + (N_{d\text{CM}} - N_{p\_\text{REQ}})\, N_{u\_\text{ack}}]\; t_{\text{ms}}}{N_{d\text{CM}} - N_{p\_\text{REQ}}}. \tag{6}$$

Note that since only ACK packets are transmitted on the upstream channel, all upstream packets have the same length. As a consequence, the number of minislots for each data grant (i.e., N_{u_ack}) is fixed, and the number of pending requests (i.e., N_{p_REQ}) is also a fixed value. In addition, we fix the number of contention minislots (i.e., N_c). Given the downstream and upstream channel capacities and TCP packet length, the asymmetry ratio η varies only with the number of CMs transferring simultaneously (i.e., $N_{d\text{CM}}$). Furthermore, from (5) and (6), η decreases as $N_{d\text{CM}}$ increases. This implies that as the number of simultaneous transfer increases, the network asymmetry decreases, matching the observation we made in Figs. 2 and 3. From (6), when $N_{d\text{CM}}$ becomes very large, η will be approximated by $(C_d)/(d \times L_{\text{data}}) \times t_{\text{ms}} N_{u\_\text{ack}}$. Substituting $N_{u\_\text{ack}} = \lceil (L_{\text{ack}})/(C_u) \times (1)/(t_{\text{ms}}) \rceil$ into this approximation, we obtain

$$\eta = \frac{C_d}{d \times L_{\text{data}}} \times \frac{L_{\text{ack}}}{C_u}. \tag{7}$$

Interestingly, η in (7) is equal to k in (1),[1] just as if the DOCSIS MAC mechanism had not been taken into account. Therefore, we can conclude that when the number of simultaneous transfers becomes very large, the effect caused by the DOCSIS MAC mechanism on TCP becomes negligible. In this case, the value of η is determined by the asymmetry ratio of the pair of channels and the packet size ratio of TCP to ACK packets. However, when there is only one CM performing TCP transfers in the downstream direction, the asymmetry ratio reaches its maximum value. This maximum value of η may be much larger than one, rendering the bandwidths rather asymmetric.

B. Effect of DOCSIS MAC on TCP RTT

We now analyze the RTT of sending one TCP packet in DOCSIS systems. To send an ACK packet, the CM needs to send a request first and wait for a data grant allocated by the CMTS. The average round-trip time is then given by

$$\mathrm{RTT} = 2T + \frac{L_{\mathrm{data}}}{C_d} + \frac{L_{\mathrm{ack}}}{C_u} + T_{\mathrm{ad}}. \tag{8}$$

Access delay T_{ad} for a packet is defined as the time between when the packet is received by the CM from the upper layer and when the CM starts sending it out. The packet may experience a queueing delay if the buffer at the CM is nonempty. Each time the CM wants to send an ACK packet, it will issue a request. Then, it will wait for the MAP that contains the data grant specifying the period of time it is allowed to send. This needs two times more than the propagation delay between the CMTS and the CM (i.e., one propagation delay for the request, one propagation delay for the MAP, plus the processing delay at the CMTS). Later, at the scheduled time, it will send the head-of-line packet from the buffer.

The asymmetry ratio η greatly affects the value of T_{ad}. If $\eta > 1$, the upstream ACK buffer will always stay full. All requests experience long queuing delay and are sent piggybacked. If $\eta \leq 1$, requests will often find an empty queue, and will be sent via contention. In the following, we calculate T_{ad} in these two cases.

Case 1: Asymmetry Ratio $\eta > 1$: When η is larger than one, the upstream buffer will build up rapidly with ACK packets and stay full constantly. Due to the piggyback mechanism, the order of data grants placed in the consecutive frames remains intact until at least one CM has no more packets to send. The distribution of the upstream user service time for each packet will almost always remain unchanged, and each packet in the upstream buffer experiences a delay of $(B_{\mathrm{CM}} \times T_{\mathrm{usv}})$. Thus, the access delay can be expressed by

$$T_{\mathrm{ad}} = B_{\mathrm{CM}} \times T_{\mathrm{usv}} \tag{9}$$

where B_{CM} is the upstream buffer size (i.e., backlog) at the CM, in terms of the number of ACK packets.

