Statistically Optimized Minislot Allocation for Initial and Collision Resolution in Hybrid Fiber Coaxial Networks

Wei-Ming Yin and Ying-Dar Lin, *Member, IEEE*

***Abstract*—In a two-way hybrid fiber coaxial (HFC) network, the headend broadcasts in downstream channels, whereas all stations share the upstream channels. Hence, collision occurs when multiple stations send their bandwidth requests in a minislot. The headend determines how many minislots to allocate to manage collisions. This paper proposes a minislot allocation (SOMA) algorithm to optimize minislot throughput based on statistical estimation. A time proportional scheme is adopted to estimate the number of new requests in the initial resolution process. In addition, the number of retry requests in the collision resolution process is estimated by looking up a predetermined table of the most likely number of requests (MLR). In addition, SOMA is modified to reduce the request access delay by relaxing its allocation policy in a specific situation. We use a self-similar traffic model for simulation and analysis to compare SOMA with the optimal bound and the 3-ary tree algorithm.**

***Index Terms*—Allocation, collision resolution, DOCSIS, hybrid fiber coax, IEEE 802.14, minislot, statistical estimation, upstream.**

I. Introduction

HYBRID FIBER Coax (HFC) technology provides both upstream and downstream channels in coaxial cable networks. Upgraded amplifiers to support two-way amplification and fiber replacement for long distance transmission allow subscribers to send data back to the headend. To help make cable modems and headends designed by different vendors interoperable, two standards—*IEEE 802.14* [1], [2] and *Data-Over-Cable Service Interface Specification (DOCSIS)* [3]—have been produced. These standards specify physical and medium access control (MAC) layers. Both IEEE 802.14 and DOCSIS view an upstream channel as a stream of *minislots*. Basically, the headend allocates some upstream minislots as *contention minislots* (CS) where stations send bandwidth requests, and the others as *data minislots* (DS) where stations send their data. To reduce bandwidth wastage from collisions, stations first send small requests to the headend, with each request fitting into a single minislot and subject to collision. In the HFC networks, the request contention process comprises two phases [4]: 1) the *initial resolution* in which newly active stations follow a first transmission rule (FTR) to send their requests, and 2) the *collision resolution* in which the stations retransmit collided requests according to a retransmission rule (RTR). FTRs can be further classified as two types: "free-access" [5] and "blocked-access" [6]. The former allows new requests to be transmitted in the minislots used to resolve collisions, while the latter forbids this. RTRs can also be divided into two types [7]: "free-for-all" and "perfectly scheduled." In the former, *all* collided requests can be retried in the next opportunity, as in p-persistent [8]. However, in the latter, collided requests are split into a number of independent sets, as in n-ary tree walk [9], [10]. Table I summarizes the access strategies used in initial and collision resolution in IEEE 802.14 and DOCSIS.

TABLE I
Initial and Collision Resolution in IEEE 802.14 and DOCSIS

access strategies	FTR in Initial Resolution	RTR in Collision Resolution
IEEE 802.14	blocked-access (with priority, FIFO)	perfectly-scheduled (n-ary tree)
DOCSIS	free-access (no restriction)	free-for-all (exponential backoff)

This work first defines *minislot throughput* as a target to optimize. The statistical observation that the optimal number of allocated minislots equals the number of requests to resolve implies that the core problem lies in accurately *estimating* the number of requests. Several researchers have addressed these issues. Abi-Nassif and Lee [11] proposed an estimation scheme to measure the number of requests in DOCSIS networks. However, their study assumed the number of retransmitted requests to be negligibly small compared to the number of new requests, which does not reflect the real situation. Based on the observation that the ratio of the number of request minislots to the number of ATM cells should not exceed 3, Sriram [12] presented a heuristic algorithm to compute the number of minislots to allocate. However, its multiplication factor in the heuristic formula is difficult to determine dynamically. Thus, the algorithm cannot achieve the optimal minislot throughput. A previous study by the present authors [13] also listed heuristic recommendations on minislot allocation. The rest of this paper is organized as follows. Section II gives formal statements for minislot allocation problems. Section III describes the proposed statistically optimized minislot allocation algorithm. Section IV then analyzes the minislot throughput. Numerical results are presented in Section V. Conclusions are finally made in Section VI.

Manuscript received October 13, 1999; revised March 7, 2000. This work was supported in part by MOE Program of Excellence Research 89-E-FA04-1-4, National Science Council, and in part by Mentor Data Systems Inc., Taiwan.

The authors are with the Department of Computer and Information Science, National Chiao-Tung University, Hsin-Chu 30050, Taiwan (e-mail: wmyin@cis.nctu.edu.tw; ydlin@cis.nctu.edu.tw).

Publisher Item Identifier S 0733-8716(00)07122-5.

II. Problem Statement

The minislot allocation problem consists of two parts: the initial allocation for new requests and the collision allocation for collided requests. The minislot throughput for initial resolution, σ_I, is defined as S_I/A_I where A_I and S_I are the numbers of *allocated* minislots and *successful* minislots, respectively. Assume that R is the number of requests, the most likely S_I is a function of A_I and R, say $\Psi(A_I, R)$; and again, A_I is believed to be a function of R, say $f(R)$. Thus, σ_I can be rewritten as

$$\sigma_I = \frac{S_I}{A_I} = \frac{\Psi(A_I, R)}{A_I} = \frac{\Psi(f(R), R)}{f(R)}. \tag{1}$$

Therefore, to maximize the minislot throughput in initial resolution, R must be estimated as accurately as possible, and the optimal functions f and Ψ must be found. Meanwhile, the minislot throughput for collision resolution, σ_C, is defined as

$$\sigma_C = \frac{R - S_I}{\sum_{j=1}^{j=K} A_j - A_I} \tag{2}$$

where K is the number of *minislot clusters* allocated to fully resolve R requests, and A_j is the number of minislots allocated in cluster j. Notably, a minislot cluster is a set of contiguous minislots allocated in a round to resolve a set of new requests or a set of retry requests which were collided in the same minislot. The optimal scheme for collision resolution should allocate the right amount of A_j to optimize σ_C.

III. Statistically Optimized Minislot Allocation

A. Motivation

Assume that there are r requests contending for m minislots. The probability of a minislot being successful in transmitting a single request is

$$f_r(m) = \binom{r}{1}\frac{1}{m}\left(1 - \frac{1}{m}\right)^{r-1}, \qquad \text{where } r,\, m > 0.$$

Differentiating the above with respect to m produces

$$f_r'(m) = r\left[\frac{1}{m^3}\left(1 - \frac{1}{m}\right)^{r-2}(r - m)\right], \quad \text{where } r,\, m > 0.$$

Letting the above equation equal zero results in $m = r$. Restated, allocating r minislots to resolve r requests maximizes the minislot throughput. This transforms the issue of the optimal minislot allocation into how to accurately estimate the number of requests. We hereby propose a *statistically optimized minislot allocation (SOMA)* algorithm to estimate the number of requests for each resolution phase.

