The Contention Behavior of DOCSIS in CATV Networks

Kai-Chien Chang and Wanjiun Liao

***Abstract*—In this paper, we study the contention behavior of DOCSIS in cable TV networks. Specifically, we focus on the behavior of TCP over DOCSIS. We determine the expected access delay for TCP transmissions in CATV networks. The access delay here is defined as the interval between the time when a data packet arrives at a Cable Modem (CM) and the time when that packet is successfully sent by the CM. The analytical model is comprised of two parts. The first part is to calculate the probability that a CM sends a request in a randomly selected minislot, and the second part is to derive the expected access delay based on the probability derived in the first part. The accuracy of the analytical model is validated by simulations. The results show that our analytical model can accurately model the contention behavior of DOCSIS in CATV networks.**

***Index Terms*—CATV, contention access, DOCSIS, TCP.**

I. INTRODUCTION

THE Hybrid Fiber Coax (HFC) cable TV network is an alternative of broadband access networks to the home. The typical architecture of HFC networks is a tree-and-branch network, in which the downstream channel is a broadcast channel and the upstream one is a random access channel, as shown in Fig. 1. Here the downstream channel refers to the channel from the Cable Modem Termination System (CMTS) at the headend to Cable Modems (CMs) at the home. Thus, the CMTS is the only transmitter and all the CMs are the receivers in the downstream direction. The upstream channel is from CMs to the CMTS; thus all the CMs are the transmitters and the CMTS is the only receiver in the upstream direction. The media access control (MAC) mechanism in HFC may follow the Data-Over-Cable Service Interface Specifications (DOCSIS) of the Multimedia Cable Network System Partners (MCNS) [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, an upstream channel is modeled as frames of mini-slots. Each frame comprises contention minislots and data minislots. The detail of each frame is specified via a control message, called MAP, periodically transmitted on the downstream channel from the CMTS. In a MAP, a variable number of information elements (IEs) are specified, each of which defines a range of mini-slots to be used. The number of IEs, and the corresponding mini-slots, 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; a Data Grant IE describes the transmission interval for a CM to transmit packets, and the Null IE terminates the assignment list. Each MAP specifies the detail of the next frame, and thus it must be received by all CMs before the next frame starts.

Each time a CM has a packet to transmit, it requests for a Data Grant IE in a later frame. The request may be sent via piggyback or via contention. If it is a backlogged CM (i.e., a CM with a non-empty packet queue) and a Data Grant IE has been assigned to it for the next frame, the request is piggybacked on an outgoing packet using the assigned Data Grant IE; otherwise, the request is sent via contention, i.e., the CM contends for the use of the upstream channel via contention minislots. Each contention request occupies one mini-slot. If several CMs compete for one mini-slot in the contention interval, all the requests in the minislot are corrupted due to collision. After sending a request, each 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 transmits a data packet using the Data Grant IE if assigned, or keeps waiting for a Data Grant IE and sends 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 back-off window (i.e., contention window) by a factor of two, as long as the window size is less than the maximum back-off window. The CM randomly selects a number within its new back-off window, pauses until the timer expires, and then repeats the contention. This retry process continues until the maximum number of retries has been reached, at which time the packet is discarded.

The design issues of HFC networks have been widely studied for years. An efficient MAC layer scheduling and allocation algorithm is proposed in [3]–[5] for the upstream channel of HFC networks. The performance of HFC networks is evaluated in [6]–[10]. The flow control mechanism of TCP is tuned in [11]–[14] to improve the performance of TCP over asymmetric networks. The performance of TCP in HFC networks is investigated in [15]–[19]. The impact of DOCSIS on the performance of TCP is studied in [20]–[22].

In this paper, we provide an in-depth analytical study on the "contention" behavior of DOCSIS in HFC networks. Specifically, we determine the expected access delay for TCP transmissions over DOCSIS-based HFC networks. The access delay is defined as the interval between the time when a data packet

---

Manuscript received August 17, 2006; revised October 17, 2006. This work was supported in part by theNational Science Council (NSC), Taiwan, under a Center Excellence Grant NSC95-2752-E-002-006-PAE, and in part by theNSC under Grant Number NSC95-2221-E-002-066.

K.-C. Chang is with the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan.

W. Liao is with the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, and also with the Graduate Institute of Communications Engineering and the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 106, Taiwan (e-mail: wjliao@ntu.edu.tw).

Digital Object Identifier 10.1109/TBC.2006.887169

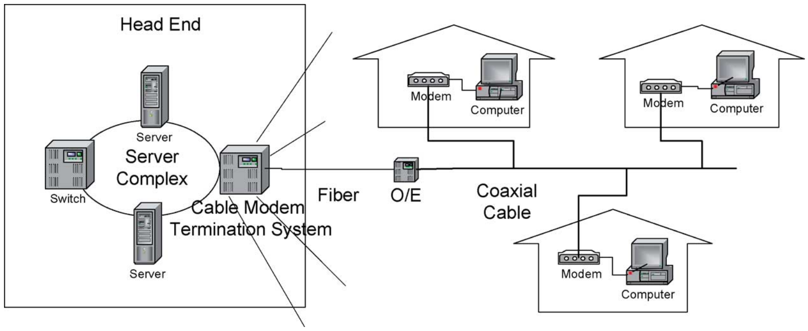


Fig. 1. The tree-and-branch topology of the HFC network.

arrives at a CM and the time when the packet is successfully sent by the CM. Since we have studied "piggyback" in DOCSIS in [19]–[22], this paper will focus only on the access delay of packets whose requests are sent by contention.

The rest of the paper is organized as follows. In Section II, the system model is described and the problem is stated. In Section III, the probability of sending a request by a CM in a randomly selected minislot is derived. In Section IV, the mean access delay for sending a packet is obtained. In Section V, the simulation results are presented. Finally, the conclusion is drawn in Section VI.

II. SYSTEM MODEL AND PROBLEM SPECIFICATION

The MAC operation of DOCSIS consists of two parts: requests by contention and requests by piggyback. The expected access delay for sending a piggybacked request has been studied in [19]–[22]. The focus of this paper will be placed only on the contention part.

A. *The Contention Operation of DOCSIS: An Overview*

The contention resolution method used in DOCSIS is truncated binary exponential backoff [1]. The CMTS assigns the size of the initial backoff window (W_0) and the maximum backoff window (W_m) in the MAP. When a CM has a request to transmit via contention, it randomly chooses a value in its window. For example, if $W_0 = 16$, the CM will choose a backoff value randomly between 0 and 15. The backoff value indicates the number of contention minislots the CM has to defer before it can transmit the request. If a collision incurs, which will be detected through no data grant or data pending in the subsequent MAP, the CM will double its window size, until the maximum window size is reached. If a request has tried 16 times and still fails, the request is discarded.

