

# Adaptive Slot Allocation in DOCSIS-Based CATV Networks

Wanjiun Liao, *Member, IEEE*, and Huei-Jiun Ju

**Abstract**—This paper proposes an adaptive slot allocation mechanism to improve the performance of Transmission Control Protocol (TCP) in DOCSIS-based hybrid fiber coaxial (HFC) networks. The proposed mechanism is comprised of two parts: fast request transmission (FRT) and long packet deferment (LPD). FRT is designed to handle one-way TCP transfers, while LPD targets two-way transfers. Here “one-way transfers” means all active cable modems perform downloading; while “two-way transfers” indicates some perform downloading and some perform uploading. We analyze the proposed mechanism and conduct simulations using network simulator *ns-2* to evaluate the performance of our mechanism. The results show that the proposed mechanism has better performance in terms of aggregate downstream throughput, access delay, and required buffer size, as compared to the original control mechanism of DOCSIS.

**Index Terms**—Asymmetric network, DOCSIS, HFC, slot allocation, TCP.

## I. INTRODUCTION

CABLE TV networks are a promising solution of providing broadband access to the home. Cable operators are upgrading their cable plants into two-way hybrid fiber coax (HFC) networks. The Data-Over-Cable Service Interface Specifications (DOCSIS) [1] by the Multimedia Cable Network System Partners (MCNS), and IEEE 802.14 [2] are major standards developed to enable data communication over HFC networks and to ensure the interoperability of conforming products built by different vendors. HFC access networks allow data to be transported at a rate of tens of megabits per second, thereby offering several hundredfold increase in access rate compared to conventional dial-up modems.

An HFC network has a tree-and-branch network topology. Each branch is comprised of a downstream channel and an upstream channel. The downstream channel is a point-to-multipoint, broadcast channel, while the upstream channel is a multipoint-to-point, bus channel. The only transmitter on the downstream channel is the Cable Modem Termination System (CMTS) at the Head End, and all the cable modems (CMs) connected to the channel are receivers. For the upstream channel, all the CMs are the transmitters and the CMTS is the only

receiver. CMs in HFC cannot detect transmission collisions on the upstream channel as in the Ethernet. To arbitrate random access to the upstream channel, a contention-based reservation control mechanism plus a truncated binary exponential backoff scheme are used. The capacities of downstream and upstream channels in HFC networks are asymmetric. Typically, the downstream bandwidth is about 30 Mbps, while the upstream bandwidth is about 3 Mbps.

In this paper, we focus only on MCNS's DOCSIS. DOCSIS specifies the physical layer modulation and the MAC layer operation for HFC networks. In DOCSIS, the upstream channel is modeled as a stream of minislots, the use of which is controlled by the CMTS. To send packets on the upstream channel, each CM needs to request a slot, and waits for a reply of slot allocation from the CMTS. The CMTS periodically transmits downstream management messages called upstream bandwidth allocation (denoted as MAP), which defines how the upstream bandwidth is allocated in a transmission cycle (i.e., frame). Each MAP contains a variable number of information elements (IEs), each of which defines a range of minislots to be used. The number of IEs, and the corresponding minislots, varies from MAP to MAP. At the maximum, a MAP is bounded by 240 IEs. At the minimum, a MAP must contain two IEs: a Request IE and a Null IE. Typically, each MAP contains a Request IE, some Data Grant IEs, and a Null IE. The Request IE defines the request contention period for the next transmission cycle, the Data Grant IE defines data periods for CMs to transmit packets, and the Null IE terminates the assignment list. In DOCSIS, multiple outstanding MAPs are allowed, but the total number of minislots accounting for all MAPs is limited to 4096. Each time a new request arrives at the CMTS, it is queued. When it comes time to generate a new MAP for the next transmission cycle, the CMTS will process and grant the requests in the queue on a FCFS manner until the limits of minislots and IEs are reached. Each CM can have one or more 14-bit Service Identification (SID) and a 48-bit address, but at most one outstanding request for each SID at a time. In DOCSIS, the bandwidth allocation and transmission scheduling schemes are left open to vendors. Typically, the CMTS employs simple first-come-first-served (FCFS) scheduling to process requests.

The design issues of HFC networks have been widely studied for years. An efficient MAC layer scheduling and allocation algorithm is designed in [3]–[6] for the upstream channel of HFC networks. 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].

Manuscript received February 2, 2002; revised September 5, 2002. This work was supported in part by the MOE program for Promoting Academic Excellence of Universities under Grant 89E-FA06-2-4-7, and in part by the National Science Council of Taiwan, R.O.C., under Grant NSC 93-2213-E-002-001. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Anna Hac.

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

Digital Object Identifier 10.1109/TMM.2004.827498

In this paper, a new adaptive slot allocation mechanism is proposed to improve the performance of TCP in HFC networks. Our mechanism is comprised of two parts: fast request transmission (FRT) and long packet deferment (LPD). FRT is designed to handle one-way TCP transfers (i.e., data packets downstream and ACK packets upstream), while LPD targets two-way transfers (i.e., both data and ACK packets can go in either direction). We analyze the proposed mechanism and conduct simulations using network simulator *ns-2* [18] to evaluate the performance of our mechanism.

The rest of the paper is organized as follows. Section II analyzes the performance of TCP over DOCSIS. Section III presents the proposed adaptive slot allocation mechanism to improve the performance of TCP over DOCSIS-based HFC networks. Section IV shows the simulation results to compare the original DOCSIS's MAC layer control mechanism with our mechanism. Finally, the concluding remark is included in Section V.

## II. PERFORMANCE OF TCP OVER DOCSIS

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 the performance of TCP is that TCP ACK clocking may be disrupted. To better understand the behavior of TCP in asymmetric networks, an asymmetry ratio is defined as the ratio of the capacity of the system to send data to the capacity of the system to send the corresponding ACK. When the asymmetry ratio is smaller than or equal to one, TCP operates in a symmetric network and behaves normally (i.e., small backlog at the CM and high aggregate throughput). When the ratio is larger than one, 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 and slows down the growth of the congestion window, which in turn lowers the throughput in the downstream direction.

In [19], the asymmetry ratio for DOCSIS-based HFC networks is expressed as follows, considering the MAC layer operation of DOCSIS v1.1, i.e.,

$$\eta = \alpha \times \frac{C_d \times T_{usv}}{d \times L_{data} \times N_{dCM}} \quad (1)$$

$$\text{where } \alpha = \begin{cases} 1, & \text{for one-way transfers} \\ \frac{N_{dCM} L_{data}}{N_{dCM} L_{data} + \frac{N_{uCM} L_{ack}}{d}}, & \text{for two-way transfers.} \end{cases}$$

where  $d$  is a parameter of the delayed ACK policy (i.e., sending one ACK packet to acknowledge the receipt of  $d$  data packets),  $T_{usv}$  is defined as the average time between sending two consecutive packets in the CM buffer,  $N_{dCM}$  is the number of simultaneous TCP downloading, and  $N_{uCM}$  is the number of simultaneous TCP uploading.