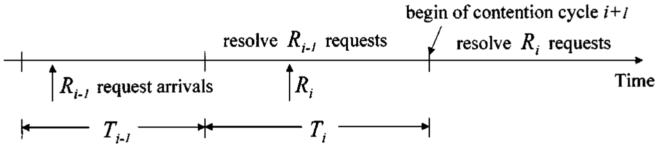

begin of contention cycle i+1
resolve R_{i-1} requests
resolve R_i requests
Time
R_{i-1} request arrivals
R_i
T_{i-1}
T_i

Fig. 1. Contention cycles $i - 1,\ i,\ i + 1$.

number of requests		number of collided minislots										
		0	1	2	3	4	5	6	7	8	9	...
number of successful minislots	0	0	2	4	6	8	10	12	16	21	23	
	1	1	3	5	7	9	12	14	17	21	25	
	2	2	4	6	8	10	13	15	18	21	25	
	3	3	5	7	9	11	14	16	20	21	25	
	4	4	6	8	10	12	15	18	20	23	26	...
	5	5	7	9	11	13	16	18	21	25	28	
	6	6	8	10	12	15	17	20	22	26	30	
	7	7	9	11	13	16	18	21	24	27	31	
	8	8	10	12	14	17	19	22	25	29	31	
	9	9	11	13	15	18	20	23	26	30	32	
	⋮					⋮						

Fig. 2. A part of MLR $(A,\ S,\ C)$ table with $A = 20$.

B. Initial Estimation—Time Proportional

For networks that adopt blocked access strategy as the FTR, requests arriving during contention cycle $i - 1$ can be resolved in contention cycle i, as illustrated in Fig. 1. Similarly, the requests joining contention cycle $i+1$ are those that arrive during contention cycle i. The time proportional scheme is statistically valid, i.e., $R_i/R_{i-1} \approx T_i/T_{i-1}$. Therefore, R_i can be approximated by the following expressions

$$A_{i+1,\,1} = R_i \approx \text{Round}\left(\frac{R_{i-1} \times T_i}{T_{i-1}}\right) \tag{3}$$

where $A_{i+1,\,1}$ is the number of allocated minislots in the 1st cluster of contention cycle $i + 1$. Therefore, to maximize minislot throughput, σ_I, the headend allocates $A_{i+1,\,1}$ minislots for initial resolution of the contention cycle $i + 1$. If the additional access control is in place, requests arriving during cycle $i - 1$ might not be allowed to access cycle i, which would affect the accuracy of our time proportional scheme.

C. Collision Estimation—MLR-Bbased

1) MLR (Most Likely Number of Requests) Table: An MLR table uses the pattern of contention results for finding the most likely number of requests. The table can be indexed by a 3-tuple $(A,\ S,\ C)$ where A, S, and C denote the number of allocated minislots, successful minislots, and collided minislots, respectively. Given A, we start from 1 request to 500 requests for these requests to contend A minislots and repeat each case 10^5 times. For the entry indexed by $(A,\ S,\ C)$, we select the number of requests that occurs most frequently from the possible requests that can lead the scenario $(A,\ S,\ C)$. Fig. 2 is a part of the MLR table for 20 allocated minislots. Since the MLR table reflects the contention result which is time- independent, its estimation error is *independent* of the traffic models.

2) Estimation Scheme: For each contention round after the first one in contention cycle i, SOMA performs the following steps to allocate minislots. Herein, M_{ij}, A_{ij}, S_{ij}, and C_{ij} represent the most likely number of requests, the number of allocated minislots, the number of successful minislots, and the number of collided minislots, respectively, in cluster j of cycle i.

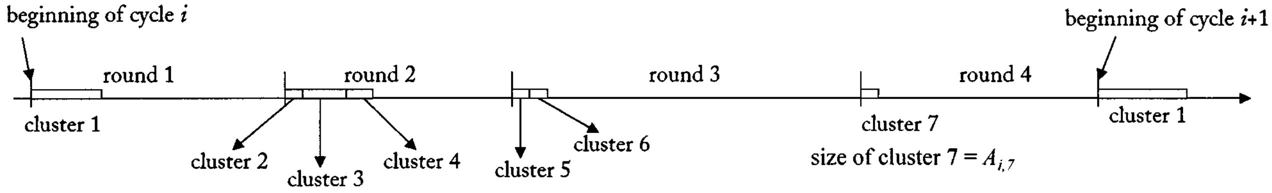


Fig. 3. Minislot allocation components.

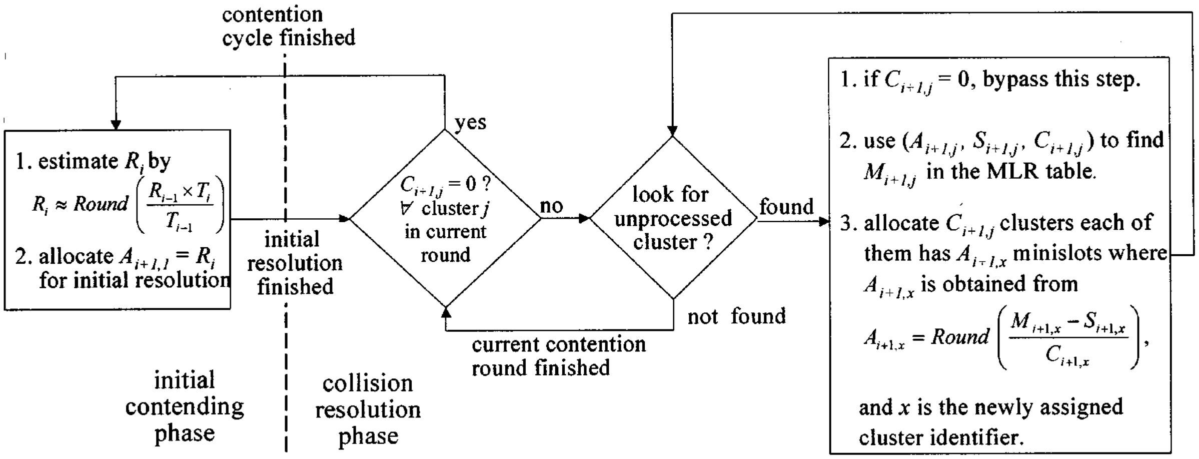

contention cycle finished
1. estimate R_i by
$R_i \approx Round\left(\frac{R_{i-1}\times T_i}{T_{i-1}}\right)$
2. allocate $A_{i+1,1} = R_i$ for initial resolution
initial resolution finished
yes
$C_{i+1,j} = 0$? ∀ cluster j in current round
no
look for unprocessed cluster ?
found
not found
current contention round finished
1. if $C_{i+1,j} = 0$, bypass this step.
2. use $(A_{i+1,j}, S_{i+1,j}, C_{i+1,j})$ to find $M_{i+1,j}$ in the MLR table.
3. allocate $C_{i+1,j}$ clusters each of them has $A_{i+1,x}$ minislots where $A_{i+1,x}$ is obtained from
$A_{i+1,x} = Round\left(\frac{M_{i+1,x} - S_{i+1,x}}{C_{i+1,x}}\right)$,
and x is the newly assigned cluster identifier.
initial contending phase
collision resolution phase

Fig. 4. The flow chart of the SOMA.

Fig. 3 illustrates the relationship among contention cycles, contention rounds, and minislot clusters.