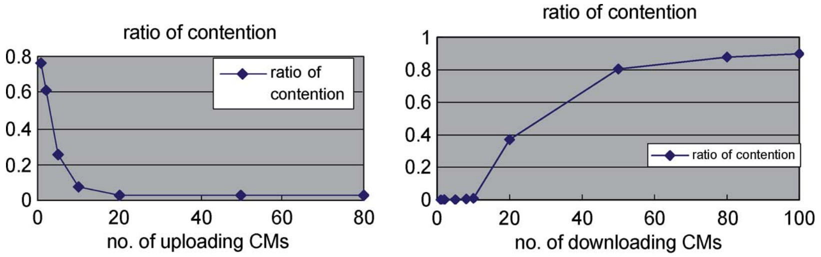
B. *Notations*

τ	the probability that a CM sends its request in a randomly chosen contention minislot
p_c	the probability that a collision seen by a request transmitted on the channel
p_s	the probability of a successful transmission in a contention minislot
N_{CM}	the number of CMs downloading files
N_{frame}	the number of minislots in a frame
Nc	the number of contention minislots described by a MAP
T_{ad}	Access delay
T_{req_ad}	Request access delay
T_{req_sch}	Request scheduling delay
W_i	Contention window size in backoff stage i
W_0	Initial contention window
W_m	Maximum contention window
m	Maximum transmission attempt
$Boff_i$	Backoff value in terms of number of minislots selected by a CM whose request is in backoff stage $i-1$
L_{data}	the size of a data packet (in bits),
L_{ack}	the size of an ACK packet (in bits),
C_d	Downstream channel bandwidth (in bps)
C_u	Upstream channel bandwidth (in bps)
t_{ms}	one mini-slot time on the upstream channel (in second)

C. *Assumptions and Some Definitions*

In this paper, we make the following assumptions:

1) When the number of simultaneous transfers is fixed, the probability that a CM transmits a request in a randomly selected contention minislot is constant and is independent of the probabilities of other CMs.
2) When the number of simultaneous transfers is fixed, the probability that a request suffers from a collision is constant and is independent of the backoff window size and the number of retransmissions attempted for sending the request.
3) The number of minislots in a frame, i.e., N_{frame}, is constant.
4) The number of contention minislots in a DOCSIS frame, i.e., Nc, is also constant.
5) There is no error on the channel. That is, if the CMTS does not receive a request correctly, it must be caused by a collision.
6) The initial window size is assumed smaller than the contention interval, i.e., $W_0 < Nc$.
7) For simplicity, we assume that the maximum backoff window size equals the maximum number of retries.



(a) (b)

Fig. 2. The portion of requests sent via contention. (a) Two-way TCP transfers. (b) One-way TCP transfers.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Cd	26.97Mbps
Cu	2.56Mbps
t_{ms}	50 us
Nc	50 minislots
Ldata	1024 bytes
Lack	64 bytes
W_0	16
m	16
d	2

In this paper, the following terms are defined:

1) The backoff stage is defined as the number of collisions a request has experienced.
2) The backoff counter is defined as the number of contention minislots a request has to defer before it can be transmitted.

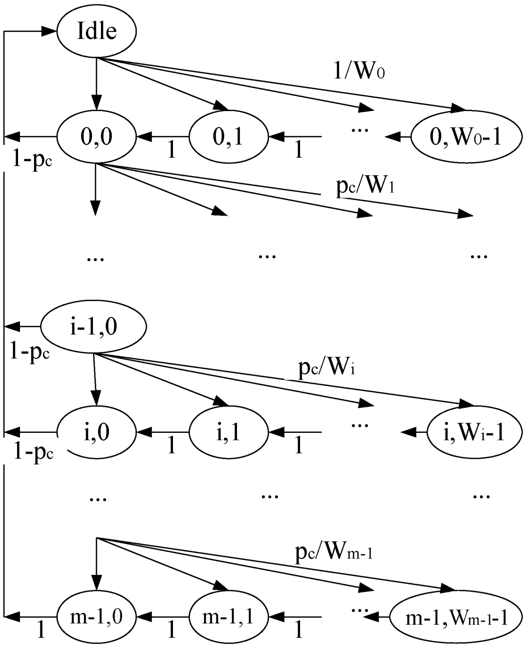


Fig. 3. The Markov model considering the backoff mechanism of DOCSIS.

D. Problem Statement

This paper studies the contention behavior of DOCSIS in HFC networks. When two-way transfers (i.e., some active CMs perform downloading and some perform uploading) are allowed in DOCSIS, the requests for the use of the upstream channel tend to be sent via piggyback. However, when only one-way transfers are performed, i.e., all the active CMs are downloading, the requests mostly go via contention. This phenomenon can be observed in Fig. 2, which shows the ratio of requests sent via contention. The statistics is collected by simulations, using ns-2 with the parameters listed in Table I.

In Fig. 2(a), the total number of active CMs is 100, and some are downloading and some are uploading. The portion of requests sent by contention is quite small when the number of CMs is more than 20, i.e., most requests are sent via piggyback. In Fig. 2(b), the result is collected using the same simulation environment as Fig. 2(a) except that all the active CMs are downloading. We can see that when the number of downloading CMs with one-way transfers is more than 20, the portion of contention requests becomes significant. Thus, in this paper we consider only one-way TCP transfers and a large number of downloading CMs in DOCSIS networks (otherwise most requests may go by piggyback, which has been studied in [19]–[22]).

The access delay is an important performance metric for broadband access networks. Thus the focus of this paper is placed on the access delay of packets whose requests are sent by contention. To estimate the expected value of access delay, a two-part procedure is taken. In the first part, the probability that a CM transmits a request in a randomly selected slot is estimated. In the second part, the expected value of access delay that a packet experiences is calculated using the probability derived in the first part. Based on the request access delay, the average round trip time (RTT) for TCP transmission can also be estimated.

III. THE PROBABILITY OF A CM SENDING A REQUEST IN A RANDOMLY SELECTED MINISLOT

In this section, a model is developed in steps to estimate the probability of a CM transmitting a request in a randomly selected minislot, i.e., τ.

