Modeling the Contention Mechanism of DOCSIS in HFC Networks

Kai-Chien Chang and Wanjiun Liao
Department of Electrical Engineering and
Graduate Institute of Communication Engineering
National Taiwan University
Taipei, Taiwan

***Abstract*- The Hybrid Fiber Coax (HFC) network is a widely-used technology of broadband access networks to the home. The Data-Over-Cable Service Interface Specifications (DOCSIS) designed by the Multimedia Cable Network System Partners (MCNS) is the *de facto* standard in the cable industry. This paper studies the contention effect of DOCSIS in HFC networks. In particular, we focus on the performance of TCP. We develop a Markov model to estimate the probability of a CM transmitting a request in a randomly selected mini-slot. Based on the derived request transmission probability, the mean access delay for TCP transfers is calculated. We also conduct simulations via Network Simulator version 2 (ns-2) to validate the analysis.**

I. INTRODUCTION

The Hybrid Fiber Coax (HFC) network is a promising solution 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. Here the downstream channel refers to a channel transmitted from the Cable Modem Termination System (CMTS) at the headend to Cable Modems (CMs) at the home. The media access control (MAC) mechanism in HFC may obey the Data-Over-Cable Service Interface Specifications (DOCSIS) designed by 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 focus on DOCSIS's operation in this paper.

In DOCSIS, the upstream channel is modeled as frames of minislots. Each frame consists of contention minislots. The detail of each frame is specified via control messages called MAP, periodically transmitted on the downstream channel by the CMTS. The MAC operation of DOCSIS consists of two phases: requests by contention and requests by piggyback. Each time a CM has a packet to transmit, it should request for the assignment of a Data Grant IE in a later MAP. If it is a backlogged CM and is assigned a Data Grant IE in the newly received MAP, the CM piggybacks the request on an outgoing packet using the assigned Data Grant IE in the data minislots; otherwise, the CM contends for the use of the upstream channel via the mini-slots of the contention interval. If several CMs compete for one mini-slot in the contention interval, all requests are corrupted, and the collision resolution process is invoked. The collision resolution mechanism used in DOCSIS is truncated binary exponential backoff algorithm.

This paper studies the contention effect of DOCSIS on the performance of TCP in HFC networks. We develop a Markov model to estimate the probability of a CM transmitting a request in a randomly selected mini-slot. Based on the derived request transmission probability, the mean access delay for TCP transfers is calculated. We also conduct simulations via Network Simulator version 2 (ns-2) to validate the analysis.

The rest of the paper is organized as follows. In Sec. II, the system model is described. In Sec. III, the probability of a CM transmitting a request in a randomly selected mini-slot is estimated via a Markov chain. In Sec. IV, simulation results are presented. Finally, the conclusion is drawn in Sec. V.

II. SYSTEM MODEL

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 in contention, it randomly chooses a value in its window. For example, if W_0=16, the CM will choose a backoff value between 0 and 15 randomly. The backoff value indicates the number of contention mini-slots the CM has to defer before it can transmit the request. If a

0-7803-8623-X/04/$20.00

collision occurs, 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 mini-slot
p_c	The probability that a collision is seen by a request transmitting on the channel
p_s	The probability of a successful transmission in a contention mini-slot
N_{CM}	The number of CMs downloading files
N_{map}	The number of mini-slots described by a MAP
Nc	The number of contention mini-slots described by a MAP
W_i	The contention window size in backoff stage i
m	The maximum transmission attempt

Table I. The notations used in the analysis

C. Assumptions and some definitions

In this paper, we make the following assumptions:

1. The probability of a CM transmitting a request in a randomly selected contention mini-slot is a constant and is independent of that of other CMs.
2. The probability that a request in transmission suffers from a collision is a constant and is independent of the window size and the retransmission it has attempted.
3. The number of mini-slots that a MAP describes, i.e., N_{map}, is a constant.
4. The number of contention mini-slots in a DOCSIS frame, i.e., Nc, is also a 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., $W_0 < Nc$.

In this paper, we define the **backoff stage** as the number of collisions a request has suffered, and the **backoff counter** as the number of contention mini-slots a request has to defer before it can be transmitted. For convenience, we assume that the maximum backoff window size equals the maximum retries.

III. THE PROBABILITY OF A CM SENDING A REQUEST IN A RANDOMLY SELECTED MINI-SLOT

The objective of this paper is to develop a Markov model to estimate the probability τ of a CM transmitting a request in a randomly selected mini-slot.