1) If all C_{ij}'s in this round equal zero, all requests are resolved and a new contention cycle, i.e., cycle $i+1$, is initiated; otherwise, the following steps must be performed.
2) Look for a cluster j with a positive C_{ij}. If no such cluster exists, the minislot allocation scheme in this round is terminated. Otherwise, use the MLR scheme to guess M_{ij} based on the contention pattern (A_{ij}, S_{ij}, C_{ij}).
3) Then, the number of collided requests is estimated as $M_{ij} - S_{ij}$.
4) Finally, allocate C_{ij} minislot clusters, each with a separated identifier x which is not used in this contention cycle. Each allocated minislot cluster contains A_{ix} minislots determined from the expression

$$A_{ix} = \text{Round}\left(\frac{M_{ij} - S_{ij}}{C_{ij}}\right). \tag{4}$$

Go back to step 2.

Combining the above two subsections, Fig. 4 illustrates the flow chart of SOMA.

D. Relaxed SOMA

Fig. 5 displays the probability distribution of collision multiplicity, i.e., the number of transmissions in a collided minislot, which indicates that for most R/M's, there is a high probability that a collided minislot involves 2 requests. According to SOMA, the minislot throughput and the average number of rounds required to resolve these two requests are $0.5 \times 1 + 0.5 \times 0 = 0.5$ and 2, the solution of equation $x = 0.5 \times 1 + 0.5 \times (1 + x)$, respectively. In other words, these two requests are likely, with probability 0.5, to play another round, which significantly delays request access. Based on this observation, we modify the original SOMA to reduce the request access delay. In the relaxed SOMA, the headend always allocates 3 minislots to resolve a collision if it estimates two collided requests. Therefore, the minislot throughput and the average number of rounds to resolve the above situation now are $0.667 \times 0.667 + 0.333 \times 0 = 0.444$ and 1.5, the solution of equation $x = 0.667 \times 1 + 0.333 \times (1 + x)$, respectively. Although the minislot throughput decreases by 11% (from 0.5 to 0.444), the request access delay is shortening by 25% (from 2 to 1.5). Thus, in this situation, relaxing SOMA would appear to be efficient. Fig. 6 shows the flow chart of the relaxed SOMA.

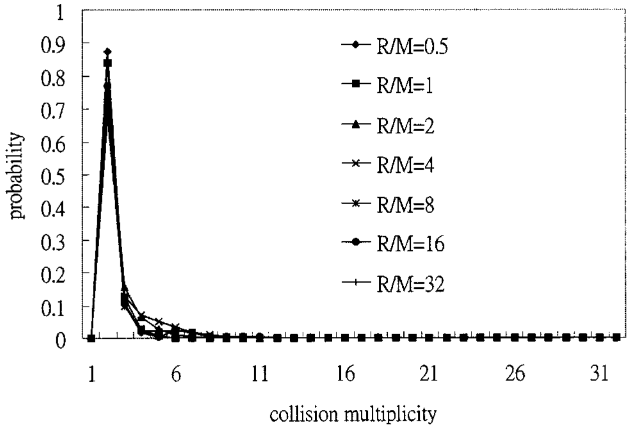


Fig. 5. Probability distribution of collision multiplicity.

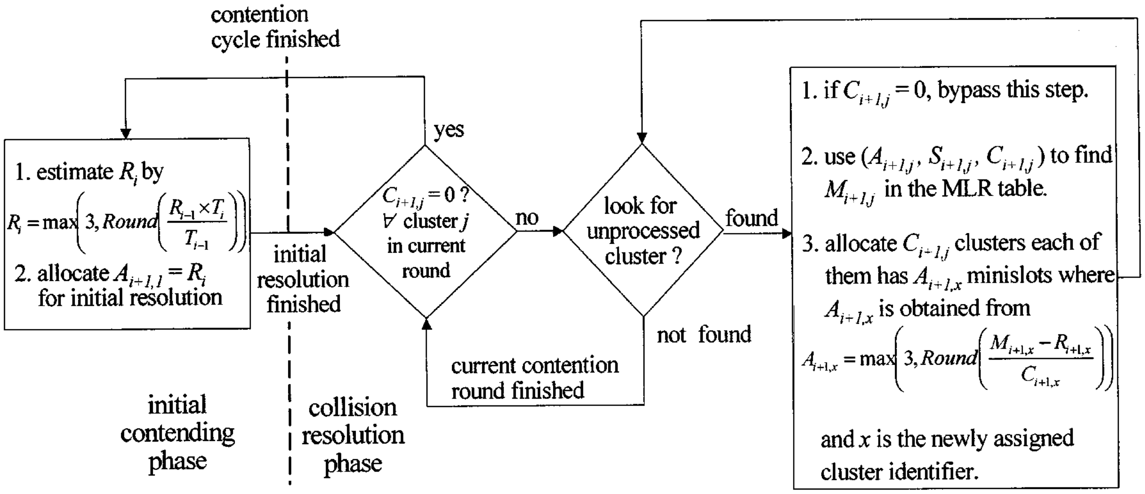

contention cycle finished
1. estimate R_i by
$R_i = \max\left(3, Round\left(\frac{R_{i-1}\times T_i}{T_{i-1}}\right)\right)$
2. allocate $A_{i+1,1} = R_i$ for initial resolution
yes
initial resolution finished
$C_{i+1,j} = 0$? $\forall$ cluster j in current round
no
look for unprocessed cluster ?
found
not found
current contention round finished
1. if $C_{i+1,j} = 0$, bypass this step.
2. use $(A_{i+1,j}, S_{i+1,j}, C_{i+1,j})$ to find $M_{i+1,j}$ in the MLR table.
3. allocate $C_{i+1,j}$ clusters each of them has $A_{i+1,x}$ minislots where $A_{i+1,x}$ is obtained from
$A_{i+1,x} = \max\left(3, Round\left(\frac{M_{i+1,x} - R_{i+1,x}}{C_{i+1,x}}\right)\right)$
and x is the newly assigned cluster identifier.
initial contending phase
collision resolution phase

Fig. 6. The flow chart of the relaxed SOMA.

IV. Throughput Analysis

A. Analysis Model

The minislot throughput of initial allocation is obtained from (1). We define an initial resolution estimator and a complete collision function to derive A_I and S_I. Afterwards, a collision resolution estimator is presented to derive the total number of allocated minislots in a contention cycle, and hence calculate minislot throughput of collision resolution base on (2). We define these estimators as follows. Denote α^R as the initial resolution estimator for R requests, and let α_k^R indicate the probability of R requests being estimated as k requests; thus, α^R could be defined as

$$\alpha^R = \left(\alpha_1^R \quad \alpha_2^R \quad \cdots \quad \alpha_k^R \quad \cdots\right), \qquad \text{where} \sum_{r=1}^{\infty} \alpha_r^R = 1.$$

In the optimal scheme, α_k^R equals 1 if k equals R; otherwise, α_k^R equals 0. Similarly, denote $\boldsymbol{\beta}^R$ as the collision resolution estimator for R requests. Letting β_k^R indicate the probability of having R requests estimated as k requests, there exists

$$\beta^R = \left(\beta_1^R \quad \beta_2^R \quad \cdots \beta_k^R \quad \cdots\right), \qquad \text{where} \sum_{r=1}^{\infty} \beta_r^R = 1.$$

In the optimal scheme, β_k^R equals 1 if k equals R; otherwise, β_k^R equals 0. In the 3-ary scheme, β_k^R equals 1 if k equals 3; otherwise, β_k^R equals 0. Both initial resolution and collision resolution estimators for the SOMA scheme are statistical results calculated from the simulation.