A. Base Model

We first develop a discrete-time Markov chain model considering the truncated binary exponential backoff mechanism in DOCSIS. As shown in Fig. 3, a state transition only takes place at the end of each contention minislot. The major assumption in this model is that each CM always contends for the use of the upstream channel, via a new request or a retry. In other words, each time a request is successfully transmitted, a new request is generated before the next frame starts. Note that the upstream data queues at each CM are not backlogged (otherwise, requests are sent via piggyback); we can consider that there is a request queue at each CM, and the request queues are backlogged.

The state $\{\mathrm{s}(t), b(t)\}$ in this model is defined as follows. $s(t)$ is the backoff stage of a request packet at time t, defined as the number of collisions the request packet has suffered; $b(t)$ is the backoff counter at time t. When the backoff counter becomes zero, the request is sent. The transition probabilities of this model are derived as follows. Let $P\{i_1, k_1|i_0, k_0) = P\{s(t+1) = i_1, b(t+1) = k_1|s(t) = i_0, b(t) = k_0\}$.

$$\begin{cases} P\{0,k|IDLE\} = 1/W_0, & k \in (0, W_0 - 1) \\ P\{IDLE|i,0\} = 1 - p_c, & i \in (0, m-2) \\ P\{i,k|i-1,0) = p_c/W_i, & i \in (1, m-1) k \in (0, W_i - 1) \\ P\{i,k|i,k+1) = 1, & i \in (0, m-1) k \in (0, W_i - 2) \\ P\{IDLE|m,0\} = 1 \end{cases} \tag{1}$$

The first equation in (1) is obtained with the assumption that after a request has been successfully transmitted, a CM will start processing the transmission of another request. Thus, the probability of leaving the IDLE state is one. Since the CM randomly chooses a backoff value in its backoff window, and the probability of choosing initial backoff values is uniformly distributed among the backoff window, the transition probability from the IDLE state to any states of stage 0 is $1/\mathrm{W}_0$. Since the collision probability seen by a transmitting request is p_c, under the assumption of no channel errors, the probability of a successful transmission is $1 - p_c$. This gives the second equation. If a collision occurs when a CM is sending a request, its contention window doubles, and in the model, the backoff stage is incremented; then it selects a backoff value in the window. This results in the third equation. The fourth equation results because the backoff counter decreases by one at every contention minislot. The fifth is due to the fact that once the window achieves its maximum value, if the transmission still fails this time, the request will be aborted. As a result, it returns to the IDLE state, whether or not the attempt is successful.

The stationary distribution of the Markov chain is defined as seen in (2) at the bottom of the page.

$$b_{i,k} = \lim_{t \to \infty} P\{s(t) = i, b(t) = k\}.$$

From the transition probabilities in (1) and the fact the sum of all states is one, we can derive the limiting probability of the IDLE state as seen in (2) at the bottom of the page.

When the backoff counter decreases to zero, a request is sent. Hence, the probability of a CM transmitting a request in a randomly selected contention minislot τ is the summation of the stationary probabilities of all states whose backoff counter is zero. That is, $\tau = \sum_{i=0}^{m} b_{i,0}$. Furthermore, in the steady state, this probability equals the stationary distribution of the IDLE state. This gives the following equation.

To derive the value of p_c, another fact is used. Given the probability of a CM transmitting a request in a randomly selected contention minislot τ, the probability that a collision occurs to a transmitting request is that at least one of the other CMs is transmitting a request within the same minislot. This gives the following equation

$$p_c = 1 - (1 - \tau)^{N_{CM} - 1}. \tag{3}$$

Let (2) be $\tau = f_1(p_c)$, (3) be $p_c = g(\tau)$, and the inverse of (3) be $\tau = g^*(p_c)$. In the range $p_c \in (0, 1)$, $f_1(p_c)$ is continuous and monotonic decreasing and $g^*(p_c)$ is continuous and monotone increasing. Moreover, $f_1(0) > g^*(0)$ and $f_1(1) < g^*(1)$. Thus, we can conclude that there is a unique solution for a (τ, p_c) pair, which can be derived by numerical techniques.

B. DOCSIS Model

The base model focuses on the truncated exponential backoff mechanism of DOCSIS for minislot contention. Now, we turn to the effect of MAPs periodically transmitted in the downstream channel on determining the value of τ. Under the DOCSIS MAC operation, CMs do not know whether requests have collided or not immediately after the requests are transmitted. They have to wait till the next MAP arrives. Thus, the only thing the CM can do is waiting for three more minislots (i.e., the three gray minislots) in the contention interval. To model this delay, some modifications to the base model are made.

Here, we introduce another counter, called the *waiting counter*. This counter counts the number of contention minislots the CM has to wait before it can tell whether or not its request has been sent successfully. We assume that the probability of a request arriving in a CM is the same at any contention minislot. Thus, the CM has to wait an additional n_w minislots for the next MAP to come, where n_w is a random variable uniformly distributed in $[0, Nc - 1]$. The modified Markov chain model can then be developed, as shown in Fig. 4. When the backoff counter decreases to 0, the request is sent. Meanwhile, the waiting counter will be initialized to n_w and start to decrement. When the waiting counter also decreases to 0, the CM receives the MAP, and it can tell whether or not the request is successfully received by the CMTS.

The state of this modified model becomes $\{\mathrm{s}(t), b(t), nw(t)\}$. In fact, the chain is two-dimensional because *i*) the waiting counter will not be initiated until the backoff counter becomes

---

$$b_{IDLE} = \frac{2(1-p_c)(1-2p_c)}{2(1-p_c)(1-2p_c) + W_0(1-p_c)\left(1-(2p_c)^m\right) + (1-2p_c)\left(1-p_c^m\right)}$$

$$\tau = \sum_{i=0}^{m} b_{i,0} = b_{IDLE} = \frac{2(1-p_c)(1-2p_c)}{2(1-p_c)(1-2p_c) + W_0(1-p_c)\left(1-(2p_c)^m\right) + (1-2p_c)\left(1-p_c^m\right)} \tag{2}$$