A. DOCSIS Model

To model the binary exponential backoff mechanism, a discrete-time Markov chain is developed. Unlike pure-contention based MAC mechanisms such as CSMA/CD or CSMA/CA [5], in DOCSIS, the CMs transmitting a request via contention have to wait for the subsequent MAP to tell if the request is collided or not. An example is shown in Fig. 1. The CM transmits its request at t_0, and it does not know if the request succeeds until t_1. Thus, the only thing the CM can do is waiting in the three gray mini-slots in the contention interval.

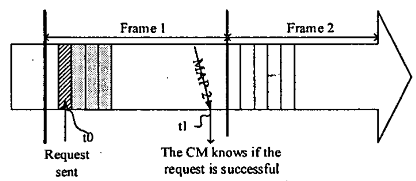

Figure 1. An example of DOCSIS MAC operation at a CM

To take the MAC operation in DOCSIS into consideration, another counter is introduced. The counter, called **waiting counter**, counts the number of contention mini-slots the CM has to wait before it can tell whether or not its request has sent successfully.

The state of the Markov chain is defined as $\{s(t), b(t), n_w(t)\}$, where $s(t)$ is the backoff stage of a request packet at time t, $b(t)$ is the backoff counter at time t, and $n_w(t)$ is the waiting counter at time t. State transitions take place at the end of each contention mini-slot. The Markov chain of a CM is shown in Fig. 2.

For convenience, we assume that the after successfully transmitted a request, the CM enters the IDLE state. When a CM has a packet to send, a request is generated and the CM leaves the IDLE state and starts its contention process. Its backoff stage is set to 0, and the backoff counter is randomly selected among [0, W_0-1]. When the backoff counter decreases to 0, the request is sent. After that, the CM has to wait additional n_w mini-slots for the next MAP to come to know if the request has arrived at the CMTS without collisions. If the probability of a request arriving in a CM is the same at any contention mini-slot, n_w is a

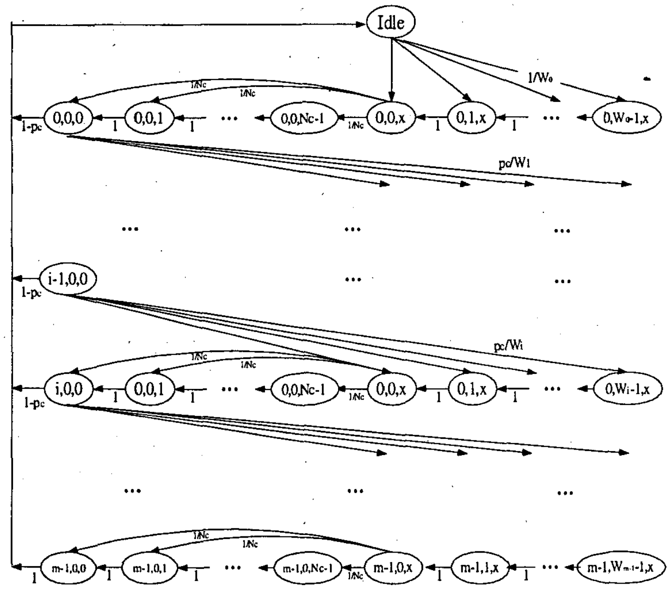

Figure 2. The Markov model of a CM

random variable uniformly distributed in [0, Nc-1].

The transition probabilities of the model are as follows.

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

The notation in this paper is defined as

$$P\{i_1, k_1 \mid i_0, k_0\} = P\{s(t+1) = i_1, b(t+1) = k_1 \mid s(t) = i_0, b(t) = k_0\}$$

The first equation in (1) is obtained from the assumption that after a request is successfully transmitted, a CM starts 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 $1/W_0$. Since the collision probability seen by a transmitting request is p_c, under the assumption of no channel error, 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 one is because the backoff counter decreases by one at every contention mini-slot. 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 sixth one is the decrement of the waiting counter, while the last one determines how many contention minislots that the CM has to wait before the next MAP is received.

The stationary distribution of the Markov chain is defined as follows.