In DOCSIS, the upstream channel is modeled as a stream of minislots. A transmission starts only in the beginning of any minislot. Let  $N_{u\_ack}$  be the number of minislots used to transmit one ACK packet upstream. Given  $L_{ack}$ ,  $C_u$ , and  $t_{ms}$ , we can

derive the number of minislots 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 minislot on the upstream channel, and  $C_u$  is the upstream channel capacity. Similarly, we can derive the number of minislots used to transmit a data packet as  $N_{u\_data} = \lceil L_{data}/C_u \times 1/t_{ms} \rceil$ , where  $L_{data}$  is the size of a data packet.

A MAP describes how the upstream bandwidth is allocated to CMs in a transmission cycle, and should be received by all CMs before its effective time. Each MAP may be transmitted before some requests, especially piggybacked ones, have arrived and been processed at the CMTS. The late requests are deemed pending in the current transmission cycle and become backlogged requests in the next cycle. These pending requests plus new requests which arrive at the CMTS during the next transmission cycle will wait to be granted in the next MAP. Let  $D_{MAP}$  be the time difference between when a MAP is transmitted and when it goes into effect. Therefore, the maximum number of pending requests,  $N_{p\_REQ}$ , (i.e., those who have arrived at the CMTS during a  $D_{MAP}$ .) can be expressed as

$$N_{p\_REQ} = \left\lfloor \frac{D_{MAP}}{t_{ms}} \times \frac{1}{N_{u\_ack}} \right\rfloor. \quad (3)$$

Let  $T_{usv}$  denote the average time between sending two consecutive packets in the buffer of the CM. We calculate the two bounds of  $T_{usv}$  for downstream transfers in the following two cases.

1) When  $N_{dCM} \leq 2N_{p\_REQ}$ ,  $T_{usv}$  is bounded by

$$(N_c + N_{uCM} N_{u\_data} + N_{dCM} N_{u\_ack}) t_{ms} \leq T_{usv} \leq (2N_c + 2N_{uCM} N_{u\_data} + N_{dCM} N_{u\_ack}) t_{ms}, \quad (4)$$

where  $N_c$  is the number of minislots allocated to the contention period.

2) When  $N_{dCM} > 2N_{p\_REQ}$ ,  $T_{usv}$  is bounded by

$$(N_c + N_{uCM} N_{u\_data} + N_{dCM} N_{u\_ack}) t_{ms} \leq T_{usv} \leq \frac{[N_c + N_{uCM} N_{u\_data} + (N_{dCM} - N_{p\_REQ}) N_{u\_ack}] t_{ms} N_{dCM}}{(N_{dCM} - N_{p\_REQ})}. \quad (5)$$

Substituting (4) and (5) into (1), we can derive the two bounds of  $\eta$  for the two cases accordingly.

## III. ADAPTIVE SLOT ALLOCATION MECHANISM FOR DOCSIS

In this section, we describe the proposed mechanism for DOCSIS-based CATV networks. Our mechanism is comprised of two parts: FRT and LPD. FRT is designed to handle one-way TCP transfers, while LPD targets two-way transfers. In the following, we first describe FRT, and then LPD. We then present an integrated mechanism of FRT and LPD to conclude this section.

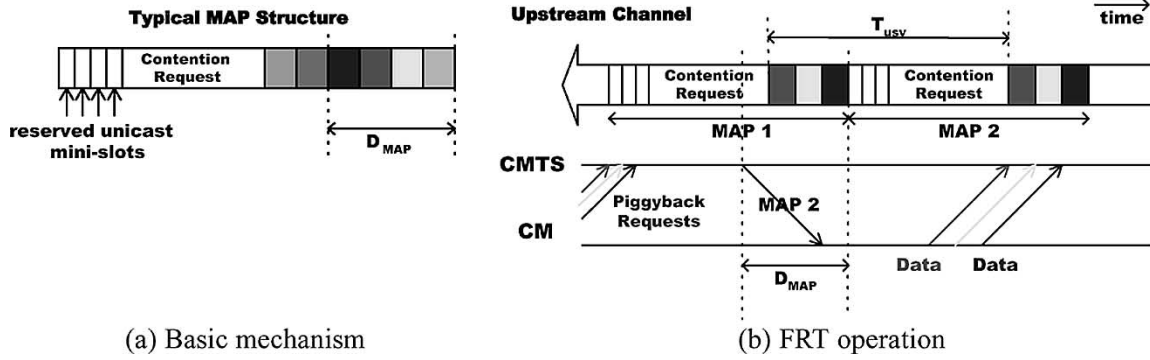


Fig. 1. FRT mechanism. (a) Basic mechanism (b) FRT operation.

#### A. Fast Request Transmission

From the analysis in Section II, we learn that asymmetry ratio is high when the number of simultaneous transfers is small. This phenomenon is mainly due to the control mechanism of DOCSIS. In DOCSIS, a MAP is sent before all requests have arrived at the CMTS. As shown in [19], each CM can obtain only one data grant in every other transmission cycle when the number of active CMs is small. This results in low bandwidth utilization on the upstream channel. One way to improve the utilization is to shorten the contention period. However, it is hard to accurately predict the exact length of the contention period. The number of simultaneous transfers varies from time to time. If the contention period is set too short, many collisions among requests may occur, resulting in very long and unstable access delay. It is also a challenge to balance between bandwidth efficiency and access delay if an adaptive contention period is used. 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 then waits until its turn, when a packet will be transmitted and piggybacked a request in the header if necessary. According to the analytical results in [19], some piggybacked requests may not be granted in the next MAP. If we skip the piggyback mechanism and force the requests to go to the contention period directly, the requests might arrive at the CMTS earlier, but at the expense of a higher collision probability. In the following, we demonstrate how FRT preserves the advantage of piggybacked requests while having higher bandwidth utilization.

1) *FRT Operation*: The FRT mechanism should be supported by both the CMTS and CMs, and needs a minor modification to the operation of the original DOCSIS control mechanism. It works as follows (as shown in Fig. 1).

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 before 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  $n$  minislots at the front of the contention period and describes them by using unicast request IEs to each SID. The contention period is reduced by  $n$  minislots instead of adding  $n$  more minislots. This

avoids reserving too many minislots for requests, thus reducing the bandwidth available for data transmission.

b) *CM operation*: In DOCSIS, there are three ways for a CM to send a request, as follows. It first considers piggybacking the request on the Data Grant IE assigned in a newly received MAP, then it may send an unicast REQ IE, and finally, it may send a contention broadcast REQ IE. In our mechanism, a CM also tries to piggyback the request on the Data Grant IE first. If its data grant is in the backend 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 case applies, the CM will use the multicast/broadcast request IE to send the request via contention.

Note that when the CM finds its data grant placed in the backend of the MAP, and the new transmission cycle has not yet started, it puts the “piggybacked” request in the reserved minislot in the frontend of the MAP. However, if there currently are no packets waiting in the buffer, but after a while, a new packet comes before the start time of its data grant, but after the start of the reserved minislot, the CM just sends the request via piggyback as in the original DOCSIS.