[1]Here we mean (1) with the consideration of delayed ACK, i.e., $k = (C_d)/(C_u) \times (L_{\mathrm{ack}})/(L_{\mathrm{data}}) \times (1)/(d)$.

Substituting (9) into (8), we obtain the TCP RTT in the asymmetric case.

Case 2: Asymmetry Ratio $\eta \leq 1$: If $\eta \leq 1$, the upstream channel is not the bottleneck anymore, and the upstream buffer will not stay full. The buffer size may drop to zero rapidly, and the requests for sending ACK packets may not be piggybacked to the CMTS. Assume that there is no collision in the contention minislot. The mean waiting time of a request is about half a frame time. When a request arrives at the CMTS, it should wait until the next MAP is issued, at which time the data grant will be assigned. Later, the CM will receive the MAP with its data grant assigned. It will then keep waiting until its turn to send. On average, this waiting time is about another half a frame time. Thus, the average access delay on the upstream channel can be approximately given by two frame times, i.e.,

$$T_{\mathrm{ad}} = 2N_{\mathrm{frame}} t_{\mathrm{ms}}. \tag{10}$$

The total number of minislots in a frame is determined as follows. ACK packets use only a portion of bandwidth on the upstream channel, $(L_{\mathrm{ack}})/(d \times L_{\mathrm{data}})C_d$. The remaining bandwidth is used by the contention minislots. The average number of minislots used by ACK packets is then given by $(N_c \times (L_{\mathrm{ack}})/(d \times L_{\mathrm{data}})C_d)/(C_u - (L_{\mathrm{ack}})/(d \times L_{\mathrm{data}})C_d)$. Thus, on average, the total number of minislots in a frame is expressed by

$$N_{\mathrm{frame}} = N_c + \frac{N_c \times \frac{L_{\mathrm{ack}}}{d \times L_{\mathrm{data}}} C_d}{C_u - \frac{L_{\mathrm{ack}}}{d \times L_{\mathrm{data}}} C_d}. \tag{11}$$

Substituting (11) in (10) and then in (8), we obtain the TCP RTT in the symmetric case.

To reflect the impact of these two cases of η on RTT, the expression in (8) can be revised as

$$\mathrm{RTT} = w \times \mathrm{RTT}_a + (1 - w) \times \mathrm{RTT}_s \tag{12}$$

where RTT_a is the RTT calculated based on $\eta > 1, \mathrm{RTT}_s$ is based on $\eta \leq 1$, and w is a weighting factor with

$$w = \begin{cases} 1, & \eta > 2 \\ 1 - e^{\frac{1-\eta}{t}}, & 1 < \eta \leq 2 \\ 0, & \eta < 1. \end{cases}$$

The interpretation of this setting will be described later in the performance evaluation section.

IV. Two-Way TCP Transfers

In this section, we study two-way TCP transfers over DOCSIS-based HFC networks. With two-way transfers, some clients download TCP traffic (i.e., do TCP transfers in the downstream direction), and some upload TCP traffic (i.e., transfer TCP traffic in the upstream direction). Thus, both data and ACK packets can go in either direction.

A. *Effect of DOCSIS MAC on Bandwidth Asymmetry*

Since the network has asymmetric bandwidth and different operations for both channels, downloading CMs experience network asymmetry differently from uploading CMs. In what follows, we will discuss the asymmetry ratios for clients doing TCP transfers in these two directions.

1) Downstream TCP Transfers: With two-way TCP transfers, both data and ACK packets can be transmitted on the upstream channel. These two kinds of packets are very different in size. ACK packets are fixed and small, and TCP data packets are variable and large. Typically, the data packet is ten times larger than the ACK packet. Thus, the data packet transmission time $(L_{\rm data})/(C_u)$ may be larger than the delivery offset of a MAP $D_{\rm MAP}$. When a frame ends with a data grant for a long data packet, it is very likely that no pending requests will arrive at the CMTS in the frame. However, in practice, data grants may be ordered in a frame rather randomly. Consequently, it is hard to accurately determine the number of pending requests in each frame. We can only conclude that if there are pending requests, they are usually from ACK packets.