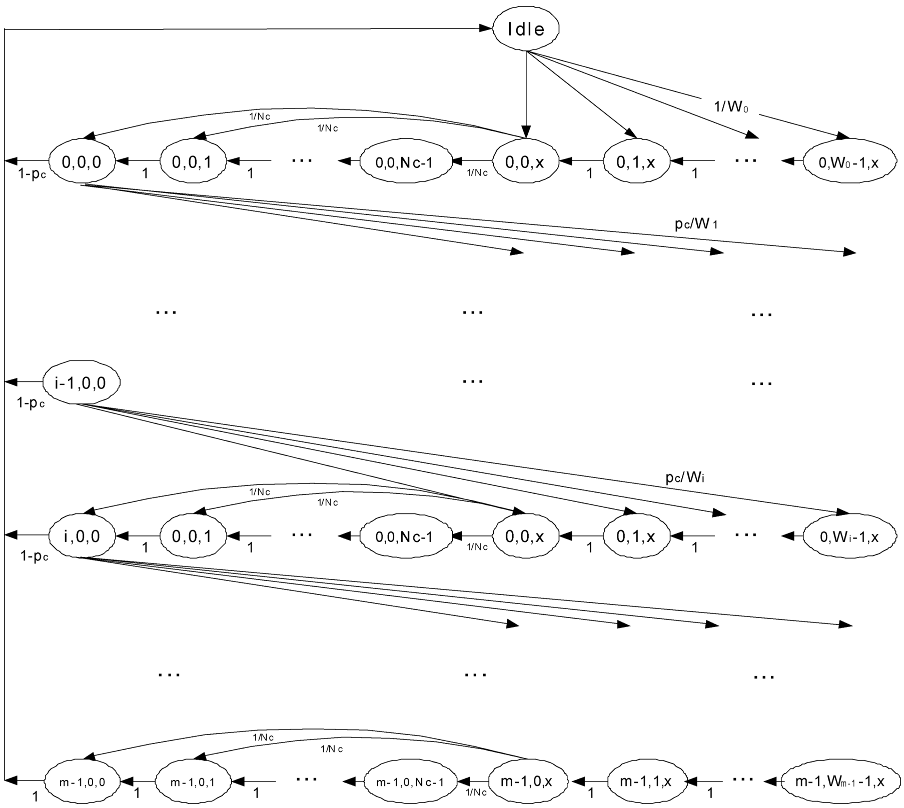


Fig. 4. Modified Markov chain model considering the effect of MAP in DOCSIS.

zero, and *ii*) when the waiting counter functions, the backoff counter must be zero. The transition probabilities of this model are derived as follows.

$$\begin{cases} P\{0,j,x|IDLE\}=1/W_0, & j\in(0,W_0-1)\\ P\{IDLE|i,0,0\}=1-p_c, & i\in(0,m-2)\\ P\{i,j,x|i-1,0,0)=p_c/W_i, & i\in(1,m-1)j\in(0,W_i-1)\\ P\{i,j,x|i,j+1,x)=1, & i\in(0,m-1)j\in(0,W_i-2)\\ P\{IDLE|m,0,0\}=1 & \\ P\{i,0,k|i,0,k+1)=1, & i\in(0,m-1)k\in(0,Nc-2)\\ P\{i,0,k|i,0,x)=1/Nc, & i\in(0,m-1)k\in(0,Nc-2) \end{cases}$$

Basically they are the same as those for the base model, except the last two equations. The sixth one is the decrement of the waiting counter, while the last one determines how many contention minislots the CM has to wait before the next MAP is received. Note that the notation used here is also the same as the base model.

With the transition probabilities, the stationary probability of the IDLE state can be derived as shown in the equations at the bottom of the page.

C. TCP Model

The DOCSIS model is developed without considering the upper layer protocol, namely, TCP. Now we consider TCP in our model. Due to the self-clocking nature of TCP and the one-way transfer we assume, the upstream request queue is unlikely to be backlogged. This argument can be easily verified with the following example. Assume that the capacities of the downstream and upstream channels are set to 26.97 Mbps and 2.56 Mbps, respectively; the length of an ACK packet is 64 bytes and that of a data packet is 1024 bytes. When the number of downloading CMs is more than 20, the system behaves as in a symmetric network, i.e., the asymmetry ratio, defined as the capacity ratio of downstream to upstream channels times the length ratio of ACK to data packets, is less than one (i.e., $(26.97/2.56)\times(64/1024)\cong 0.66$). This implies that the upstream channel is not the bottleneck anymore, and the upstream packet buffer is often empty. We assume the downloading bandwidth is fully utilized, and every CM shares an equal amount of bandwidth. Assume that there are 100 downloading CMs, each minislot is

---

$$b_{IDLE}=\frac{2(1-p_c)(1-2p_c)}{2(1-p_c)(1-2p_c)+W_0(1-p_c)\left(1-(2p_c)^m\right)+(N_c+2)(1-2p_c)\left(1-p_c^m\right)}$$

and again, $\tau=b_{IDLE}$. Thus,

$$\tau=\frac{2(1-p_c)(1-2p_c)}{2(1-p_c)(1-2p_c)+W_0(1-p_c)\left(1-(2p_c)^m\right)+(N_c+2)(1-2p_c)\left(1-p_c^m\right)}\qquad(4)$$

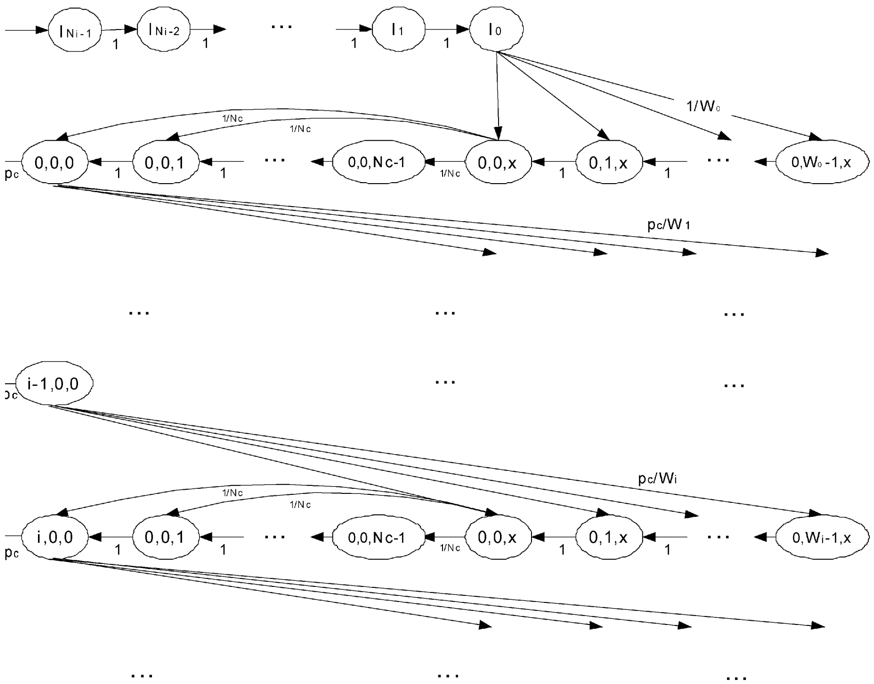


Fig. 5. The Markov model considering the TCP ACK mechanism.