2) *Analysis of FRT*: In our mechanism, the CMTS can successfully receive most of the piggybacked requests before sending the next MAP. A CM with nonzero upstream buffer gets a data grant in every MAP. Thus, no matter how many simultaneous transfers there are, the upstream user service time is given by

$$T_{usv} = (N_c + N_{dCM}N_{u\_ack})t_{ms}. \quad (6)$$

The average total number of minislots in a MAP frame is

$$N_{MAP\_ms} = N_c + N_{dCM}N_{u\_ack}. \quad (7)$$

Substituting (6) into (1), we can obtain the asymmetry ratio  $\eta$  for FRT as

$$\eta = \frac{C_d}{d \times L_{data}} \times \frac{(N_c + N_{dCM}N_{u\_ack})t_{ms}}{N_{dCM}}. \quad (8)$$

Comparing (8) with (4) and (5) in [19], we learn that  $\eta$  is smaller for the FRT mechanism when  $N_{dCM}$  is a small fixed value. A smaller  $\eta$  renders higher throughput and lower round trip delay for a connection [19]. Therefore, our mechanism has higher throughput and shorter TCP round trip delay, thanks to shorter  $T_{usv}$ . Furthermore, a smaller  $\eta$  can make the value of  $\eta$  go down to one earlier as the number of simultaneous transfers increases.

TABLE I  
PARAMETERS USED

PARAMETER	VALUE
$C_d$	26.97 Mbps
$C_u$	2.56 Mbps
$T$	0.5 ms
$D_{MAP}$	2 ms
$t_{ms}$	50 $\mu$ s
$N_c$	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

Thus, the system can enter the symmetric situation earlier, at which time the TCP round trip delay will reach the minimum.

3) *Numerical Examples and Discussion:* Table I lists the parameters used to evaluate the performance of FRT and the original DOCSIS mechanism based on the analytical results presented in previous sections.  $L_{data}$  and  $L_{ack}$  denote TCP and ACK packet sizes, respectively. Each packet includes a payload (i.e., 1000 bytes and 40 bytes for data and ACK packets, respectively), an Ethernet header (18 bytes) and a DOCSIS header (6 bytes). Using the values listed in Table I, we obtain  $N_{u-ack}$  as 5 (minislots) and  $N_{p-REQ}$  as 8.

We first look at the original DOCSIS mechanism. Fig. 2 shows that  $\eta$  decreases as the number of simultaneous transfers increases. Asymmetry ratio  $\eta$  exceeds one when the number of simultaneous transfers is less than 13; otherwise,  $\eta$  is less than or equal to one. This implies the performance of TCP becomes worse when the number of simultaneous transfers is small. In the extreme case, if there is only one TCP transfer, asymmetry ratio  $\eta$  reaches its maximum and CMs have the worst performance. Fig. 3 shows the TCP round trip delay as a function of the number of simultaneous transfers. It can be observed that the asymmetry ratio  $\eta$  has significant impact on TCP round trip delay. When  $\eta > 1$ , the round trip delay is of the order of 100 ms, and when  $\eta \leq 1$ , the round trip delay is of the order of 10 ms. The linear growth of the curve for  $\eta > 1$  results from more minislots being allocated in a MAP. The curve stays flat for  $\eta \leq 1$ , invariant with the number of simultaneous transfers.

We now examine the FRT mechanism. Fig. 2 shows that the asymmetry ratio  $\eta$  falls below one when the number of simultaneous transfers is larger than six. The asymmetry boundary has been shifted from 13 to six as compared to the original mechanism. Fig. 3 shows the advantage of moving the asymmetry boundary forward on TCP round trip time. As mentioned in the previous section, each time a packet is sent upstream, a CM waits for an access delay. If  $\eta$  is larger than one, the upstream buffer will be filled with ACKs rapidly and incur extremely long access delay, resulting in long round trip time. In FRT, TCP transfers can achieve minimum delay when the number of simultaneous transfers exceeds six, instead of 13. Meanwhile, round trip time determines the growth of the TCP sender's congestion window, which in turn determines downstream throughput. The longer the round trip time, the lower

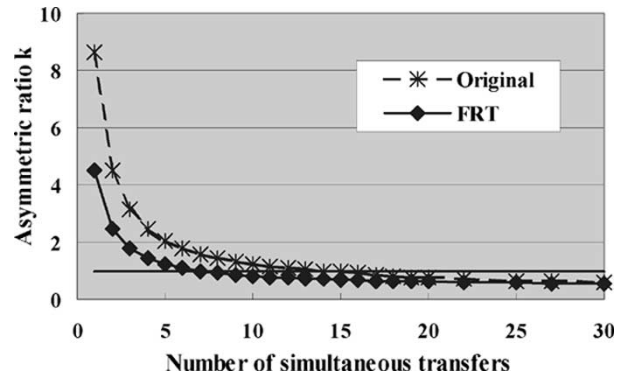


Fig. 2. Asymmetry ratio for one-way transfers.

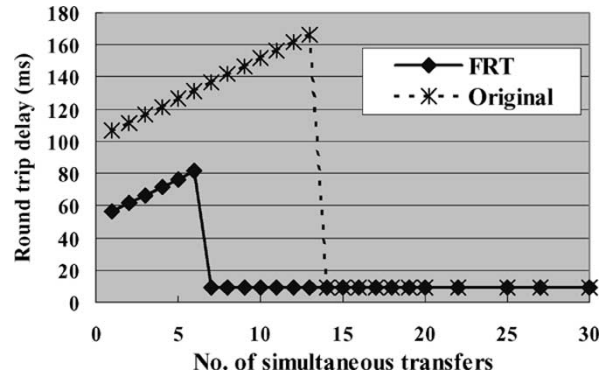


Fig. 3. Round trip delay for one-way transfers.

the throughput. Thus, our mechanism has significant improvement on the downstream throughput. This will be demonstrated in Section IV.

### B. Long Packet Deferment

In DOCSIS, the CMTS allocates at most one data grant to each CM in a MAP, irrespective of the number of minislots requested. The design philosophy behind “one-CM-one-Data-IE” in a MAP is to fairly share the channel bandwidth, i.e., no one will monopolize all of the bandwidth in a transmission cycle. Such fairness, however, holds only for one-way TCP transfers. With two-way TCP transfers, both data and ACK packets can be transmitted upstream. Typically, TCP packets are ten times larger than ACK packets. Thus, they need different sizes of data grants to be allocated by the CMTS. With two-way transfers, the upstream TCP traffic (i.e., long data packets) might throttle the downstream traffic (i.e., short ACK packets). The reasons are described as follows.

- 1) Allowing data packets (i.e., long packets) going upstream may cause high asymmetry ratios to downloading CMs, and thus, long round trip time to ACK packets (i.e., short packets). Long round trip delay in turn reduces the growth rate of the downloading CM's congestion window. This is because the congestion window of the TCP sender grows at a rate inversely proportional to average round trip delay.
- 2) The “one-CM-one-Data-IE” allocation of DOCSIS leads to the same round trip delay for packets of both upstream and downstream transfers. As a consequence, the congestion windows of TCP senders in both transfer direc-

tions grow at approximately the same rate. This results in low bandwidth utilization on the downstream channel because the downstream channel typically has far higher capacity than the upstream one. If the TCP senders of both transfer directions grow their congestion windows at the same rate, the bandwidth on the downstream channel will be underutilized.