Moreover, both data and ACK packets can be transmitted in the downstream direction. Only a portion of the downstream capacity is used to transmit data packets. Considering the "delayed ACK" policy, the asymmetry ratio of downstream TCP transfers can be expressed by

$$\eta = \frac{N_{d{\rm CM}}L_{\rm data}}{N_{d{\rm CM}}L_{\rm data} + \frac{N_{u{\rm CM}}L_{\rm ack}}{d}} \times \frac{C_d}{d \times L_{\rm data}} \times \frac{T_{\rm usv}}{N_{d{\rm CM}}}. \quad (13)$$

The first term in (13) [i.e., $(N_{d{\rm CM}}L_{\rm data})/(N_{d{\rm CM}}L_{\rm data} + (N_{u{\rm CM}}L_{\rm ack})/(d))$] is close to one, because 1) the ACK packet length is relatively small and 2) the "delayed ACK" policy is used to reduce the number of ACK packets to be sent. Thus, transmitting ACK packets in the downstream direction has a very minor impact on η, due to only a small portion of the channel capacity being used for ACK packets. The MAP limitation of 2048 minislots, however, may easily be exceeded in the two-way transfer case. Fortunately, the MAP limitation has a very minor impact on η, thanks to the global ordering of data grants specified by MAPs remaining intact. The only difference is that the data grants unable to fit in one MAP should be put in the next MAP. To save space, we assume this limitation is not exceeded, although the analytical approach is still applicable to handle the case when it is exceeded.

Based on the analysis in Section III-A [i.e., (5) and (6)], η is highly dependent on the relationship between $N_{d{\rm CM}}$ and $2N_{p\_{\rm REQ}}$. In addition, the backlog in the upstream buffer is usually nonzero in the two-way transfer case. Due to the uncertain number of pending requests, we can only determine the upper and lower bounds of η, instead of the deterministic value, as in the one-way case. Using the same approach as in Section III-A, we can derive the two bounds of $T_{\rm usv}$ for downstream transfers in two cases.

Case 1: $N_{d{\rm CM}} \le 2N_{p\_{\rm REQ}}$

First, we determine the asymmetry ratio η for the case of $N_{d{\rm CM}} \le 2N_{p\_{\rm REQ}}$, regardless of the number of uploading CMs. The worst case $T_{\rm usv}$ of downloading CMs occurs when all data grants for ACK packets are placed in the end of a frame. In this case, the piggybacked requests of downloading CMs are all pending. Thus, the worst-case $T_{\rm usv}$ of downloading CMs is given by

$$T_{\rm usv} = (2\,N_c + 2N_{u{\rm CM}}N_{u\_{\rm data}} + N_{d{\rm CM}}N_{u\_{\rm ack}}) \times t_{\rm ms} \quad (14)$$

where

$$N_{u\_{\rm data}} = \left\lceil \frac{L_{\rm data}}{C_u} \times \frac{1}{t_{\rm ms}} \right\rceil \quad \text{and} \quad N_{u\_{\rm ack}} = \left\lceil \frac{L_{\rm ack}}{C_u} \times \frac{1}{t_{\rm ms}} \right\rceil.$$

The best-case $T_{\rm usv}$ of downstream TCP transfers occurs when the data grant of a TCP data packet is put in the end of a frame, and the data packet length is long enough to satisfy the condition of $N_{u\_{\rm data}} \times t_{\rm ms} > D_{\rm MAP}$. In this case, all piggybacked requests can obtain their data grants in the next MAP. Thus, the best-case $T_{\rm usv}$ is given by

$$T_{\rm usv} = (N_c + N_{u{\rm CM}}N_{u\_{\rm data}} + N_{d{\rm CM}}N_{u\_{\rm ack}}) \times t_{\rm ms}. \quad (15)$$

Substituting (14) and (15) in (13), we can obtain the two bounds of η accordingly.