B. Minislot Throughput for Initial Resolution

Since minislots are allocated according to the estimated number of requests, the mean number of minislots allocated, given R requests in the initial contending phase, in initial resolution is

$$A_{R,I} = \sum_{r=1}^{\infty} r\alpha_r^R. \tag{5}$$

To facilitate deduction of the number of successful requests, we define a complete collision function, $\psi(r, m)$, which gives the number of possible combinations that no request succeeds when r requests contend for m minislots, as

$$\begin{aligned}
\psi(1, \_) &= 0\\
\psi(\_, 1) &= 1\\
\psi(2, m) &= m\\
\psi(r, m) &= m^r - \sum_{i=1}^{m-1} \binom{m}{i}\binom{r}{i} i!\psi(r-i, m-i),\\
&\qquad \text{where } r > m
\end{aligned}$$

and

$$\begin{aligned}
\psi(r, m) &= m^r - \sum_{i=1}^{r-2} \binom{m}{i}\binom{r}{i} i!\psi(r-i, m-i)\\
&\quad - \binom{m}{r} r!, \qquad \text{where } r \le m.
\end{aligned}$$

Accordingly, the formula giving the mean number of successful requests in the initial resolution when R requests are participating in a contention cycle is derived as

$$\begin{aligned}
S_{R,I} = &\sum_{r=1}^{R-1} \alpha_r^R \left(\sum_{i=1}^{r-1} i \times \frac{\binom{r}{i}\binom{R}{i} i!\psi(R-i, r-i)}{r^R}\right)\\
&+ \sum_{r=R}^{\infty} \alpha_r^R \left(\sum_{i=1}^{R-2} i \times \frac{\binom{r}{i}\binom{R}{i} i!\psi(R-i, r-i)}{r^R}\right.\\
&\left. + R\frac{\binom{r}{R} R!}{r^R}\right).
\end{aligned} \tag{6}$$

The above formula is an expectation form of mean successful requests given a total of R requests, whereas the forms within the parentheses are themselves expectation forms of mean number of successful requests given a total of r requests. Denote $A_{R,I}^{(O)}$ and $A_{R,I}^{(P)}$ as the mean number of minislots allocated for initial resolution in the optimal and the time proportional schemes, respectively, and let $S_{R,I}^{(O)}$ and $S_{R,I}^{(P)}$ be the mean number of suc-

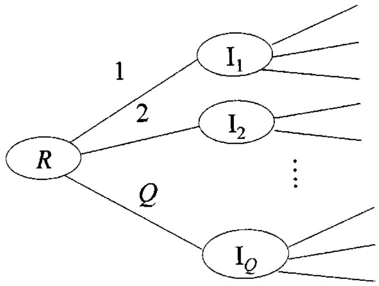


Fig. 7. A contention tree.

cessful requests in initial resolution in the optimal and the time proportional schemes, respectively,. The minislot throughputs for initial allocation in the optimal scheme and the proportional scheme are $S_{R,I}^{(P)}/A_{R,I}^{(P)}$ and $S_{R,I}^{(O)}/A_{R,I}^{(O)}$, respectively.

C. Minislot Throughput for Collision Resolution

Fig. 7 depicts a contention tree for blocked-access collision resolution protocols. R is the number of requests collided initially, and I_k is the number of requests choosing the kth subtree out of Q subtrees.

Denote Y_R as the number of minislots in a contention tree where R requests are collided initially. We hereby obtain the following recursive formula

$$Y_R = 1 + \sum_{j=1}^{Q} Y_{I_j}, \qquad \text{where } R \geq 2$$

with the initial conditions $Y_0 = Y_1 = 1$. The probability generating function of Y_R is thus derived as

$$\begin{aligned} G_R(s) &\equiv \sum_{k=0}^{\infty} \Pr\{Y_R = k\} \times s^k \\ &= E\left[s^{Y_R}\right] \\ &= E\left[s^{Y}|R\right] \\ &= E\left[E\left[s^{1+\sum_{j=1}^{Q} Y_{I_j}}\right]\middle| R\right]. \end{aligned}$$

Expanding the right-hand side of the equation, we have

$$\begin{aligned} G_R(s) &= sE\left[G_{I_1}(s)G_{I_2}(s)\cdots G_{I_Q}(s)\right] \\ &= s\sum_{i_1\cdots i_Q}^{R} \binom{R}{i_1\cdots i_Q} \prod_{j=1}^{Q} p_j^{i_j} G_{i_j}(s) \end{aligned}$$

where p_j is the probability of choosing the jth subtree. The initial conditions are now transformed into $G_0(s) = G_1(s) = s$. Let T_R denote the first moment of Y_R. T_R is obtainable by simply taking the first derivative of $G_R(s)$ with respect to s and evaluating it at $s = 1$, which yields

$$\begin{aligned} T_R &= E\left[Y_R\right] \\ &= \left.\frac{dG_R(s)}{ds}\right|_{s=1} \\ &= 1 + \sum_{j=1}^{Q}\sum_{i_j=0}^{R} \binom{R}{i_j} p_j^{i_j} \left(1-p_j\right)^{R-i_j} T_{i_j}. \end{aligned} \tag{7}$$

According to the equation, the size of a contention tree equals one plus Q times of the mean size of a subtree. Generally, $p_j = 1/Q$. Furthermore, Q equals R and 3 in the optimal scheme and the 3-ary scheme, respectively. Denote $T_R^{(O)}$ and $T_R^{(T)}$ as the sizes of contention trees of the optimal scheme and the 3-ary scheme, respectively, given R requests collided initially. We obtain their recursive formulas as

$$\begin{aligned} T_R^{(O)} = &\left(1 + \sum_{j=1}^{R}\sum_{i_j=0}^{R-1} \binom{R}{i_j} \frac{(R-1)^{R-i_j}}{R^R} T_{i_j}^{(O)}\right) \\ &\times \left(1 - \frac{1}{R^{R-1}}\right)^{-1} \end{aligned}$$

and

$$\begin{aligned} T_R^{(T)} = &\left(1 + \sum_{j=1}^{3}\sum_{i_j=0}^{R-1} \binom{R}{i_j} \left(\frac{2^{R-i_j}}{3^R}\right) T_{i_j}^{(T)}\right) \\ &\times \left(1 - \frac{1}{3^{R-1}}\right)^{-1}. \end{aligned}$$

Denote $T_R^{(\mathrm{MLR})}$ as the size of an MLR-based contention tree, and the number of requests collided initially is taken as the estimated number of requests. Thus, we have its expectation form as

$$T_R^{(\mathrm{MLR})} = \sum_{r=1}^{\infty} \left(\beta_r^R \times T_r^{(\mathrm{MLR})}\right).$$

Following the idea of (7), we derive $T_R^{(\mathrm{MLR})}$ as

$$\begin{aligned} T_R^{(\mathrm{MLR})} = 1 + \sum_{r=1}^{\infty} \beta_r^R &\left(\sum_{j=1}^{r}\sum_{i_j=0}^{R} \binom{R}{i_j} \left(\frac{1}{r}\right)^{i_j}\right. \\ &\left.\times \left(1 - \frac{1}{r}\right)^{R-i_j} T_{i_j}^{(\mathrm{MLR})}\right). \end{aligned}$$