In summary, DOCSIS treats TCP data packets in the same way as ACK packets on the upstream channel. Each CM is allowed to send at most one data grant in every transmission cycle irrespective of the frame size. This causes the same round trip delays for packets of both downstream and upstream transfers, and results in poor bandwidth utilization on the downstream channel. To solve this problem, we propose LPD, which treats long packets (data packets) differently from short packets (ACK packets) on the upstream channel. The design goal of the LPD mechanism is to reduce the sending rates of long packets and increase those of short packets, in an attempt to achieve true fairness in resource sharing. Consequently, we can shorten the round trip delay of downstream TCP transfers without seriously degrading the performance of upstream TCP transfers. Note that the LPD mechanism is designed to operate at the CMTS only. No modification is required to CMs.

1) *LPD Fundamentals:* Here we assume all data packets are of fixed size, and will extend the discussion to packets with variable size in Section III-B3. All we need to do is distinguish between two types of requests (i.e., for long and short data grants). Let  $\delta$  be the threshold to determine the type of a request, where  $\delta = N_{u\_data} - N_{u\_ack}/2$ . Intuitively, if the data grant size requested exceeds the threshold  $\delta$ , the request should be deferred a few more steps (i.e., transmission cycles) before being granted. Let  $N_{def}$  be the number of transmission cycles a long packet should be deferred (called “deferment step” in the rest of the paper). Each downloading CM (i.e., those transferring short ACK packets) can get a data grant in every MAP, but each uploading CM (those transferring long data packets) can only get one data grant in every  $N_{def}$  MAPs.

Suppose that the CMTS has two types of queues to store requests: a long job queue and a short job queue. Upon receipt of a new request, the CMTS processes the request as follows. If the data grant size of a request is larger than  $\delta$ , the CMTS initializes the request’s deferment step to  $N_{def}$  and puts it into the long job queue; otherwise, the number of deferment steps is set to one and the request is put into the short job queue. Later when it comes to the transmission time of the next MAP, the CMTS will start processing requests from the short job queue, followed by the long job queue, all on a first-come-first-served basis. The CMTS allocates data grants only to those requests with a number of deferred steps less than or equal to one, and removes the allocated requests from the respective queues. For those with larger-than-one deferment steps, the CMTS decrements the values of their deferment steps, issues data pending IEs for them in the MAP, and puts them back into the queues. This process continues until the limitation of a MAP is reached (2048 minislots and 240 IEs), when the CMTS will stop allocating data grants and decrementing the deferment steps. It will then start issuing each of the remaining eligible requests (i.e., those with a “less than or equal to one” deferment step) a Data

Pending IE in the MAP because the CMs should be notified that their requests are pending, not lost.

2) *Analysis of the LPD Mechanism:* The LPD mechanism makes downloading CMs experience different round trip delays from uploading CMs. The ratio of the upstream packet sending rates of these two types of CMs is  $N_{def} : 1$ . In addition, LPD forces the CMTS to place long data grants in the backend of each MAP because it always starts allocation from the short job queue. Thus, the LPD mechanism keeps the asymmetry ratio and TCP round trip delay at the lower bound [of (4) or (5)] when there is any long data grant allocated in a MAP.

a) *Asymmetry ratio:* The LPD mechanism causes different round trip delays to downstream and upstream TCP transfers. We derive the upstream user service times for both downloading and uploading CMs as follows.

1) For downloading CMs (i.e., ACK packet),

$$T_{usv} = \left( N_c + \frac{N_{uCM}}{N_{def}} \times N_{u\_data} + N_{dCM} N_{u\_ack} \right) t_{ms}. \quad (9)$$

Since the number of minislots for a long data grant (for TCP packet) is much larger than the number of minislots for a short data grant (for ACK packet),  $N_{u\_data} \gg N_{u\_ack}$  and  $N_{def} \gg 1$ . Comparing (9) with (4) and (5), we see that the upstream user service time of downloading CMs is significantly reduced.

2) For uploading CMs (i.e., data packet),

$$T_{usv} = (N_{def} N_c + N_{uCM} N_{u\_data} + N_{def} N_{dCM} N_{u\_ack}) t_{ms}. \quad (10)$$

Comparing (10) with (4) and (5), we see that the LPD mechanism may slightly increase the upstream user service time of the uploading CM.

The asymmetry ratio  $\eta$  of downstream TCP transfers for LPD can be derived as

$$\eta = \frac{N_{def} N_{dCM} L_{data}}{N_{def} N_{dCM} L_{data} + \frac{N_{uCM} L_{data}}{d}} \times \frac{C_d}{d \times L_{data}} \times \frac{\left[ N_c + \frac{N_{uCM}}{N_{def}} \times N_{u\_data} + N_{dCM} N_{u\_ack} \right] t_{ms}}{N_{dCM}}. \quad (11)$$

Compared with the original scheduling (i.e., simple FCFS), LPD has smaller  $\eta$  (i.e., compare  $\eta$  in (1) with (11)) thanks to smaller upstream user service time. A smaller  $\eta$  will speed up the dropping of the value of  $\eta$  to one, when the system will behave normally, i.e., in the symmetric network. This will in turn shorten the round trip delay, and increase the downstream throughput.

b) *Round trip delay:* With two-way transfers, the system is mostly operated as in asymmetric networks. Equation (12) shows the average round trip delay of sending a packet when  $\eta > 1$ .

$$RTT = 2T + T_{trans} + B_{CM} \times T_{usv} \quad (12)$$

where  $B_{CM}$  is the buffer size of the CM, and

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

Substituting (9) and (10) into (12), we can derive the average round trip delays of downstream TCP transfers and upstream TCP transfers accordingly.

The last term of (12) dominates RTT. Hence, LPD has far shorter round trip delay for the downstream TCP transfer as compared to the original scheduling (i.e., simple FCFS) but slightly longer round trip delay for the upstream TCP transfer.

3) *Number of Deferment Steps*: We now determine the number of steps a packet should be deferred. Let  $L_x$  be the data grant size requested by a new request  $R_x$ , and  $D_x$ , the number of steps request  $R_x$  should be deferred before being granted.  $D_x$  is determined based on the following two considerations.

a) *Balance the throughput between the downstream and upstream transfers*: In typical HFC networks, the downstream channel capacity is far higher than the upstream one. In DOCSIS, since “one-CM-one-data grant” is implemented, the downloading CM has data throughput very close to the uploading one. As a result, the downstream capacity may be underutilized if the numbers of CMs in the two directions are equal. To remedy this problem, we attempt to push the data throughput of a downloading transfer to its limit but not degrading the uploading one. The first step is to distinguish from whom the request is sent. If the request is from a downloading CM, it must be for an ACK packet. On the other hand, if it is from an uploading CM, it must be for a data packet. However, due to packet encapsulation in the layer architecture (or with encryption for security reasons), the CMTS (i.e., a layer-2 component) may not be able to access the payload type of a TCP segment. As a result, it may not be able to tell if a request is for a data packet or for an ACK packet. What the CMTS knows is the number of minislots asked for allocation. Since an ACK packet is small and fixed,  $N_{u\_ack}$  can be used as a basic unit to make further decisions.