50 μs, and the average length is 146.7 minislots. Thus, the bandwidth each CM shares is equal to 27 $\mathrm{Mbps}/100 = 0.27$ Mbps. The number of packets transmitted on the downstream channel for each CM in the following time interval can be calculated as follows:

1) in a sec: $0.27\ \mathrm{Mbps}/1024\ \mathrm{bytes} = 34.56$ (packets/sec);
2) in a minislot: $34.56\ (\mathrm{packets/sec}) * 50 * 10^{-6} = 1.728 * 10^{-3}$ (packets/minislot);
3) in a frame: $1.728 * 10^{-3}$ (packets/minislot) $*$ 146.7 (minislots/frame) $= 0.25$ (packets/frame).

Thus, on the average, each CM receives one packet to be transmitted every $1/0.25 = 4$ frames.

From the discussion above, there must be some idle stages between when the CM knows if the previous request is successfully transmitted and when the CM starts sending another request. The revised Markov chain model is shown in Fig. 5.

The transition probabilities of this model are derived as follows.

$$\begin{cases}
P\{0,j,x|I_0\}=1/W_0, & j\in(0,W_0-1)\\
P\{I_{Ni-1}|i,0,0\}=1-p_c, & i\in(0,m-2)\\
P\{i,j,x|i-1,0,0)=p_c/W_i, & i\in(1,m-1)j\in(0,W_i-1)\\
P\{i,j,x|i,j+1,x)=1, & i\in(0,m-1)j\in(0,W_i-2)\\
P\{I_{Ni-1}|m,0,0\}=1 & \\
P\{i,0,k|i,0,k+1)=1, & i\in(0,m-1)k\in(0,Nc-2)\\
P\{i,0,k|i,0,x)=1/Nc, & i\in(0,m-1)k\in(0,Nc-2)\\
P\{I_l|I_{l+1}\}=1, & l\in(0,Ni-2)
\end{cases}$$

The transition probabilities of this model are basically the same as those of the DOCSIS model, except for the last equation, which corresponds to the decrement of the idle stages. The problem is how to determine the number of idle stages, N_i. Here we estimate the number of stages to be added based on the data rate of the channel and the number of the CMs downloading files. Due to the difficulty to modeling the actual pattern of ACK generation, we use downstream channel bandwidth and the number of CMs downloading files to estimate the mean ACK arrival rate seen by the MAC layer. Thus, the inter-arrival time of ACK packets is

$$\text{ACK interval (in slots)} = \frac{L_{data}}{C_d/N_{CM}} \times \frac{1}{t_{ms}},$$

where L_{data} is the size of a data packet (in bits), C_d is the downstream channel bandwidth (in bps), N_{CM} is the number of CMs downloading files, and t_{ms} is the time defined as one mini-slot on the upstream channel (in second).

The interval includes contention and data minislots. Since in our model transitions only occur at the end of contention minislots, we translate the ACK interval into contention minislots spanning multiple frames (i.e., only counting contention minislots in each frame) as $N_i = \lfloor(\text{ACK interval}/N_{frame}) \times N_c\rfloor$. We assume that after a request is successfully transmitted, the next request will arrive after N_i transitions.

If the delayed ACK mechanism is used (i.e., sending one ACK packet to acknowledge the receipt of d data packets), the ACK interval will be multiplied by a factor of d. Under the assumption of a fixed frame size, N_{frame} can be estimated as follows.

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

where d is the parameter for the delayed ACK policy, L_{data} is the size of a data packet (in bits), L_{ack} is the size of an ACK packet (in bits), C_u is the upstream bandwidth (in bps), C_d is the downstream bandwidth (in bps), N_c is the number of contention minislots in a frame.

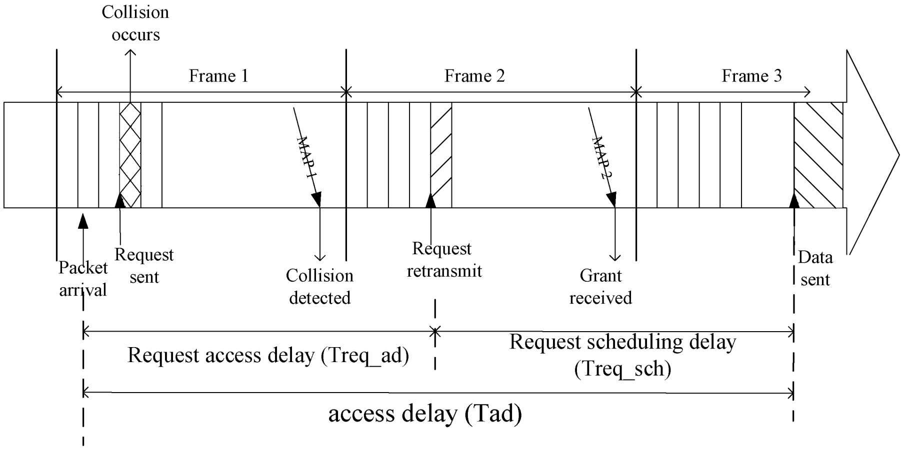


Fig. 6. An example of MAC operation of DOCSIS.

Note that this equation is based on the delayed ACK mechanism. If there is no delayed ACK, the equation for the frame size becomes

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

As a result, given the number of downloading CMs, the idle stages, N_i, can be derived accordingly. The stationary distribution of state I_0 is then expressed as shown in the equation at the bottom of the page.

IV. THE EXPECTED ACCESS DELAY OF SENDING A PACKET

In the previous section, we have calculated the probability that a CM sends a request in a randomly chosen contention minislot (i.e., τ), and the collision probability of a request transmitted on the channel (i.e., p_c). In this section, we will compute the mean access delay, T_{ad}, based on τ and p_c.