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

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

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

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 mini-slot, τ, is the summation of the stationary probabilities of all states whose backoff counter is zero. That is, $\tau = \sum_{m} b_{0,i,x}$. Furthermore, in the steady state, this probability equals the stationary distribution of the IDLE state. This gives the following equation.

$$\tau = \frac{2(1-p_c)(1-2p_c)}{2(1-p_c)(1-2p_c)+W_0(1-p_c)(1-(2p_c)^m)+(N_c+2)(1-2p_c)(1-p_c^m)} \quad (2)$$

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

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

Let $\tau = f_1(p_c)$, and $p_c = g(\tau)$, 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 monotonic increasing. Moreover, $f_1(0) > g^*(0)$ and $f_1(1) < g^*(1)$. Thus, we can conclude that there is a unique solution of f_1 and g. Thus, we can derive τ and p_c by numerical techniques.

B. TCP Model

The DOCSIS model is developed without considering the upper protocol being TCP. Now we turn to handle 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. There must be some idle stages between the CM knowing if the previous request is successfully transmitted and starting to send another request. The revised Markov chain model is shown in Fig. 3.

The transition probabilities of this model are derived in (4). The transition probabilities of this model are basically the same as those of the DOCSIS model, except for the latest equation, which represents for the decrement of the idle stages. The problem is how to determine the number of idle stages, Ni. 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.

$$\begin{cases} P\{0,j,x \mid I_0\} = 1/W_0, & j \in (0,W_0-1) \\ P\{I_{Ni-1} \mid i,0,0\} = 1-p_c, & i \in (0,m-2) \\ P\{i,j,x \mid i-1,0,0) = p_c/W_i, & i \in (1,m-1)\ j \in (0,W_i-1) \\ P\{i,j,x \mid i,j+1,x) = 1, & i \in (0,m-1)\ j \in (0,W_i-2) \\ P\{I_{Ni-1} \mid m,0,0\} = 1 & \\ P\{i,0,k \mid i,0,k+1) = 1, & i \in (0,m-1)\ k \in (0,Nc-2) \\ P\{i,0,k \mid i,0,x) = 1/Nc, & i \in (0,m-1)\ k \in (0,Nc-2) \\ P\{I_l \mid I_{l+1}\} = 1, & l \in (0,Ni-2) \end{cases} \quad (4)$$

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_{dCM} is the number of CMs doing 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 mini-slots. Since in our model transitions happen only at the end of contention mini-slots, we map the ACK interval into contention mini-slots spanning over many frames (i.e., only counting contention mini-slots in frames) as

$$N_i = \left\lfloor \frac{\text{ACK interval}}{N_{map}} \times N_c \right\rfloor.$$

We assume that after a request is successfully transmitted, the next request will arrive after Ni 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 MAP size, N_{map} can be estimated as follows.

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

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 mini-slots in a frame.

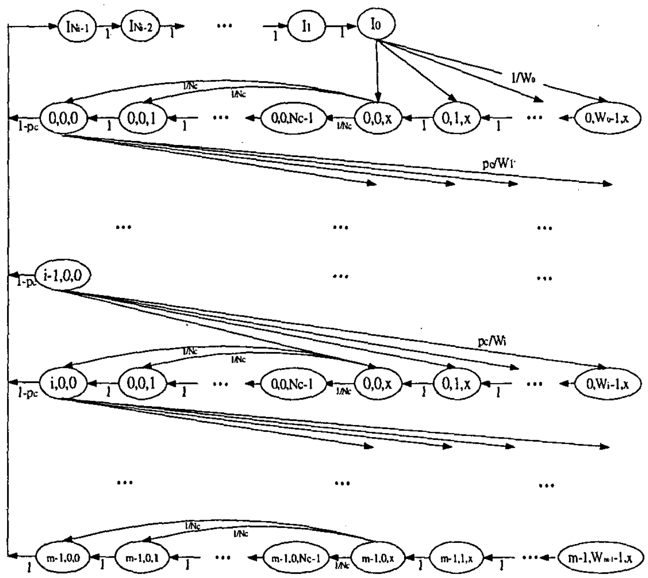

Figure 3. The Markov model considering the TCP ACK mechanism

As a result, given the number of downloading CMs, the idle stages, Ni, can be derived accordingly. The stationary distribution of state I_0 is then expressed as

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

Thus,

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

IV. PERFORMANCE EVALUATION

To verify the analytical results, some simulations using ns-2 are conducted. The network architecture and parameters used in the simulation are shown in Fig. 4 and Table II, respectively.