The upper bound of  $D_x$  should be  $\lfloor N_{u\_data}/N_{u\_ack} \rfloor$ , the data grant ratio of the data packet to an ACK packet and approximately the packet ratio of  $L_{data}/L_{ack}$ . Should  $D_x$  be set to a larger value, the mechanism would penalize long packets, i.e., the CMs sending long packets would have lower data rates. If we set  $D_x = \lfloor N_{u\_data}/N_{u\_ack} \rfloor$ , the uploading TCP transfers would have the same throughput as the downloading ones. Actually, if we set  $D_x = \lfloor L_x/N_{u\_ack} \rfloor$ , the system would behave like an IEEE 802.14 mechanism, which uses fixed ATM cells as the data unit. In this case, the upstream user service times of downloading and uploading CMs can be given as follows.

a) For downloading CMs:

$$T_{usv\_down} = [N_c + (N_{uCM} + N_{dCM})N_{u\_ack}]t_{ms}. \quad (13)$$

b) For uploading CMs:

$$T_{usv\_up} = [N_c + (N_{uCM} + N_{dCM})N_{u\_ack}] \frac{N_{u\_data}}{N_{u\_ack}} \times t_{ms}. \quad (14)$$

Comparing (13) with (14),  $T_{usv\_up}$  is much larger than  $T_{usv\_down}$  because  $N_{u\_data}/N_{u\_ack}$  is usually about 10 to 20. Note that the upstream user service times in both (13) and (14) depend on the total number of simultaneous transfers, not the number of uploading CMs as in (9) and (10). This implies that using ATM-like transport, the number of uploading CMs does not affect the performance of downloading CMs.

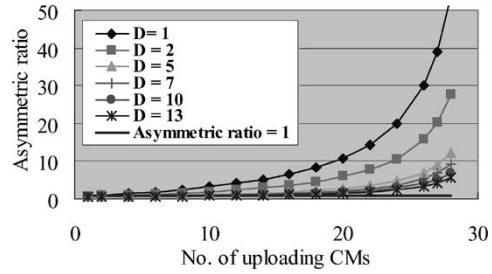


Fig. 4. Asymmetry ratio for two-way transfers.

b) *Balance the round trip delay between downstream and upstream transfers*: The congestion window of the TCP sender grows linearly at a rate of one per round trip time in the congestion avoidance phase. With the “one-CM-one-datagrants” principle in DOCSIS, a downstream TCP transfer fills up the downstream channel at the same rate as an upstream one. Due to the asymmetric upstream and downstream channel capacities, the bandwidth utilization on the downstream channel is very low. Recall that the congestion window of the TCP sender grows at a rate inversely proportional to its round trip delay. If we attempt to make the ratio of the aggregate window growth rate equal to the channel capacity ratio by tuning the value of  $D_x$ , the optimal performance will be obtained. Since the last term of (12) dominates RTT, the relationship between the window growth ratio and the capacity ratio may be approximated as follows.

$$\begin{aligned} & \frac{\frac{1}{RTT_d} N_{dCM}}{\frac{1}{RTT_u} N_{uCM}} \\ &= \frac{C_d}{C_u} \\ & \Rightarrow \frac{\frac{1}{[N_c + (\frac{N_{uCM}}{N_{def}}) N_{u\_data} + N_{dCM} N_{u\_ack}]} t_{ms} B_{CM}}{(\frac{1}{N_{def} N_c + N_{uCM} N_{u\_data} + N_{def} N_{dCM} N_{u\_ack}}) t_{ms} B_{CM}} N_{uCM} \\ &= \frac{C_d}{C_u}. \end{aligned} \quad (15)$$

By simplifying (15), we obtain  $D_x = N_{def} = (N_{uCM}/N_{dCM})(C_d/C_u)$ . Note that  $D_x$  is equal to the ratio of channel capacity times the ratio of the number of uploading to downloading CMs. It has nothing to do with the packet length.

Based on the discussion above, we can classify  $D_x$  into three cases, as follows.

- If  $N_{uCM} = N_{dCM}$ ,  $D_x = \lfloor C_d/C_u \rfloor$ , i.e., the capacity ratio of downstream to upstream channels. Thus, the optimal  $D_x$  depends only on the capacity ratio.
- If  $N_{uCM} < N_{dCM}$ ,  $D_x < \lfloor C_d/C_u \rfloor$ . This is reasonable because when there are less CMs uploading, there would be less TCP packets on the upstream channel to throttle the downstream traffic. Thus, we can set  $D_x$  lower to avoid the overhead caused by too short MAP durations but still maintain good downstream throughput.
- If  $N_{uCM} > N_{dCM}$ ,  $D_x > \lfloor C_d/C_u \rfloor$ . This is also reasonable because when there are more CMs uploading, there would be many TCP packets to fill up the upstream channel and throttle the downstream traffic. Thus,

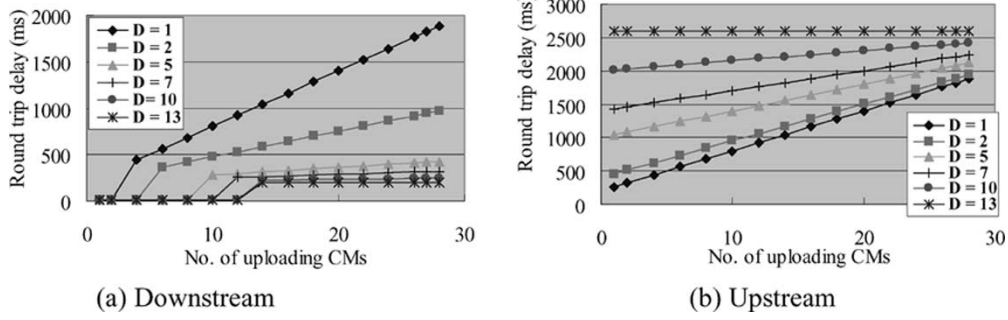


Fig. 5. Round trip delay for two-way transfers. (a) Downstream and (b) Upstream.

we should set  $D_x$  higher to improve the downstream throughput.

In reality, it is very hard to dynamically adjust the number of deferment steps based on the ratio of downloading to uploading CMs because this value often fluctuates greatly. The CMTS cannot estimate the ratio accurately and effectively from the information received from CMs. Thus, we could just use  $\lfloor r(C_d/C_u) \rfloor$  as the number of deferment steps, where  $0 < r < 1$ .

From the previous discussion,  $D_x$  is upper bounded by  $\lfloor r(L_x/N_{u\_ack}) \rfloor$  and  $\lfloor r(C_d/C_u) \rfloor$ , where  $0 < r < 1$ . Meanwhile, a TCP data packet is not of fixed size. Whenever a request arrives at the CMTS, a deferment step is assigned according to the number of minislots the request asks for. In LPD, we design an efficient way to determine deferment steps for requests by using multiple thresholds, say,  $\xi$  thresholds. The number of deferment groups (i.e., queues) is set to  $w = \lfloor r(C_d/C_u) \rfloor = \xi - 1$ , where  $0 < r < 1$ . Each queue is used to store requests in a certain range of data grant size. The data grant size of an ACK packet (i.e.,  $N_{u\_ack}$ ) is chosen as the basic unit. The deferment step  $D_x$  for request  $R_x$  is determined as follows.