Again, the recursive form is

$$\begin{aligned} T_R^{(\mathrm{MLR})} = &\left(1 + \sum_{r=1}^{\infty} \beta_r^R \left(\sum_{j=1}^{r}\sum_{i_j=0}^{R-1} \binom{R}{i_j}\right.\right. \\ &\left.\left.\times \frac{(r-1)^{R-i_j}}{r^R} \times T_{i_j}^{(\mathrm{MLR})}\right)\right) \\ &\times \left(1 - \sum_{r=1}^{\infty} \beta_r^R \times \left(\frac{1}{r^{R-1}}\right)\right)^{-1}. \end{aligned}$$

Given the above, the minislot throughputs of the optimal contention tree, the 3-ary contention tree, and the MLR-based contention tree are $R/T_R^{(O)}$, $R/T_R^{(T)}$, and $R/T_R^{(\mathrm{MLR})}$, respectively, given R requests collided initially. The number of allocated contention minislots in the optimal scheme, the 3-ary scheme, and the SOMA scheme, respectively, are hence derived as

$$\begin{aligned} A_R^{(O)} &= T_R^{(O)} \\ A_R^{(T)} &= 1 + \sum_{r=1}^{\infty} \alpha_r^R \left(\sum_{j=1}^{r}\sum_{i_j=0}^{R} \binom{R}{i_j}\right. \\ &\qquad \left.\times \frac{(r-1)^{R-i_j}}{r^R} \times T_{i_j}^{(T)}\right) \end{aligned}$$

TABLE II
INITIAL AND COLLISION ALLOCATION MECHANISMS OF 8 SCHEMES

schemes	initial allocation	collision allocation	piggybacking
Optimal	exactly the number of new requests	exactly the number of retry requests	no
SOMA	the time proportional scheme	MLR table based	no
rSOMA	the time proportional scheme	MLR table based + relaxation	no
3-ary	the time proportional scheme	3-ary tree	no
Optimal-p	exactly the number of new requests	exactly the number of retry requests	yes
SOMA-p	the time proportional scheme	MLR table based	yes
rSOMA-p	the time proportional scheme	MLR table based + relaxation	yes
3-ary-p	the time proportional scheme	3-ary tree	yes

and

$$A_R^{(S)} = 1 + \sum_{r=1}^{\infty} \alpha_r^R \left(\sum_{j=1}^{r} \sum_{i_j=0}^{R} \binom{R}{i_j} \times \frac{(r-1)^{R-i_j}}{r^R} \times T_{i_j}^{(\mathrm{MLR})} \right). \tag{8}$$

Thus, the minislot throughputs for collision resolution in the optimal scheme, the 3-ary scheme, and the SOMA scheme are $(R - S_{R,I}^{(O)})/(A_R^{(O)} - 1 - A_{R,I}^{(O)})$, $(R - S_{R,I}^{(T)})/(A_R^{(T)} - 1 - A_{R,I}^{(T)})$, and $(R - S_{R,I}^{(S)})/(A_R^{(S)} - 1 - A_{R,I}^{(S)})$, respectively, given R requests participating in a contention cycle.

V. NUMERICAL RESULTS

A. Models

1) Network Model: This work compares many aspects of SOMA and relaxed SOMA with two allocation schemes. One of these schemes is the optimal one in which the headend *knows exactly* the number of requests and allocates that number of minislots. The other scheme is the 3-ary tree [14] in which the headend always allocates three minislots to every collided minislot. Table II summarizes the mechanisms in all schemes.

In the simulation, the upstream channel capacity is 6 Mbits/s and the minislot size is 8 bytes and 6.25 μs. The farthest station is 80 km away from the headend; thus, the longest round-trip delay is 800 μs, equivalent to a period of about 128 minislots. The length of a round is dynamically allocated and should be larger than the longest round-trip delay; 1000 stations are involved in the simulation.

2) Traffic Model: While the traditional traffic-generating models, like the Poisson packet arrival, are bursty over very limited timescales, real world traffic seems to be self-similar [15], [16], i.e., long-range dependent. Thus, we assume the packet interarrival time distribution is the Pareto distribution [16] with shape parameter α and location parameter $\beta(\alpha, \beta > 0)$, and with density and distribution functions

$$f(t) = \begin{cases} 0, & \text{if } t \le \beta \\ \dfrac{\alpha}{\beta}\left(\dfrac{\beta}{t}\right)^{\alpha+1}, & \text{if } t > \beta \end{cases}$$

TABLE III
PACKET SIZE DISTRIBUTION

Packet size (bytes)	64	128	256	512	1024	1518
Probability	0.6	0.06	0.04	0.02	0.25	0.03

Mean packet size = 368.1 bytes

and

$$F(t) = \begin{cases} 0, & \text{if } t \le \beta \\ 1 - \left(\dfrac{\beta}{t}\right)^{\alpha}, & \text{if } t > \beta \end{cases}$$

respectively, and a mean value

$$E[T] = \frac{\alpha\beta}{\alpha - 1} = \frac{1}{\lambda}, \qquad \text{if } \alpha > 1$$

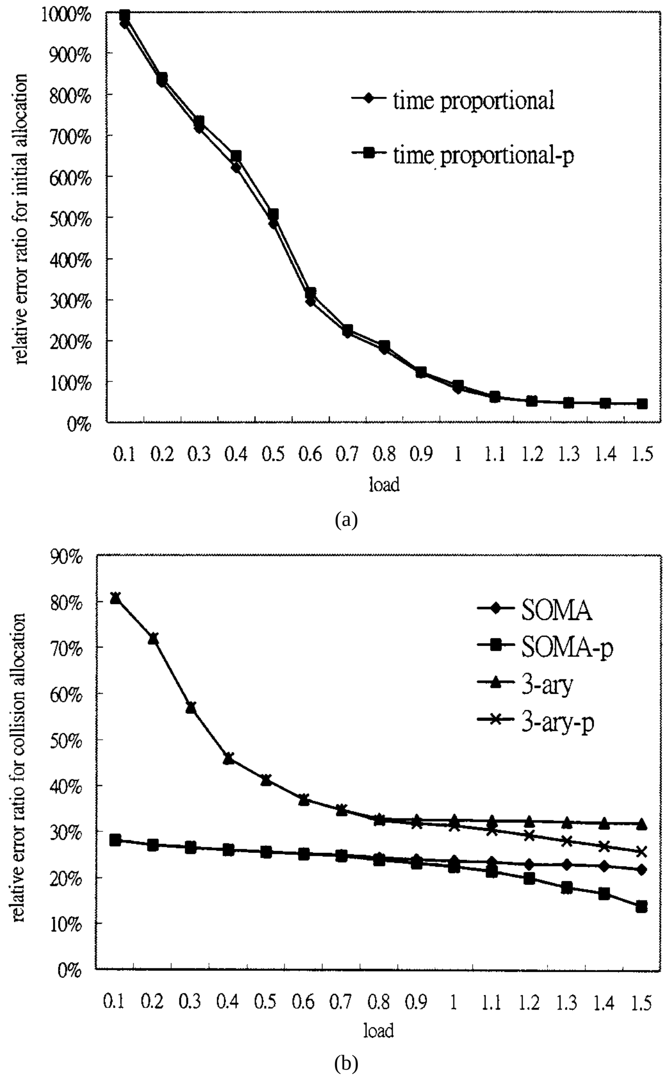
where λ is the packet arrival rate. If $\alpha \le 2$, then the distribution has infinite variance, and if $\alpha \le 1$, it has infinite mean and variance. The packet size distribution follows Table III [17]. For each station, the packet arrival rate λ is

$$\frac{6000000\,\mathrm{bits} \times \frac{1}{8} \times L}{1000 \times MPS}$$

where L is the offered load, and MPS is the mean packet size in bytes from Table III. The offered load is defined as the ratio of the offered traffic to the channel capacity. The burstiness of each traffic source is controlled by α which is set to 1.3 in the simulation. Thus, β can be derived as $((\alpha - 1)/\alpha\lambda)$. Consequently, given L, such traffic in each station is generated by the upper layers and delivered to the MAC sublayer transmission queue. Upon receiving packets, stations with scheduled data may piggyback the bandwidth requests; otherwise, these requests must arrive at the headend through the contention process. Moreover, the headend employees FCFS (first come–first serve) as bandwidth scheduling discipline for both the contending requests and the piggybacking requests in the simulation.