Fig. 6 shows a typical example of the contention process of DOCSIS. As stated before, the access delay is the interval between the time when a packet arrives at the CM and the time when the CM sends it out. Tad can be divided into two parts: a request access delay (T_{req_ad}) and a request scheduling delay (T_{req_sch}). T_{req_ad} is the time between the packet arrival and the successful transmission of the request. T_{req_sch} is the time between the successful transmission of the request and the beginning of the data transmission. Here T_{req_ad} and T_{req_sch} are two non-overlapping intervals. Thus,

$$E[T_{ad}] = E[T_{req\_ad}] + E[T_{req\_sch}] \tag{6}$$

In what follows, we derive both $E[T_{req\_ad}]$ and $E[T_{req\_sch}]$, which together obtain $E[T_{ad}]$.

A. T_{req_ad}

Let $\mathrm{E}[T_{req\_adi}]$ be the expectation of request access time of successful transmission at the i-th try. Thus, the mean request access time $\mathrm{E}[T_{req\_ad}]$ is expressed as follows.

$$\begin{aligned}
&E[T_{req\_ad}] \\
&= \sum_{i=1}^{16} p_i \times E[T_{req\_adi}] \\
&= p_1 \times E[T_{req\_ad1}] + p_2 \times E[T_{req\_ad2}] + \sum_{i=3}^{16} p_i \times E[T_{req\_adi}] \\
&= (1-p_c) \times \Bigg(W_0 + \frac{1}{2}N_{frame} - \frac{1}{2}N_C - 2 \\
&\qquad + \frac{W_0 + 4N_c - 1}{2N_{frame}} \Bigg) + (1-p_c)p_c \\
&\quad \times \Bigg\{ E\left[\left\lfloor \frac{B_{off2}}{N_c} \right\rfloor\right] \times N_{frame} + B_{off2} \bmod N_c] \\
&\qquad + \frac{3}{2}N_{frame} + \frac{1}{2}W_0 - N_c \Bigg\} + \sum_{i=3}^{16} (1-p_c)p_c^{i-1} \\
&\quad \times \Bigg\{ E\left[\left\lfloor \frac{B_{off2}}{N_c} \right\rfloor\right] \times N_{frame} + B_{off2} \bmod N_c \\
&\qquad + \sum_{k=2}^{i-1} \left\lceil \frac{B_{offk}}{N_c} \right\rceil \times N_{frame}] + \frac{3}{2}N_{frame} + \frac{1}{2}W_0 - N_c \Bigg\}.
\end{aligned}$$

where p_c is the collision probability in a minislot. Note that p_1 and p_i, $i \geq 2$, are expressed separately to reflect the fact that the retried requests need to wait till the next contention period.

$$b_{IDLE} = \frac{2(1-p_c)(1-2p_c)}{2N_i(1-p_c)(1-2p_c) + W_0(1-p_c)\left(1-(2p_c)^m\right) + (N_c+2)(1-2p_c)(1-p_c^m)}.$$

Thus

$$\tau = \frac{2(1-p_c)(1-2p_c)}{2N_i(1-p_c)(1-2p_c) + W_0(1-p_c)\left(1-(2p_c)^m\right) + (N_c+2)(1-2p_c)(1-p_c^m)} \tag{5}$$

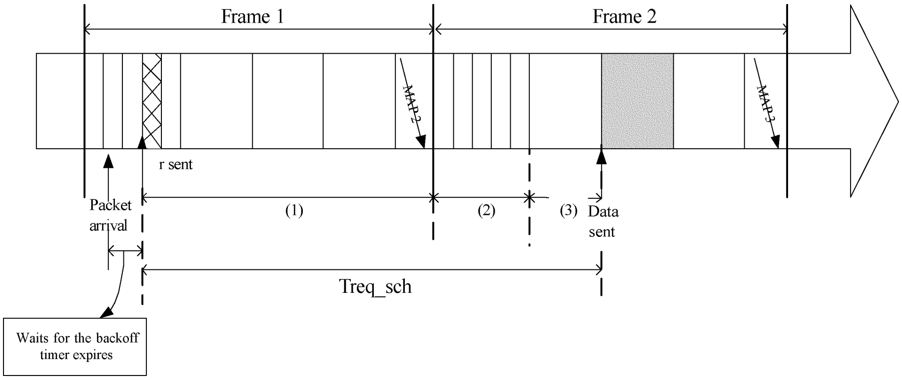


Fig. 7. An example of T_{req_sch}.

Moreover, B_{offi} is a random variable uniformly distributed in $[0, 2^{i-1}W_0 - 1]$. Taking the expectation of T_{req_ad}, we have

$$\begin{aligned}
&E[T_{req\_ad}]\\
&=(1-p_c)\Bigg\{W_0+\frac{1}{2}N_{frame}-\frac{1}{2}N_c\\
&\qquad -2+\frac{W_0+4N_c-1}{2N_{frame}}+\sum_{i=2}^{16}p_c^{i-1}\\
&\qquad \times\left(\frac{3}{2}N_{frame}-N_c+W_0\left(2^{i-2}+\frac{1}{2}\right)-2^{i-2}\right)\\
&\qquad +\sum_{i=2}^{16}p_c^{i-1}\times E\left[\left\lfloor\frac{B_{offi}}{N_c}\right\rfloor\right]\\
&\qquad \times(N_{frame}-N_c)+\sum_{i=2}^{15}\sum_{k=i}^{15}p_c^{k}E\left[\left\lceil\frac{B_{offi}}{N_c}\right\rceil\right]\Bigg\}.
\end{aligned} \tag{7}$$

where

$$E\left[\left\lfloor\frac{B_{offi}}{N_c}\right\rfloor\right]=\left\lfloor\frac{W_i-1}{N_c}\right\rfloor\left(1-\frac{N_c}{2W_i}-\frac{N_c}{2W_i}\left\lfloor\frac{W_i-1}{N_c}\right\rfloor\right),$$

and

$$E\left[\left\lceil\frac{B_{offi}}{N_c}\right\rceil\right]=\left\lceil\frac{W_i-1}{N_c}\right\rceil\left(1+\frac{N_c}{2W_i}-\frac{1}{W_i}-\frac{N_c}{2W_i}\left\lceil\frac{W_i-1}{N_c}\right\rceil\right).$$

B. T_{req_sch}

Next, we derive the expectation of the request scheduling time, E$[T_{req_sch}]$. Here the First Come First Served (FCFS) scheduling discipline is assumed by the CMTS. In addition, the request is assumed to be sent in the i-th minislot, and i is uniformly distributed in $[0, N_C - 1]$, as shown in Fig. 7.