- 1) If  $0 < L_x < (2/r)N_{u\_ack}$ ,  $D_x = 1$  and  $R_x$  is put to the queue of group one.
- 2) If  $(k/r)N_{u\_ack} \leq L_x < ((k+1)/r)N_{u\_ack}$ ,  $k = 2, 3, \dots, \lfloor r(C_d/C_u) \rfloor - 1$ ,  $D_x = k$  and  $R_x$  is put to the queue of group  $k$ .
- 3) If  $L_x \geq \lfloor r(C_d/C_u) \rfloor N_{u\_ack}$ ,  $D_x = \lfloor r(C_d/C_u) \rfloor$  and  $R_x$  is put to the queue of group  $w$ .

Usually,  $\lfloor r(C_d/C_u) \rfloor$  is not very large. Thus, this mechanism should work very efficiently.

The LPD mechanism is generalized as follows. The CMTS maintains  $n$  deferment queues to delay requests with different data grant sizes. Each time a request arrives at the CMTS, it will be processed as follows. According to the data grant size requested, the CMTS assigns a deferment step to the request and puts it into the corresponding queue. When it comes time for the next MAP, the CMTS allocates data grants to the requests with enough deferment (i.e., the deferment steps have been decremented to zero). This process starts from the shortest job queue to the longest job queue until the MAP limitation is reached. This gives shorter packets higher priority to send, allowing the data rates of downloading and uploading CMs to be adjusted accordingly.

4) *Numerical Example and Discussion:* Using the same network configuration as in Section IV-A, we obtain the channel capacity ratio of  $\lfloor C_d/C_u \rfloor$  as 10, and the ratio of minislots used for long to short packets,  $\lfloor N_{u\_data}/N_{u\_ack} \rfloor$ , 13.

We observe the asymmetry ratio  $\eta$  and TCP round trip delay of downloading CMs, varying the number of uploading CMs and the number of deferment groups  $w$ . From Figs. 4 and 5(a), we learn that the asymmetry ratio  $\eta$  stays below one longer when LPD is used. (The curve with  $\eta = 1$  is for the original DOCSIS). When  $w$  is set to 10 or 13 (i.e., the upper bounds), the downstream reaches the best performance in terms of both the asymmetry ratio and the average round trip delay. However, from Fig. 5(b), we find that the round trip delays of upstream transfers are rather high when  $w$  is 10 or 13. Now we look at the curve with  $w = 5$ , again, in Figs. 4 and 5. Both have slightly higher asymmetry ratio  $\eta$  and round trip delay for downstream transfers, but have much better round trip delay for upstream transfers, as compared to the two upper bounds. Thus, accounting for all these considerations,  $w = 5$  is the best setting for either direction of transfers, i.e.,  $w = 1/2 \times \lfloor C_d/C_u \rfloor$ .

### C. Adaptive Slot Allocation Mechanism

This section gives a detailed description of the proposed adaptive slot allocation mechanism comprised of the FRT and LPD mechanisms. The mechanism classifies requests into  $w$  deferment groups, where  $w = \lfloor r(C_d/C_u) \rfloor$ . There are  $w - 1$  thresholds:  $\delta_1, \delta_2, \dots, \delta_{w-1}$ , and  $\delta_x = (x + 1/r)N_{u\_ack}$ . The CMTS maintains  $w$  deferment queues to delay requests asking for different data grant sizes. Each time a request  $R_x$  arrives at the CMTS, it is processed as follows. According to the data grant size requested, the CMTS assigns a deferment step  $D_x$  to  $R_x$  and put it into the corresponding queue as described in Section III-B3. When it comes time for the next MAP, the CMTS allocates the data grants to the requests with enough deferment (i.e., the ones with zero or one deferment step) and issues the Data Pending IEs for those with more-than-one deferment step, in the order of the shortest length queue to the longest queue until the MAP limitation is reached. Then, the CMTS applies the FRT mechanism to reserve some unicast minislots for the CMs whose data grants are placed in the backend of the MAP. Note that this only happens (i.e., FRT is applied) when all CMs performs one-way transfers, because LPD enforces long packets to be placed in the backend of each MAP.

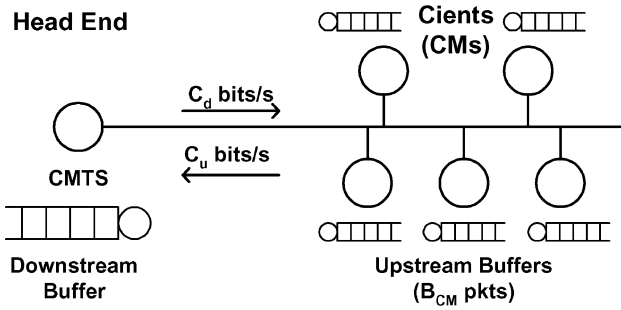


Fig. 6. Network topology used in the simulation.

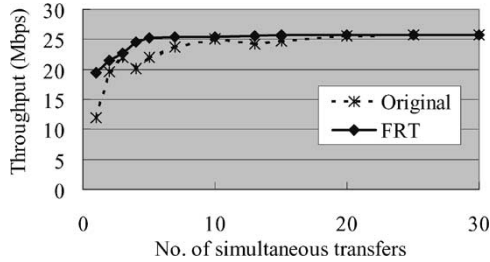


Fig. 7. Aggregate throughput for one-way transfers.

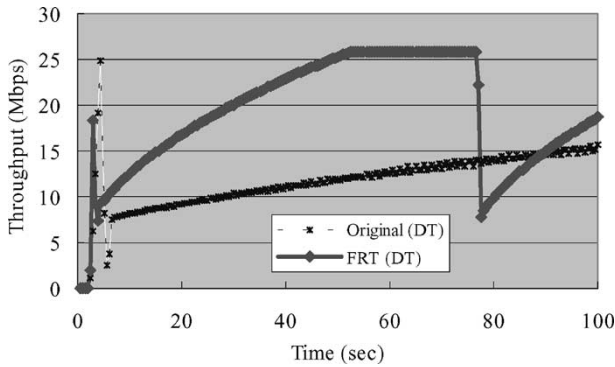


Fig. 8. Downstream throughput.

#### IV. SIMULATION

This section presents the simulation results using ns2 to show the performance improvement of the FRT mechanism over the original DOCSIS control mechanism.

##### A. Simulation Environment

An HFC network has a tree and branch network topology. Each branch is comprised of an upstream channel and a downstream channel. Since each branch works identically and independently, we consider only one branch in the simulation. Fig. 6 shows the network model for our simulation.

The simulation parameters are described as follows. Data rate: downstream channel: 26 970 350 bps, upstream channel: 2 560 000 bps. The propagation delay from the CMTS to all cable modems is a fixed value of 0.5 ms (the maximum tolerable latency in DOCSIS is 0.8 ms). The binary back-off start is 4, the back-off end is 10 and the attempt limit is 16. One upstream minislot is 50  $\mu$ s (8 times 6.25  $\mu$ s). The contention period in one MAP is fixed at 50 minislots. A MAP cannot describe more than 2048 minislots or more than 240 IEs.  $D_{MAP}$  is set to 2 ms, i.e., four times one-way latency of HFC network.