Case 2: $N_{d{\rm CM}} > 2N_{p\_{\rm REQ}}$

We determine η for the case of $N_{d{\rm CM}} > 2N_{p\_{\rm REQ}}$, regardless of the number of uploading CMs. The worst-case $T_{\rm usv}$ of downstream TCP transfers occurs when more than $N_{p\_{\rm REQ}}$ data grants for ACK packets are put in the end of a frame. Thus, the worst-case $T_{\rm usv}$ of downloading CMs is

$$T_{\rm usv} = \frac{[N_c + N_{u{\rm CM}}N_{u\_{\rm data}} + (N_{d{\rm CM}} - N_{p\_{\rm REQ}})N_{u\_{\rm ack}}]t_{\rm ms}N_{d{\rm CM}}}{(N_{d{\rm CM}} - N_{p\_{\rm REQ}})}. \quad (16)$$

As in Case 1, the best-case $T_{\rm usv}$ occurs when the data grant of a long data packet is put in the end of a frame, and the packet size is long enough to satisfy the condition of $N_{u\_{\rm data}} \times t_{\rm ms} > D_{\rm MAP}$. Thus, the best-case $T_{\rm usv}$ is given by

$$T_{\rm usv} = (N_c + N_{u{\rm CM}}N_{u\_{\rm data}} + N_{d{\rm CM}}N_{u\_{\rm ack}})t_{\rm ms}. \quad (17)$$

Substituting (16) and (17) in (13), the two bounds of η can be obtained accordingly.

Data grants placed in a frame may be rather random. Suppose that no pending requests are caused by the MAP limitation. Once a long data grant (i.e., satisfying $N_{u\_{\rm data}} \times t_{\rm ms} > D_{\rm MAP}$) is placed in the end of a frame, all requests which arrive in the frame will be able to obtain their data grants in the next MAP. Thus, we conclude that while the worst case may happen, the network tends to operate in the best case in the steady state. We will demonstrate such an effect in the simulation section.

When the numbers of downloading and uploading CMs are equal and very large, the lower bound of η can be approximated by

$$\begin{aligned}\eta &\cong \frac{N_{d{\rm CM}}L_{\rm data}}{N_{d{\rm CM}}L_{\rm data} + \frac{N_{d{\rm CM}}L_{\rm ack}}{d}} \times \frac{C_d}{d \times L_{\rm data}} \\ &\quad \times \frac{(N_{d{\rm CM}}N_{u\_{\rm data}} + N_{d{\rm CM}}N_{u\_{\rm ack}})t_{\rm ms}}{N_{d{\rm CM}}} \\ &= \frac{C_d}{d \times C_u}. \end{aligned} \quad (18)$$

In this case, η just reflects the inherent bandwidth asymmetry in HFC networks.

2) Uploading TCP Transfers: From the viewpoint of uploading CMs, the upstream user service rate (i.e., the inverse of the upstream user service time) is their packet transmission rate. In addition, the upstream service rates (in terms of packets/s) of uploading CMs are equal to those of downloading CMs. Thus, the ratio η of uploading CMs is given by

$$\eta = \frac{1}{d \times T_{\text{usv}}} \times \frac{N_{u\text{CM}} \times L_{\text{ack}}}{C_d} \times \frac{N_{d\text{CM}} L_{\text{data}} + \frac{N_{u\text{CM}} L_{\text{ack}}}{d}}{N_{u\text{CM}} L_{\text{ack}}} \tag{19}$$

where T_{usv} is the upstream user service time of the uploading CM.

Multiplying η in (19) with η in (13) yields a constant of $1/d^2$. When the asymmetry ratio of the downloading CM increases, the asymmetry ratio of the upstream CM is decreased. Due to the nature of bandwidth asymmetry in HFC networks, downstream TCP transfers usually have much larger asymmetry ratios.