For convenience, T_{req_sch} can be decomposed into three non-overlapping intervals, denoted (1)–(3) in Fig. 7. Interval (1) denotes the time to the end of the current frame after the request, say, r has successfully sent in a contention minislot; interval (2) is the contention minislots of the next frame; interval (3) denotes the total minislots of Data Grants IEs for all requests arriving at the CMTS earlier than the request r (thus their Data Grant IEs are arranged in the front of that of request r). Since these three intervals are non-overlapping, we can compute the expectation of each interval separately, as follows.

1) $E[(1)|i] = N_{frame} - i$, and $E[(1)] = \sum_{i=0}^{N_c-1}(N_{frame} - i) \times (1/N_c)$.

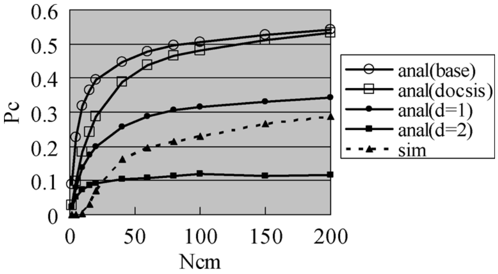


Fig. 8. Conditional collision probability (p_c) versus number of CMs.

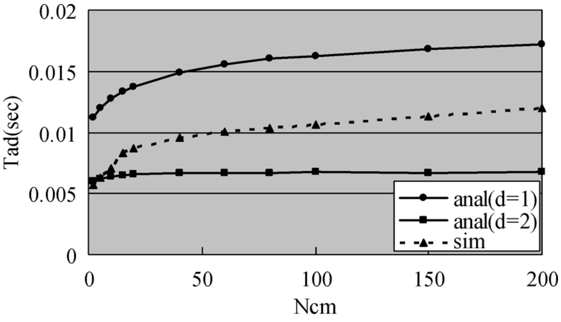


Fig. 9. Access delay versus the number of CMs.

2) $E[(2)] = N_c$

3) Assume that there are k requests successively received by the CMTS earlier than request r, and the probability of exactly one CM transmitting in a minislot is p_s. Thus,

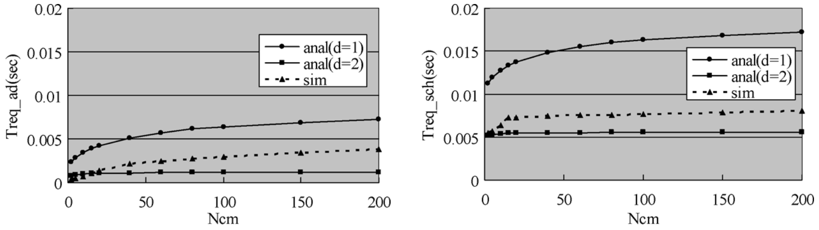
$$k=\begin{cases}0 & C_0^i(1-p_s)^i\\ 1 & C_1^ip_s^1(1-p_s)^{i-1}\\ \vdots \quad \text{with prob.} & \\ i-1 & C_{i-1}^ip_s^{i-1}(1-p_s)^1\\ i & C_i^ip_s^i\end{cases}$$

$$\begin{aligned}
E[k|i]&=\sum_{k=0}^{i}kC_k^i(1-p_s)^kp_S^{i-k}\\
&=p_s\times i\times(1-p_s+p_s)^{i-1}=ip_s
\end{aligned}$$

$$E\left[(3)\right]=\sum_{i=0}^{N_C-1}i\times p_s\times L_{ack}\times\frac{1}{N_c}=\frac{1}{2}p_sL_{ack}(N_c-1)$$

where $p_s = N_{CM}\tau(1-\tau)^{N_{CM}-1}$. From (3), we can obtain $\tau = 1 - (1 - p_c)^{1/(N_{CM}-1)}$. Thus,

$$p_s=N_{CM}(1-p_c)\left(1-(1-p_c)^{\frac{1}{N_{CM}-1}}\right). \tag{8}$$



(a) (b)

Fig. 10. The components of mean access delay. (a) $\mathrm{E}[T_{req\_ad}]$. (b) $\mathrm{E}[T_{req\_sch}]$.

From the expectations of these three intervals, we obtain

$$\begin{aligned} E&[T_{req\_sch}] \\ &= E\left[\text{interval}(1)\right] + E\left[\text{interval}(2)\right] + E\left[\text{interval}(3)\right] \\ &= N_{frame} + \frac{1}{2}(1 + p_s L_{ack})N_c - \frac{1}{2}p_s L_{ack} + \frac{1}{2} \\ &= N_{frame} + \frac{1}{2}(N_c + 1) + \frac{1}{2}N_{CM}L_{ack} \\ &\quad \times (N_c - 1)(1 - p_c)\left(1 - (1 - p_c)^{\frac{1}{N_{CM}-1}}\right) \qquad (9) \end{aligned}$$

V. Performance Evaluation

To see how accurate the analytical model is in predicting the access delay, some simulations using ns-2 are conducted. The simulation parameters are listed in Table I.

We first examine the correctness of the models in estimating the collision probability of a request transmitting on the channel, p_c. In this simulation, we vary the number of CMs performing TCP transfers from 0 to 200. The result is plotted as the dashed curve in Fig. 8. We find that the collision probability p_c increases with an increase in the number of CMs, which fits our intuition. The solid curves are plotted from the analysis model. The curve with $\mathrm{anal}(\mathrm{d} = 1)$ is the one taking TCP ACK intervals into account, and the curve with anal(d=2) is the one with delayed ACK (i.e., sending an ACK on receipt of every two packets). For reference, we also plot the first two models, i.e., the base and DOCSIS models. Recall that in Fig. 2, when there are few CMs in the system (i.e., less than 20 in this example), the probability that a request is sent via contention is small (i.e., less than 2% in this example). Most requests in this portion are sent via piggyback, which has been studied in [18]–[20]. Therefore, we are only interested in large number of active CMs doing TCP transfers (20 CMs in this example). The curves $\mathrm{anal}(\mathrm{d} = 1)$ and $\mathrm{anal}(\mathrm{d} = 2)$ provide the upper and lower bounds, respectively, for the simulation results. This is reasonable because in the simulation, the delayed ACK factor (i.e., d) is a fixed value, while in the simulation, it may be changed depending on the actual situation.

Then we compare the analytical access delay with simulation. Again, the dashed curve is from the simulation, and the solid ones are from the analysis. Fig. 9 shows that the curves of $\mathrm{anal}(\mathrm{d} = 1)$ and $\mathrm{anal}(\mathrm{d} = 2)$ provide the two bounds for the simulation result. Fig. 10 depicts the decomposition of the access delay from the analysis and the simulation. We see that the request scheduling delay, T_{req_sch}, has a more significant effect on the access delay.