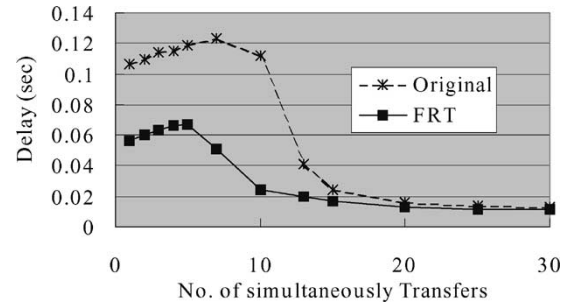


Fig. 9. Upstream channel access delay.

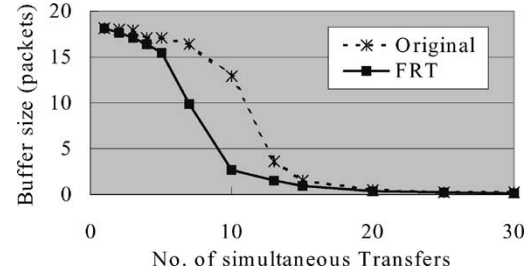


Fig. 10. Average upstream buffer size.

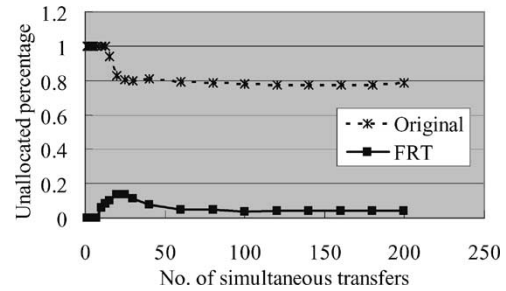


Fig. 11. Percentage of piggybacked requests unallocated in the next MAP.

Each CM in our simulation model has only one SID. The downstream buffer is set to 50 packets. The upstream buffer is set to 20 packets. We use FIFO queuing discipline to manage both CMTS and CM buffers. Each CM can have one TCP transfer only. Two performance metrics are defined as follows.

- 1) TCP aggregate throughput: defined as the total TCP data (bits/sec) received successfully by all the CMs. The traffic of ACK packets is not taken into account.
- 2) Access delay: defined as the time between when a packet is passed to the CM and when the first bit of the packet is transmitted on the upstream channel.

##### B. Simulation Results

1) *One-Way Transfers*: This experiment evaluates the performance of TCP bulk transfers for the original DOCSIS and the FRT mechanism. We assume 200 active cable modems in this network. Each TCP is of 1000 bytes, and each ACK packet is of 40 bytes. We vary the number of simultaneous transfers and observe the aggregate downstream TCP throughput, access delay, and buffer size for both approaches.

Fig. 7 shows the aggregate downstream throughputs for the original DOCSIS control and the FRT mechanism. The aggregate throughput varies from 12 Mbps to 26 Mbps for the original DOCSIS and from 19 Mbps to 26 Mbps for our



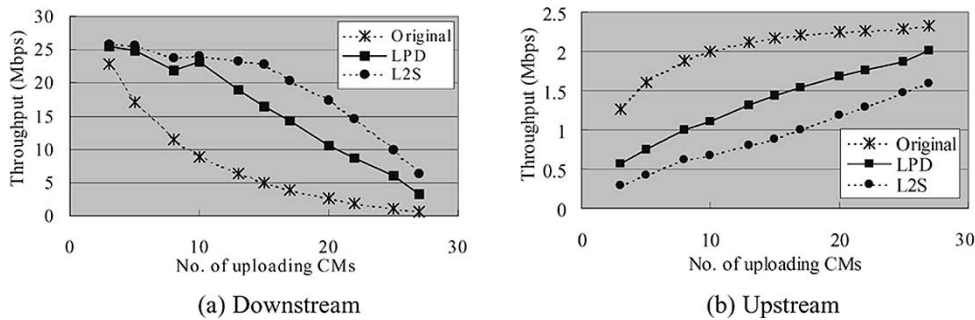


Fig. 12. Aggregate throughput for two-way transfers. (a) Downstream and (b) upstream.

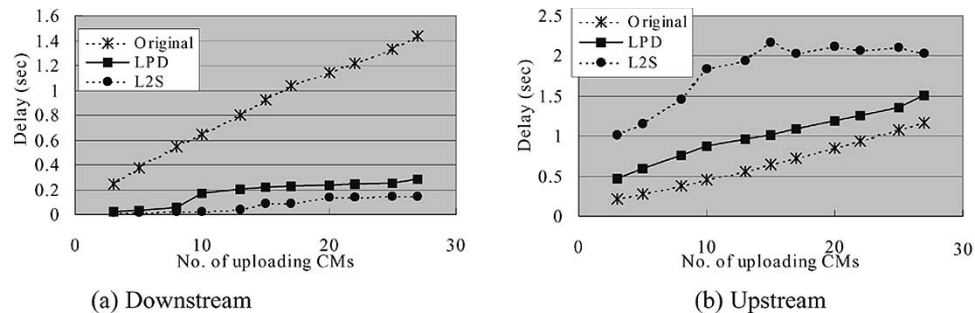


Fig. 13. Access delay. (a) Downstream and (b) upstream.

mechanism. Comparing Fig. 2 in Section III and Fig. 7 here, we learn that the FRT mechanism reduces the asymmetry ratio and upstream user service time to improve aggregate downstream throughput. Fig. 7 also shows that our mechanism not only has better throughput but is also more stable than the original DOCSIS control mechanism. Fig. 8 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 user service time to provide better downstream throughput.

Fig. 9 shows the average access delays on the upstream channel experienced by the original DOCSIS and our mechanism. Fig. 10 shows their respective average upstream buffer sizes. When the number of simultaneous transfers is less than five, both mechanisms suffer from serious bandwidth asymmetry for TCP transfers. At this time, their access delays become very large due to a full buffer at the CM. The access delays increase slightly in this region as the number of simultaneous transfers increases, due to an increase in the average size of a frame. Fig. 10 shows that both approaches have very close average sizes in this region. But compared to their access delays shown in Fig. 9, FRT obviously has an access delay approximately 50% smaller, thanks to FRT increasing the frequency of data slot allocation to CMs. Thus, FRT has better downstream throughput (see Figs. 7 and 8).

When the number of simultaneous transfers varies from five to 15, the access delays vary from 0.12 s to 0.02 s for the original DOCSIS, while the access delays vary from 0.06 s to 0.02 s for our mechanism. Since FRT makes the asymmetry ratio go below one more quickly, it makes access delays drop to one much faster. Both approaches behave similarly when there are more than 15 simultaneous transfers in one branch. In this case, both have asymmetry ratios less than or equal to one.

Fig. 11 shows the percentage of piggybacked requests not allocated in the next MAPs. We see that very small percentage of piggybacked requests remains unallocated in the next MAPs for FRT, while it is high in the original DOCSIS mechanism.

2) *Two-Way Transfers*: This section presents the simulation results using ns2 for the adaptive slot allocation mechanism (denoted as LPD in the figures), the 802.14-like mechanism (denoted as L2S in the figures), and the original DOCSIS control mechanism.