B. Effect of Bandwidth Asymmetry on TCP RTT

The relationship between the TCP RTT and the asymmetry ratio for two-way transfers is the same as the one-way case [see (8)–(11)], except that the downloading CMs mostly operate as in asymmetric networks. The average RTT of sending a packet in the two-way case is given by $\text{RTT} = 2T + T_{\text{trans}} + T_{\text{ad}}$, where

$$T_{\text{trans}} = \begin{cases} \frac{L_{\text{data}}}{C_d} + \frac{L_{\text{ack}}}{C_u}, & \text{for downstream traffic} \\ \frac{L_{\text{data}}}{C_u} + \frac{L_{\text{ack}}}{C_d}, & \text{for upstream traffic.} \end{cases} \tag{20}$$

The long RTT experienced by downloading CMs in the two-way case is due to long TCP packets going upstream. Long RTT, in turn, makes the TCP congestion window grow very slowly. In other words, the longer the RTT, the lower the downstream throughput. Thus, the two-way transfer case has lower downstream throughput, as compared with the one-way transfer.

V. Performance Evaluation

In this section, we provide numerical examples and simulation results using ns-2 to verify the analytical results presented in the previous sections. The parameter settings in this section are listed in Table I: the upstream and downstream channel capacities are 26.97 and 2.56 Mb/s, respectively; the data and ACK packets' lengths are 1024 and 64 B, respectively, and the delayed ACK factor is set to 2 in the simulation; the upstream buffer can hold 20 packets; MAP limits are 2048 minislots and 240 IEs, and the number of contention minislots is set to 50.

A. Numerical Examples

1) One Way TCP Transfer: We first examine the analysis results for one-way TCP transfers based on Table I. Fig. 6 plots η as a function of $N_{d\text{CM}}$. When $N_{d\text{CM}}$ is less than 13, η is larger than one; otherwise, η is less than or equal to one. This implies TCP has worse performance when $N_{d\text{CM}}$ is small. In the extreme case, when there is only one TCP transfer, η reaches its maximum value and the TCP transfer results in the worst

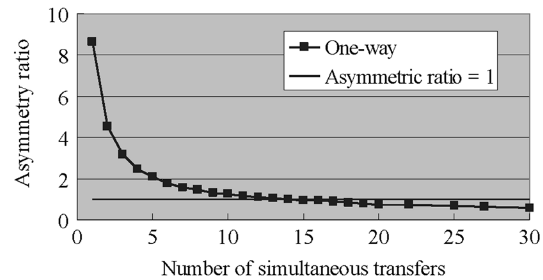

Fig. 6. Asymmetry ratio η as a function of $N_{d\text{CM}}$ in one-way transfers.

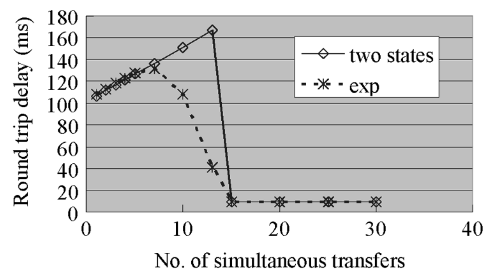

Fig. 7. TCP RTT versus the number of simultaneous transfers based on different settings on w in (12).

performance. Fig. 7 shows the relationship between the TCP RTT and the number of simultaneous transfers. The solid curve is plotted based on RTT in (8), and the dashed curve is from RTT in (12). It is observed that the asymmetry ratio η has significant impact on TCP RTT. When $\eta > 1$, the RTT is on the order of 100 ms, and when $\eta \leq 1$, the RTT is on the order of 10 ms. The large RTT for $\eta > 1$ is due to the long waiting time in the upstream buffer at CMs. The linear growth of the curve as $\eta > 1$ is due to the increasing total number of minislots granted by the MAP. The curve stays flat as $\eta \leq 1$, invariant to the value of $N_{d\text{CM}}$.