VI. Conclusion

In this paper, we have provided an in-depth analytical study on the contention aspect of DOCSIS in HFC networks. In particular, an analytical model is developed to estimate the mean access delay of the contention mechanism for TCP transmissions in DOCSIS. The analytical model is comprised of two parts. We first calculate the probability that a CM sends a request in a randomly selected contention minislot. We then derive the expected access delay based on the probability developed in the first part. The analytical model is verified via simulation. The results show that our model can accurately provide bounds for the simulation results. Finally, we also observe the average RTT for TCP transmissions. In the system of interest, the upstream channel is no longer the bottleneck. The average RTT is mainly determined by the downstream queuing delay.

This work is a follow-up work of our study in [19]–[22]. The focus of this series of studies is on data service over "TCP" in DOCSIS networks, i.e., the impact of DOCSIS on the TCP performance, via contention and piggyback. In the future, we will further discuss the effect of channel error rates on the performance of TCP in DOCSIS. We will then shift our focus to multimedia service and investigate the impact of DOCSIS on TCP-friendly UDP congestion control mechanisms.

References

[1] CableLabs, MCNS Consortium, "Data-over-cable service interface specifications—Radio frequency interface specification," SP-RFIv1.1-106-001215, 2000.

[2] "IEEE Project 802.14/a Draft 3 Revision 1," Apr. 1998.

[3] D. Sala, J. O. Limb, and S. U. Khaunte, "Adaptive Control Mechanism for Cable Modem MAC Protocols," in *Proc. IEEE INFOCOM 1998*, April 1998.

[4] 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 Transaction on Broadcasting*, vol. 44, no. 4, pp. 427–435, Dec. 1998.

[5] W.-K. Kuo, S. Kumar, and C.-C. Jay Kuo, "Improved priority access, bandwidth allocation and traffic scheduling for DOCSIS cable networks," *IEEE Transactions on Broadcasting*, vol. 49, no. 4, pp. 371–382, Dec. 2003.

[6] M. T. Ali, R. Grover, and G. Stamatelos, "Performance evaluation of candidate MAC protocols for LMCS/LMDS networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 7, pp. 1261–1270, July 2000, and D Improved Priority Access, Bandwidth Allocation and Traffic Scheduling for DOCSIS Cable Networks. D. Falconer.

[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 Transactions on Broadcasting*, vol. 45, no. 2, pp. 196–205, June 1999.

[8] T. Orfanoudakis, N. Leligou, E. Meciu, and A. Harsanyi, "Evaluation of IP oriented HFC access protocols," in *Proc. Broadband Communications*, 2000.

[9] J. Martin and N. Shrivastav, “Modeling the DOCSIS 1.1/2.0 MAC protocol,” in *Proc. Computer Communications and Networks (ICCCN) 2003*, Oct. 2003.

[10] W.-T. Lee, K.-C. Chung, K.-C. Chu, and J.-Y. Pan, “DOCSIS performance analysis under high traffic conditions in the HFC networks,” *IEEE Transactions on Broadcasting*, vol. 52, no. 1, pp. 21–30, Mar. 2006.

[11] H. Balakrishnan, V. N. Padmanabhan, and R. H. Katz, “The effects of symmetry on TCP performance,” *ACM/Baltzer Mobile Networks and Applications (MONET)*, vol. 4, pp. 219–241, 1999.

[12] Y.-D. Lia and W. J. Liao, “Improving TCP performance over asymmetric networks,” in *Proc. IEEE ICC 2001*, Helsinki, Finland, June 2001.

[13] S. Varma, “Performance and buffering requirements of TCP applications in asymmetric networks,” in *Proc. IEEE INFOCOM ’99*, New York, March 1999.

[14] T. V. Lakshman, U. Madhow, and B. Suter, “Window-based error recovery and flow control with a low acknowledgement channel: A study of TCP/IP performance,” in *Proc. IEEE INFOCOM ’97*, Kobe, Japan, March 1997.

[15] R. Cohen and S. Ramanathan, “TCP for high performance in hybrid fiber coaxial broad-band access networks,” *IEEE/ACM Transactions on Networking*, 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*, Nov. 1998.

[17] R. Cohen and S. Ramanathan, “Using proxies to enhance TCP performance over hybrid fiber coaxial networks,” *Computer Communications*, vol. 20, no. 16, pp. 1502–1518, Feb. 1998.

[18] I. T. M. Chit *et al.*, “Performance evaluation of web browsing over hybrid fiber coaxial broad-/band networks,” in *Proc. IEEE ICON 1999*, Oct. 1999.

[19] W. Liao, “The behavior of TCP in DOCSIS-based CATV networks,” *IEEE Transactions on Communications*, vol. 54, no. 9, pp. 1633–1642, Sept. 2006.

[20] H. J. Ju and W. J. Liao, “Fast Request Transmission (FRT) in DOCSIS-based CATV networks,” in *Proc. IEEE International Conferences on Multimedia and Expo (ICME)*, Aug. 2002.

[21] ——, “Long Packet Deferment (LPD) in DOCSIS-based CATV networks,” in *Proc. ICCCN 2002*, Nov. 2002.

[22] W. J. Liao and H. J. Ju, “Adaptive slot allocation in DOCSIS-based CATV networks,” *IEEE Transactions on Multimedia*, vol. 6, no. 3, pp. 479–488, June 2004.

Kai-Chien Chang received the B.S. and M.S. degrees in Electronic Engineering from the Department of Electrical Engineering, National Taiwan University in 2001 and 2003, respectively. She is currently an engineer in Media-Tek Inc. in Taiwan. Her research interest is in wireless communications.

Wanjiun Liao received her BS and MS degrees from National Chiao Tung University, Taiwan, in 1990 and 1992, respectively, and the Ph.D. degree in Electrical Engineering from the University of Southern California, Los Angeles, California, USA, 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 actively involved in the international research community. She is currently an Associate Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE TRANSACTIONS ON MULTIMEDIA. She was a tutorial co-chair of IEEE INFOCOM 2004, and the TPC vice chair of IEEE Globecom 2005 (Autonomous Network Symposium). Dr. Liao 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 K. T. Li Young Research Award honored by ACM Taipei chapter 2003 for her research achievement. Dr. Liao was elected as one of Ten Distinguished Young Women in Taiwan in 2000 and was listed in the Marquis Who’s Who in 2001–2005, and the Contemporary Who’s Who in 2003.