Fig. 12 shows the aggregate throughputs of downstream [Fig. 12(a)] and upstream transfers [Fig. 12(b)], varying the number of uploading CMs from 1 to 30. The dotted lines are for the original DOCSIS; the solid lines with squares and circles are for LPD and L2S, respectively. Fig. 12(a) shows that both LPD and L2S have about 80% improvement of aggregate downstream bandwidth over the original DOCSIS when less than one-third of CMs are transferring upstream. They have slight performance degradation in the upstream direction as shown in Fig. 12(b) (note that Fig. 12(a) and (b) have different scales in throughput). LPD has better improvement than L2S. Thus, the number of deferment groups does not need to be very large. Overall, LPD has the best balance between downstream and upstream bandwidth utilization when the proportion of uploading CMs is between 1/3 and 5/6. Since typically CMs perform downloads in HFC networks, it is rare for more than 5/6 of the CMs to be uploading. The region of interest is when the proportion of uploading CMs is less than 5/6, when performance improvement is observed.

Fig. 13 shows the access delays of downloading CMs [Fig. 13(a)] and uploading CMs [Fig. 13(b)], for the three mechanisms as a function of simultaneous uploading CMs. It shows that of the three mechanisms, simple FCFS (i.e., the original DOCSIS) has the largest access delay for downloading

CMs, and the 802.14-like mechanism (i.e., L2S) has the largest access delay for uploading CMs. The LPD mechanism renders the access delays of downloading CMs much shorter, without significantly increasing the access delays of uploading CMs.

## V. CONCLUSION

In this paper, we have proposed an adaptive slot allocation mechanism comprised of FRT and LPD to improve the performance of TCP over DOCSIS-based HFC networks. FRT is designed to handle one-way transfers and LPD targets two-ways transfers. We have also provided a mean analysis of performance and conducted simulations using network simulator *ns-2* to compare the original control mechanism of DOCSIS with our mechanism. The results show that the proposed mechanism has better performance in terms of aggregate downstream throughput, access delay, and buffer size.

This paper considers the impact of DOCSIS's MAC control mechanism on the performance of TCP only. In the future, we will further investigate the impact of DOCSIS's MAC layer operations on QoS scheduling mechanisms for multimedia traffic on HFC networks. We will then investigate how to integrate the proposed adaptive slot allocation scheme and QoS scheduling mechanisms to render HFC networks an efficient media delivery network.

## ACKNOWLEDGMENT

The authors would like to thank the three anonymous reviewers, whose comments have significantly improved the quality of this paper.

## REFERENCES

- [1] "Data-Over-Cable Service Interface Specifications – Radio Frequency Interface Specification, MCNS Consortium," CableLabs, SP-RFiv1.1–106-001 215, 2000.
- [2] *IEEE Project 802.14/a Draft 3 Revision 1*, 1998.
- [3] M. Droubi, N. Idirene, and C. Chen, "Dynamic bandwidth allocation for the HFC DOCSIS MAC protocol," in *Proc. IEEE IC3N*, Las Vegas, NV, 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 '98*, San Francisco, CA, 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. Broadcasting*, vol. 44, no. 4, pp. 627–635, Dec. 1998.
- [6] D. Sala and J. O. Limb, "Comparison of contention resolution algorithms for a cable modem MAC protocol," in *Proc. Broadband Communications*, Zürich, Switzerland, 1998, pp. 83–90.
- [7] V. Sdralia, C. Smythe, P. Tzerefos, and S. Cvetkovic, "Performance characterization of the MCNS docsis 1.0 CATV protocol with prioritised first come first served scheduling," *IEEE Trans. Broadcasting*, vol. 45, pp. 161–205, June 1999.
- [8] T. Orfanoudakis, N. Leligou, E. Meciu, and A. Harsanyi, "Evaluation of IP oriented HFC access protocols," in *Proc. Broadband Communications*, Zürich, 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. Select. Areas Commun.*, vol. 18, pp. 1261–1270, July 2000.
- [10] H. Balakrishnan, V. N. Padmanabhan, and R. H. Katz, "The effects of asymmetry on TCP performance," *Mobile Networks Applicat.*, vol. 4, pp. 219–241, 1999.
- [11] Y.-D. Lia and W. Liao, "Improving TCP performance over asymmetric networks," in *Proc. IEEE ICC 2001*, Helsinki, Finland, June 2001, pp. 1824–1828.
- [12] S. Varma, "Performance and buffering requirements of TCP applications in asymmetric networks," in *Proc. IEEE INFOCOM '99*, New York, 1999, pp. 1548–1555.
- [13] T. V. Lakshman, U. Madhow, and B. Suter, "Window-based error recovery and flow control with a slow acknowledgment channel: A study of TCP/IP performance," in *Proc. IEEE INFOCOM '97*, Kobe, Japan, 1997, pp. 38–47.
- [14] R. Cohen and S. Ramanathan, "TCP for high performance in hybrid fiber coaxial broad-band access networks," *IEEE/ACM Trans. Networking*, vol. 6, pp. 15–29, Feb. 1998.
- [15] O. Elloumi, N. Golmie, H. Afifi, and D. Su, "A study of TCP dynamics over HFC networks," in *Proc. IEEE GLOBECOM '98*, Sydney, Australia, pp. 545–550.
- [16] R. Cohen and S. Ramanathan, "Using proxies to enhance TCP performance over hybrid fiber coaxial networks," *Comput. Commun.*, vol. 20, 1998.
- [17] I. T. Ming Chit *et al.*, "Performance evaluation of web browsing over hybrid fiber coaxial broad-band networks," *Proc. ICON 1999*, pp. 372–382, 1999.
- [18] UCB/LBNL/ISI/VINT Network Simulator – ns, ver. 2 [Online]. Available: <http://www.isi.edu/nsnam/ns/>
- [19] H. J. Ju and W. Liao, "Fast request transmission (FRT) for DOCSIS-based CATV networks," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME) 2002*, Lausanne, Switzerland, Aug. 2002, pp. 543–547.



**Wanjiun Liao** (S'96–M'97) received the B.S. and M.S. degrees from National Chiao-Tung University, Hsinchu, Taiwan, in 1990 and 1992, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California (USC), Los Angeles, in 1997.

She joined the Department of Electrical Engineering, National Taiwan University (NTU), Taipei, as an Assistant Professor in 1997. Since August 2000, she has been an Associate Professor. Her research interests include wireless networks,

broadband access networks, and Internet protocols.

Dr. Liao is actively involved in the international research community. She is currently an Associate Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. She has received many research awards, including the Outstanding Research Paper Award in Electrical Engineering at USC in 1997. Two papers she co-authored with her students received the Best Student Paper Award at the First IEEE International Conferences on Multimedia and Expo (ICME) in 2000, and the Best Paper Award at the First International Conference on Communication, Circuits and Systems (New York: IEEE Press) in 2002. She was elected as one of Ten Distinguished Young Women in Taiwan in 2000 and is listed in the *Marquis Who's Who* in 2003 and the *Contemporary Who's Who* in 2003.

**Huei-Jiun Ju** received the B.S. and M.S. degrees from the Department of Electrical Engineering, National Taiwan University, Taipei, in 1999 and 2001, respectively. She is currently pursuing the Ph.D. degree in the Department of Electrical Engineering, University of California, Los Angeles.