We see that the delays of the solid curve are strictly classified into two categories, i.e., an increasing curve (as $\eta > 1$) and a flat curve (as $\eta \leq 1$), due to the result of the setting $w = 1$ when $\eta > 1$, and $w = 0$ when $\eta \leq 1$. In other words, there is a sharp drop in the curve at $N_{d\text{CM}} = 13$ (i.e., $\eta = 1$). In reality, the behavior of TCP should be similar on either side of the point of $\eta = 1$ (e.g., 1.000001 and 0.999999). This phenomenon can also be explained by Fig. 6. When Fig. 6 is rotated counterclockwise by 90°, the curve of the asymmetry ratio demonstrates an exponential-like growth. This implies that the RTT value should gradually decrease, rather than increase, when η approaches one and $1 < \eta \leq 2$. With the exponential setting, the RTT curve decreases smoothly as η approaches one, better matching our expectation.

2) Two-Way TCP Transfer: We fix the total number of simultaneous transfers at 30. Let $N_{u\text{CM}}$ out of 30 clients transfer TCP in the upstream direction, and the rest transfer downstream. The other parameters are based on Table I.

The asymmetry ratio η and the TCP RTT of downloading CMs are plotted in Figs. 8 and 9, respectively. Fig. 8 shows that η of downloading CMs increases sharply as the number of simultaneous uploading CMs increases. η is too high to tolerate

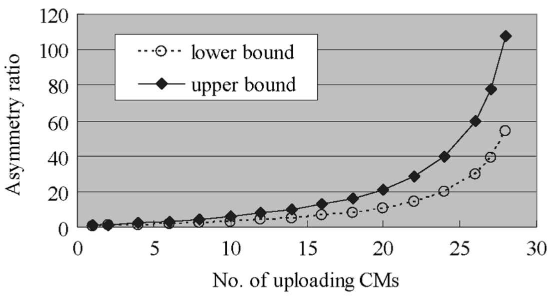

Fig. 8. Bounds of asymmetry ratio η in two-way transfers. These bounds correspond to the best and the worst case of minislot allocation for packets in each DOCSIS frame.

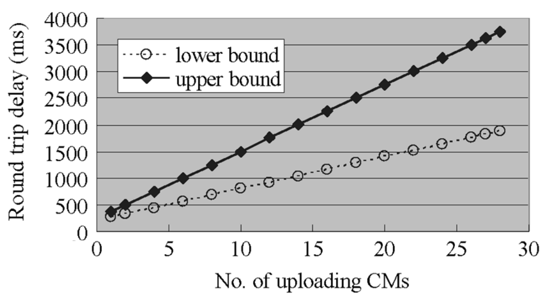

Fig. 9. Bounds of TCP RTT in two-way transfers based on the bounds of η.

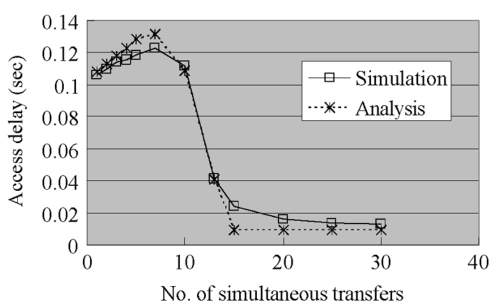

Fig. 10. Access delay for one-way transfers.

when the number of simultaneous uploading CMs is beyond 20, i.e., two-thirds of clients are uploading. Interestingly, η exceeds one so long as the number of simultaneous transfers is larger than one. This means even with just a few clients uploading, all downloading clients in the network suffer from bandwidth asymmetry. Fig. 9 shows that the TCP RTT of downloading CMs grows linearly as the number of simultaneous uploading CMs increases. The large RTT is due to a very large η, which is much larger than one, and increases exponentially as the number of uploading CMs increases. Compared with one-way transfers (see Fig. 7), Fig. 9 has a much longer delay. This is because long packets (TCP packets) are transferred on the bottleneck channel, causing long waiting times for short packets (ACK packets).