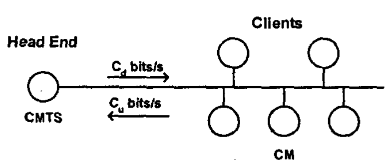

Figure 4. Network topology

Parameter	Value
Cd	26.97Mbps
Cu	2.56Mbps
t_{ms}	50 us
Nc	50 mini-slots
Ldata	1024 bytes
Lack	64 bytes
W_0	16
m	16

Table II. The parameters used in the simulation.

We first examine the correctness of the models estimating the probability that a collision seen by a request transmitting in the channel, p_c. In this simulation, we vary the number of CMs performing TCP transfers from 0 to 200. The result is plotted in the dashed curve in Fig. 5. We see that the collision probability p_c increases as the number of CMs increases, which fits our intuition. The solid curves are plotted from the analysis model using the same parameters listed in Table II. The curve with anal(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. We find that the curves with anal(d=1) and anal(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 examine the access delay for TCP transfers via the analysis and via the simulation. The access delay, T_{ad}, for TCP transfers is defined as the interval between the time that a data packet arrives at a CM and the time that the packet is sent by the CM, as shown in Fig.7. The expression of TCP access delay for DOCSIS networks can be found in [6].Again, the dashed curve is from the simulation, and the solid ones are from the analysis. Fig. 6 shows that the curves of anal(d=1) and anal(d=2) provide the two bounds for the curve of the simulation result.

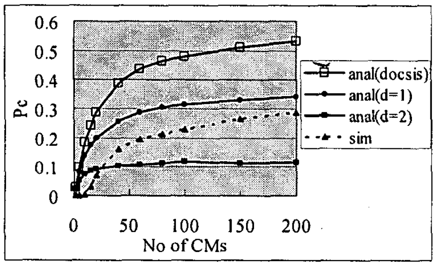

Figure 5. p_c vs. no. of CMs

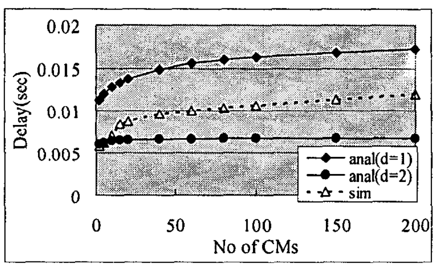

Figure 6. Access delay vs. no of CMs

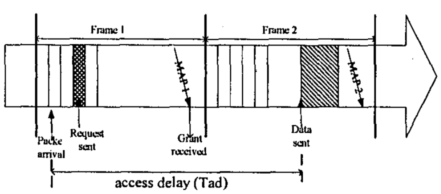

Figure 7. Definition of access delay

V. CONCLUSION

In this paper, we have studied the contention effect of DOCSIS on the performance of TCP in HFC networks. In particular, an analytical model is developed to calculate the probability that a CM sends a request in a randomly selected contention minislot. We also verify the analytical model via simulation. The results show that our model can accurately provide bounds for the simulation results.

ACKNOWLEGMENT

This work was supported in part by MOE program for Promoting Academic Excellence of Universities under grant number 89E-FA06-2-4-7, and in part by the National Science Council, Taiwan, under grant number NSC93-2213-E-002-001.

REFERENCES

[1] CableLabs, Data-Over-Cable Service Interface Specifications (DOCSIS) Radio Frequency Interface Specification, MCNS Consortium, 2000, SP-RFIv1.1-I07-010829

[2] IEEE 802.14 Working Group. Media Access and Control IEEE Std 802.14, Draft 3 Revision 1, IEEE 802.14 Working Group, April 1998.

[3] Huei-Jiun Ju and Wanjiun Liao, "Fast Request Transmission in DOCSIS-based CATV Networks," Proc. IEEE International Conference on Multimedia and Expo (ICME) '02, Lausanne, Switzerland, Aug. 2002.

[4] Huei-Jiun Ju and Wanjiun Liao, "Long Packet Deferment in DOCSIS-based CATV Networks," Proc. IEEE International Conference on Computer Communications and Networks (IC3N) '02, Miami, Florida, Oct. 2002.

[5] Giuseppe Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", IEEE Journal on Selected Areas in Communications, vol. 18, no. 3, March 2000, pp. 535-547

[6] Kai-Chien Chang and Wanjiun Liao, "Contention Effect of DOCSIS on the Performance of TCP in Hybrid Fiber Coax (HFC) Networks," Technical Report TP-NTUEE2003-02-103.