B. Simulation Results

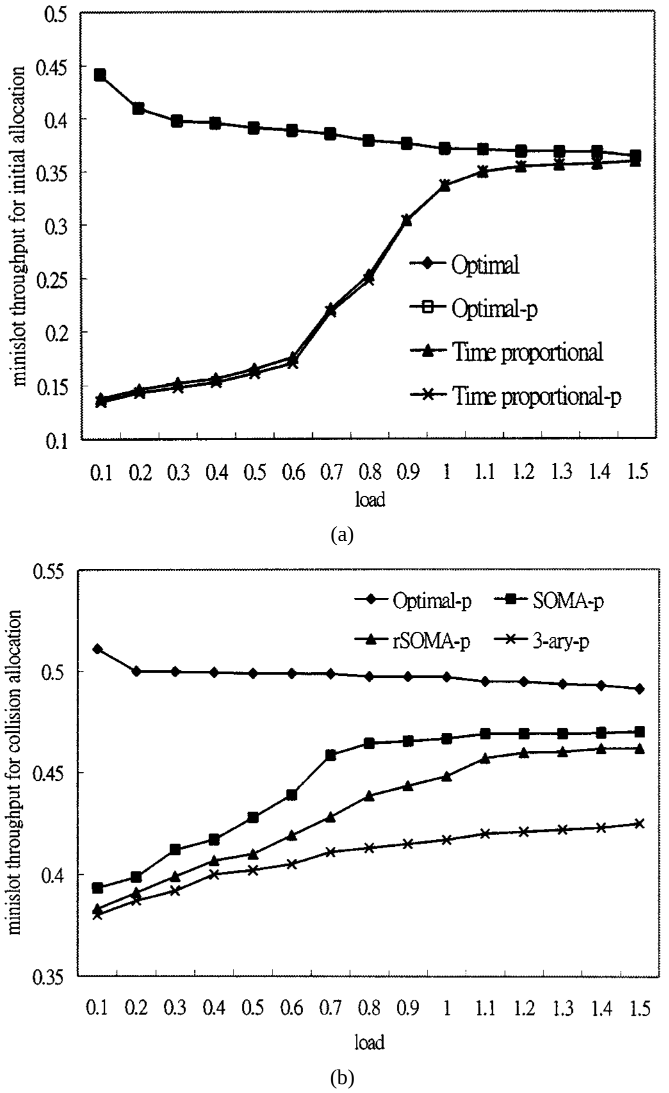
1) Relative Error Ratio for Schemes: Relative error ratio, defined as the ratio of the estimation error to the real number of



(a)

(b)

Fig. 8. Relative error ratio: (a) for initial allocation and (b) for collision allocation.



(a)

(b)

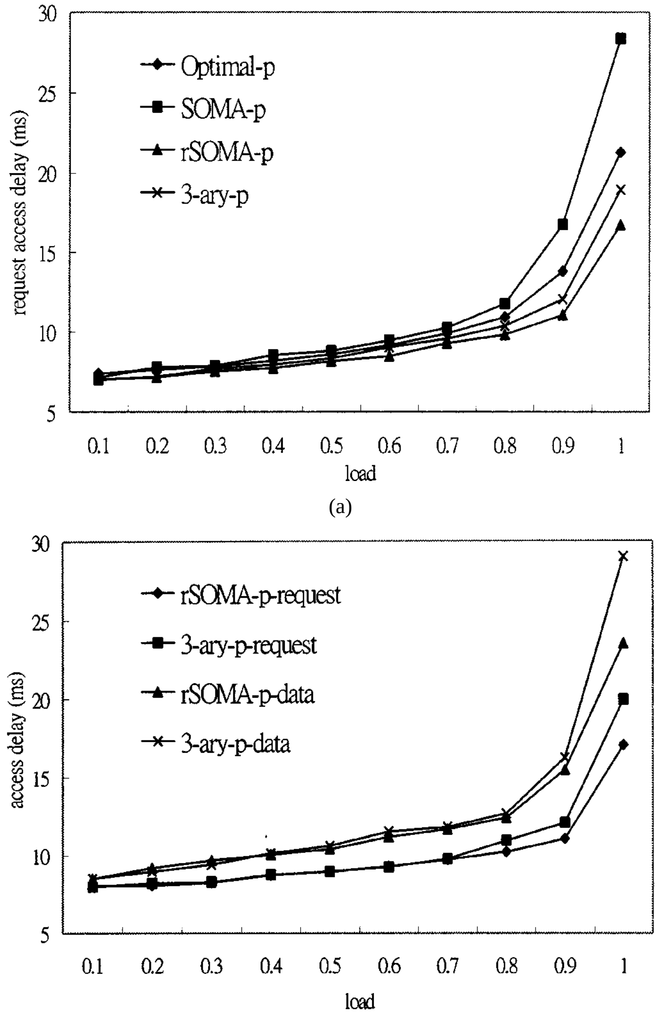
Fig. 9. Minislot throughput: (a) for initial allocation and (b) for collision allocation.

requests, is calculated to measure the accuracy of the proposed estimation schemes. Fig. 8(a) and (b) illustrates the relative error ratios for different schemes at initial contending phase and collision resolution phase, respectively. Fig. 8(a) depicts that a time proportional scheme makes poor estimations when the traffic is light, say 972.6% and 993.4% for the time proportional scheme and time proportional scheme with piggybacking, respectively, when the load is 0.1, due to fewer requests and high variance of the packet interarrival time. Since the MLR-based schemes measure the number of requests according to the pattern of contention results while the 3-ary scheme is not aware of it, the MLR-based schemes estimate more accurately than the 3-ary scheme does no matter how heavy the traffic load is. Fig. 8(b) demonstrates that the relative error ratios for the SOMA and the 3-ary schemes when the load is 1.5 are 22.03 and 31.90%, respectively.

2) Throughput: According to Fig. 9(a), the optimal scheme certainly outperforms the time proportional scheme. When the traffic is heavily loaded, the optimal scheme's throughput approaches 0.368, which is also the optimal throughput in slotted ALOHA. Since the time proportional scheme poorly estimates the number of requests at light traffic, it probably results in overallocation or underallocation—either one damages minislot throughput. Therefore, the gap between the optimal scheme and the time proportional scheme at light traffic is larger than that at heavy traffic. However, the time proportional scheme still achieves 0.358 when the load is 1.5.