B. Analysis vs. Simulation

In this section, the analytical results are compared with the simulation results. Figs. 10 and 11 show the access delays of downloading users for one-way and two-way transfers, respectively. The solid curves are from the simulation, and the dashed curves are from the analysis (U is for the upper bound, and L is

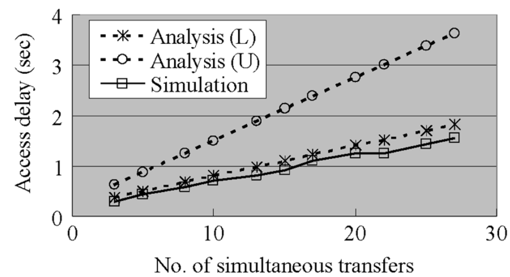

Fig. 11. Access delay for two-way transfers.

for the lower bound). We can see that in the one-way case, the two curves stay very close; in the two-way case, the simulation curve is very close to the lower bound of the analytical curve, matching our argument in Section IV-A.

VI. Conclusion

In this paper, we have studied the impact of DOCSIS MAC layer operation on the performance of TCP over HFC networks. We learned that the asymmetry ratio k expressed in [10] cannot adequately explain the behavior of TCP in DOCSIS-based HFC networks. To better capture the effect of DOCSIS, we express the asymmetry ratio in another way, denoted by η, considering the TDMA-like MAC layer operation of DOCSIS. When $\eta > 1$, TCP behaves as in a symmetric network; when $\eta \leq 1$ the system acts as in an asymmetric network, and the performance of TCP degrades. We find that the number of simultaneous transfers significantly affects the asymmetry ratio. If the number of active CMs is below two times the maximum number of pending requests in a transmission period, the value of η is larger than one, regardless of the value of k. However, as the number of active CMs becomes very large, the effect of DOCSIS on TCP becomes negligible, and the asymmetry ratio is determined by the bandwidth ratio of channels times the length ratio of data and ACK packets.

In our analysis, both one-way and two-way transfers are considered. With two-way transfers, the downloading CMs mostly experience network asymmetry, but operate in the lower bound of η in the steady state. Based on η, the RTT of sending a data packet is developed for both one-way and two-way transfers, and the buffer requirement at the CMTS is discussed. We have also conducted simulations to verify the analytical results. The results show a good match between the simulation and the analysis. To the best of our knowledge, this is the first paper providing a theoretical study on the performance of TCP over DOCSIS-based networks. The analytical result can provide useful guidelines in the design of slot allocation or scheduling mechanisms for DOCSIS-based systems. More importantly, the proposed analytical model can be used to evaluate the performance of resource management mechanisms designed for any DOCSIS-based broadband access networks, including the emerging IEEE 802.16 WiMAX networks.

References

[1] *CableLabs, Data-Over-Cable Service Interface Specifications—Radio Frequency Interface Specification*, SP-RFIv1.1-106-001215, MCNS Consortium, 2000.

[2] *IEEE Project 802.14/a Draft 3 Rev. 1*, , IEEE, Apr. 1998.