According to Fig. 9(b), the optimal scheme achieves the optimal throughput 0.491 (see Section V-C) since it always allocates the right number of minislots for collision resolution. As expected, the SOMA scheme, say 0.470, approaches nearly optimal performance, and outperforms rSOMA scheme, say 0.461, when the traffic load is 1.5. Since the 3-ary scheme always allocates 3 minislots to resolve a collision, whereas the MLR-based schemes would make an accurate estimation and allocate proper number of minislots, both SOMA scheme and rSOMA scheme perform much better than the 3-ary scheme, say 0.425 when the load is 1.5, even though the time proportional scheme is inaccurate. In fact, MLR-based schemes can resolve collisions efficiently no matter how the initial estimation scheme is designed.

3) Access Delay and Contention Cycle: Request access delay refers to the interval between the packet arrival time and



(a)

(b)

Fig. 10. Access delay: (a) for request and (b) for data.

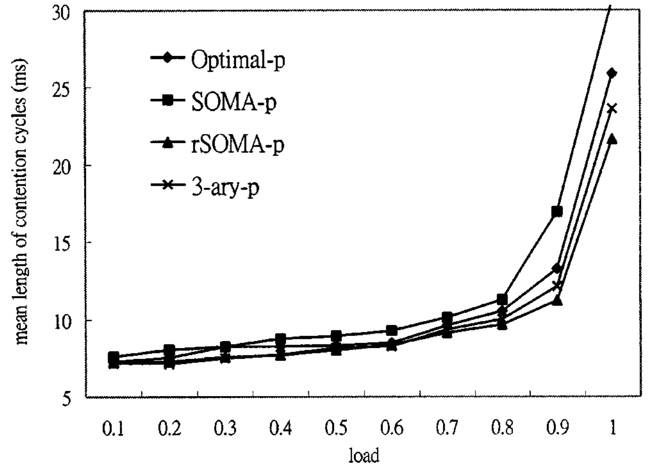


Fig. 11. Mean length of contention cycles.

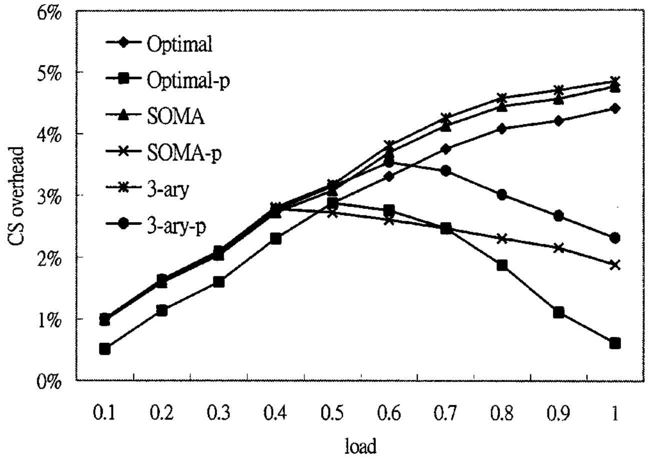


Fig. 12. CS overhead.

the time that the corresponding request is successfully received by the headend. Figs. 10(a) and 11 indicate that the rSOMA scheme is the winner in terms of the request access delay and the mean length of contention cycles. The request access delays of the rSOMA-p, the 3-ary-p, the optimal-p, and the SOMA schemes are 16.75, 18.95, 21.27, and 28.27 ms, respectively. Also, the mean lengths of contention cycles of the above four schemes are 21.65, 23.65, 25.88, and 30.53 ms, respectively. This is because when the number of collided requests is less than 3 or equal to 3, the rSOMA scheme resolves collisions as quickly as the 3-ary scheme does. Also, the performance of the 3-ary scheme degrades when the number of collided requests is larger than 3 while the rSOMA scheme resolves collision as efficiently as the SOMA does. Moreover, the SOMA scheme aims at maximizing the minislot throughput and it performs less efficiently than the optimal scheme does; thus, the SOMA scheme suffers higher request access delay. Since the data are scheduled after the corresponding requests are successfully received by the headend, and all schemes adopt the same packet size distribution, the behavior of data access delays for different schemes is, thus, similar to that of request access delays. From Fig. 10(b), the gap between request access delay and data access delay increases, from 0.47 ms (8.52–8.05 ms) at load 0.1 to 6.77 ms (23.52–16.75 ms) at load 1.0 for the rSOMA-p scheme, as the traffic load enlarges since more data have been scheduled.

4) CS Overhead: CS overhead is defined as the ratio of the mean number of CS's allocated in a round to the mean number of total minislots in a round. Fig. 12 depicts that the CS overhead is only 1.88% while adopting the SOMA scheme with piggybacking mechanism. Since the SOMA scheme has longer data access delay, it is more likely to append the request to the scheduled data, i.e., piggybacking. Consequently, the SOMA scheme starts piggybacking when the load is about 0.4, whereas the 3-ary scheme that has lower data access delay starts piggybacking when the load is about 0.6.

C. Analysis Results

Fig. 13 presents the theoretical optimum of minislot throughput, say 0.368, for initial allocation. Also, it indicates that the initial resolution estimator from the time proportional scheme is inaccurate when a fewer number of requests collided initially. Fig. 14(a) depicts that the minislot throughput for collision resolution of the MLR-based scheme approaches the optimal bound when the number of request is large, which confirms that the collision resolution estimator derived from the MLR-based scheme is more accurate than that from the 3-ary

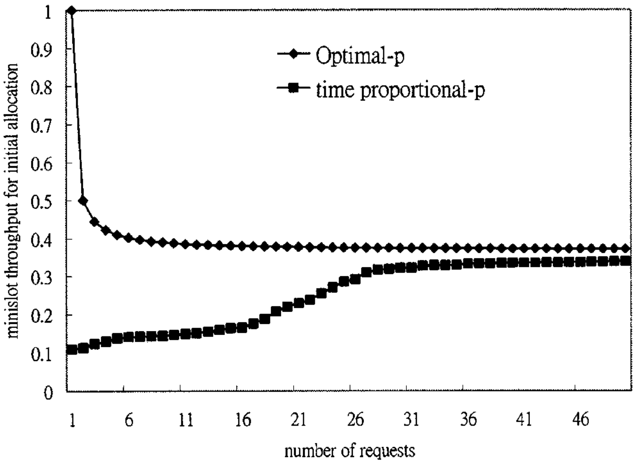
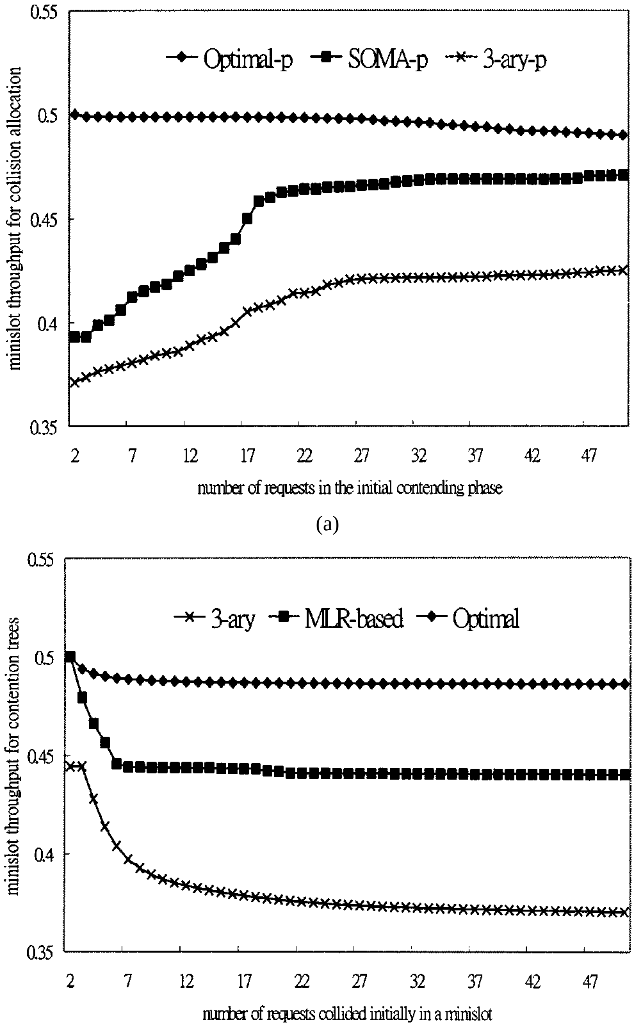


Fig. 13. Minislot throughput for initial resolution.

scheme. With a large number of requests, the minislot throughputs for the optimal scheme, the SOMA scheme, and the 3-ary scheme approach 0.491, 0.470, and 0.425, respectively. Based on Fig. 14(b), we can infer that when adopting the optimal scheme only a few requests, say 2 or 3 requests, are collided in a minislot in the initial contending phase; otherwise, the minislot throughput cannot achieve 0.491. Similarly, in most cases, there are 3, 4, or 5 requests collided in a minislot in the initial contending phase when adopting the time proportional scheme.

VI. Concluding Remarks

This paper has investigated minislot allocation in detail. The SOMA schemes are also proposed, which use the time proportional scheme for initial estimation and the statistical MLR table for collision estimation. The objective is to accurately estimate the number of requests and allocate minislots accordingly to optimize the minislot throughput. The paper has also performed simulations based on a self-similar traffic model to compare SOMA with other schemes in terms of minislot throughput, request access delay, data access delay, and CS overhead. The major findings of this comparison were as follows. First, the MLR-based schemes resolve collisions more efficient than the 3-ary scheme especially when the initial allocation is inaccurate. The inaccuracy is inevitable for the time proportional scheme when the request arrival process is self-similar. The 3-ary scheme is not robust enough to deal with bursty request arrivals, whereas the SOMA scheme can still accommodate the situation. Second, the MLR-based schemes optimize minislot throughput and outperform the 3-ary scheme. This paper also analyzes (also based on a self-similar traffic model) the minislot throughput of the time proportional scheme, which indicates that the time proportional scheme makes poor estimations at light traffic. The analysis also depicts that the time proportional scheme gives rise to worse collision multiplicity compared to the optimal scheme. Meanwhile, this paper analyzes the minislot throughputs for collision resolution of the optimal scheme, the SOMA scheme, and the 3-ary scheme to be 0.491, 0.470, and 0.425, respectively. However, despite the success of the current investigation, there is still room to design a more accurate and robust initial estimation scheme that could further reduce the estimation error.



(a)

(b)

Fig. 14. Minislot throughput: (a) for collision allocation and (b) for various contention trees.

References

[1] IEEE 802.14 Working Group, “Media access control,” IEEE draft std. 802.14 draft 3 r3, Oct. 1998.

[2] Y. D. Lin. (4th quarter, 1998) On IEEE 802.14 medium access control protocol. *IEEE Commun. Surveys* [Online]. Available: http://www.comsoc.org/pubs/surveys.

[3] Cable Television Laboratories Inc., “Data-over-cable service interface specifications—Radio frequency interface specification version 1.1,”, July 1999.

[4] N. Golmie, Y. Saintillan, and D. Su. (1st quarter, 1999) A review of contention resolution algorithms for IEEE 802.14 networks. *IEEE Commun. Surveys* [Online]. Available: http://www.comsoc.org/pubs/surveys

[5] J. Mosely and P. A. Humblet, “A class of efficient contention resolution algorithms for multiple-access channels,” *IEEE Trans. Commun.*, vol. COM-33, pp. 145–151, Feb. 1985.

[6] R. Citta, J. Xia, and D. Lin, “The Tree-Based Algorithm with Soft Blocking,” *IEEE 802.14 Working Group*, Nov. 1996. contribution IEEE802.14-96/244.

[7] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1992, pp. 271–273.

[8] L. Klienrock and F. A. Tobagi, “Packet switching in radio channels: Part 1: CSMA modes and their throughput-delay characteristics,” *IEEE Trans. Commun.*, vol. COM-23, 1975.

[9] L. Merakos and C. Bisdikian, "Delay analysis of the n-ary stack random-access algorithm," *IEEE Trans. Inform. Theory*, vol. 34, pp. 931–942, Sept. 1988.

[10] P. Jacquet, P. Muhlethalerm, and P. Robert, "Performant implementations of tree collision resolution on CATV networks," *IEEE 802.14 Working Group*, Apr. 1996. contribution IEEE802.14-96/115.

[11] F. Abi-Nassif and W. C. Lee, "Offered load estimation in a multimedia cable network system," in *Proc. ICC'99*, Vancouver, Canada, June 1999.

[12] K. Sriram, "Performance of MAC protocols for broadband HFC and wireless access networks," *Advances Perform. Anal.*, vol. 1, no. 1, pp. 1–37, 1998.

[13] 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, pp. 427–435, Dec. 1998.

[14] P. Mathys and P. Flajolet, "Q-ary collision resolution algorithms in random-access systems with free and blocked channel access," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 217–243, Mar. 1985.

[15] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Self-similarity through high variability: Statistical analysis of ethernet LAN traffic at the source level," *IEEE/ACM Trans. Networking*, vol. 5, pp. 71–86, Feb. 1997.

[16] V. Paxson and S. Floyd, "Wide area traffic: The failure of Poisson modeling," *IEEE/ACM Trans. Networking*, vol. 3, pp. 226–224, June 1995.

[17] J. Limb, "Performance evaluation process for MAC protocols," *IEEE 802.14 Working Group Meet.*, May 1996. contract IEEE802.14-96/083R2.

Wei-Ming Yin received the B.A. and M.S. degrees in computer and information science from National Chiao Tung University in 1995 and 1997, respectively. He is currently a Ph.D. candidate in the same department. His research interests include protocol design and analysis of residential networks, and bandwidth scheduling over QoS guaranteed networks.

Ying-Dar Lin (M'95) received the Bachelor's degree in computer science and information engineering from National Taiwan University in 1988, and the M.S. and Ph.D. degrees in computer science from the University of California, Los Angeles, in 1990 and 1993, respectively.

At the UCLA Computer Science Department, he worked as a Research Assistant from 1989 to 1993, and as a Teaching Assistant from 1991 to 1992. In the summers of 1987 and 1991, he was a Technical Staff Member in IBM Taiwan and Bell Communications Research, respectively. He joined the Faculty of the Department of Computer and Information Science at National Chiao Tung University in August 1993, and has been a Professor since 1999. His research interests include design, analysis, and implementation of network protocols and algorithms, wire-speed switching and routing, quality of services, and intranet services. He has authored two books. He can be reached at http://www.cis.nctu.edu.tw/~ydlin or ydlin@cis.nctu.edu.tw.

Dr. Lin is a member of ACM.