[3] M. Droubi, N. Idirene, and C. Chen, "Dynamic bandwidth allocation for the HFC DOCSIS MAC protocol," in *Proc. IEEE IC3N*, 2000, pp. 54–60.
[4] D. Sala, J. O. Limb, and S. U. Khaunte, "Adaptive control mechanism for cable modem MAC protocols," in *Proc. IEEE INFOCOM* , 1998, pp. 1392–1399.
[5] Y.-D. Lin, C.-Y. Huang, and W.-M. Yin, "Allocation and scheduling algorithms for IEEE 802.14 and MCNS in hybrid fiber coaxial networks," *IEEE Trans. Broadcast.*, vol. 44, no. 4, pp. 427–435, Dec. 1998.
[6] D. Sala and J. O. Limb, "Comparison of contention resolution algorithms for a cable modem MAC protocol," in *Proc. Broadband Commun.*, 1998, pp. 83–90.
[7] V. Sdralia, C. Smythe, P. Tzerefos, and S. Cvetkovic, "Performance characterisation of the MCNS DOCSIS 1.0 CATV protocol with prioritised first come first served scheduling," *IEEE Trans. Broadcast.*, vol. 45, no. 2, pp. 196–209, Jun. 1999.
[8] T. Orfanoudakis, N. Leligou, E. Meciu, and A. Harsanyi, "Evaluation of IP oriented HFC access protocols," in *Proc. Broadband Commun.*, 2000, pp. 133–138.
[9] M. T. Ali, R. Grover, G. Stamatelos, and D. D. Falconer, "Performance evaluation of candidate MAC protocols for LMCS/LMDS networks," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 7, pp. 1261–1270, Jul. 2000.
[10] H. Balakrishnan, V. N. Padmanabhan, and R. H. Katz, "The effects of asymmetry on TCP performance," *Mobile Netw. Appl. 4*, pp. p219–p241, 1999.
[11] Y.-D. Li and W. Liao, "Improving TCP performance over asymmetric networks," in *Proc. IEEE ICC*, Helsinki, Finland, Jun. 2001, pp. 1824–1828.
[12] S. Varma, "Performance and buffering requirements of TCP applications in asymmetric networks," in *Proc. IEEE INFOCOM*, New York, NY, 1999, pp. 1548–1555.
[13] W. Liao and H.-J. Ju, "Adaptive slot allocation in DOCSIS-based CATV networks," *IEEE Trans. Multimedia*, vol. 6, no. 6, pp. 479–488, Jun. 2004.
[14] T. V. Lakshman, U. Madhow, and B. Suter, "Window-based error recovery and flow control with a slow acknowledgement channel: A study of TCP/IP performance," in *Proc. IEEE INFOCOM*, 1997, pp. 1199–1209.
[15] R. Cohen and S. Ramanathan, "TCP for high performance in hybrid fiber coaxial broad-band access networks," *IEEE/ACM Trans. Netw.*, vol. 6, no. 1, pp. 15–29, Feb. 1998.
[16] O. Elloumi, N. Golmie, H. Afifi, and D. Su, "A study of TCP dynamics over HFC networks," in *Proc. IEEE GLOBECOM*, 1998, pp. 545–550.
[17] R. Cohen and S. Ramanathan, "Using proxies to enhance TCP performance over hybrid fiber coaxial networks," *Comput. Commun.*, vol. 20, pp. 1502–1518, 1998.
[18] I. M. C. Tam, M. K. Patnam, K. T. Chan, and F. S. Fook, "Performance evaluation of web browsing over hybrid fiber coaxial broadband networks," in *Proc. IEEE ICON*, 1999, pp. 372–382.
[19] UCB/LBNL/ISI/VINT Network Simulator-ns ver. 2 [Online]. Available: http://www.isi.edu/nsnam/ns/.

Wanjiun Liao (M'97–SM'06) received the B.S. and M.S. degrees from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1990 and 1992, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1997.

She joined the Department of Electrical Engineering, National Taiwan University (NTU), Taipei, Taiwan, as an Assistant Professor in 1997. Since August 2005, she has been a Full Professor. Her research interests include wireless networks, multimedia networks, and broadband access networks.

Dr. Liao is currently an Associate Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and the IEEE TRANSACTIONS ON MULTIMEDIA. She was a tutorial Co-Chair of IEEE Infocomm 2004, and the TPC Vice-Chair of IEEE Globecom 2005 (the Autonomous Network Symposium). She has received many research awards. Two papers she co-authored with her students received the Best Student Paper Award of the First IEEE International Conferences on Multimedia and Expo (ICME) in 2000, and the Best Paper Award of the First International Conference on Communication, Circuits and Systems in 2002. She was a recipient of the K. T. Li Young Research Award given by the ACM Taipei chapter in 2003 for her research achievement. She was elected as one of Ten Distinguished Young Women in Taiwan in 2000, and was listed in *Who's Who* (New Providence, NJ: Marquis) in 2001–2005, and the *Contemporary Who's Who* in 2003.