Short Paper

A Fast Optimal Non-Continuous Slot Reuse Scheme for CRMA High-Speed Networks*

PI-RONG SHEU AND CHARNG-MAW LIN
Department of Electrical Engineering
National Yunlin University of Science and Technology
Yunlin, Taiwan 640, R.O.C.
E-mail: sheupr@flame.yuntech.edu.tw

Cyclic-Reservation Multiple-Access (CRMA) is an effective access scheme for high-speed networks. In CRMA, the headend periodically generates *reserve* commands. If necessary, each station may reserve a number of empty slots in each *reserve* command. For each *reserve* command, the headend generates a cycle of length equal to the total number of empty slots which are reserved to serve the reservations. Generally speaking, a longer cycle length means a longer access delay and a lower throughput. Therefore, it is desirable to study how to reduce the cycle length. However, it has been shown that the problem is NP-complete if all the empty slots used by a station in a cycle are required to be consecutive [1]. In this paper, we remove the slot-contiguity constraint and propose a fast, optimal non-continuous slot reuse scheme with low time complexity $O(M^2)$, where M denotes the number of stations. Experimental results demonstrate that our new scheme has much shorter cycle lengths, much higher throughput, and much shorter MAC (Medium Access Control) delay than the original CRMA scheme.

Keywords: cyclic-reservation multiple-access, dual-bus network, high-speed network, optimal algorithm, slot reuse

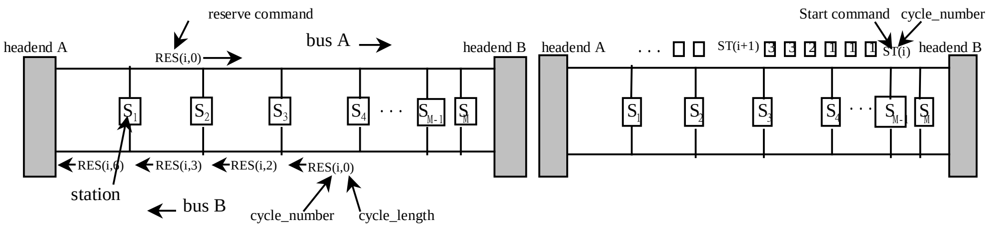
1. INTRODUCTION

In the future high-speed local and metropolitan area networks will provide capacities beyond one gigabit per second and a geographical coverage of hundreds of kilometers [2-5]. To support such networks, several high-speed and cost effective access schemes have been proposed, such as Distributed Queue Dual Bus (DQDB) [2] and *Cyclic-Reservation Multiple-Access* (CRMA) [3-5]. DQDB is the IEEE 802.6 standard. It can achieve a total throughput approaching the bus capacity regardless of network size or speed, and offers minimum access delay by constraining each station to at most one pending slot reservation. However, at high speeds or over long distances, and under the

Received March 30, 1999; revised November 24, 1999 & March 8, 2000; accepted June 14, 2000.
Communicated by Jean-Lien C. Wu.
* This work was supported by the National Science Council of the Republic of China under Grant NSC 87-2213-E-224-002.

requirement of high bandwidth utilization, this constraint is likely to suffer major throughput unfairness [6]. On the other hand, it has been shown [3-5] that CRMA is able to offer high performance even at high speeds and long distances through two novel mechanisms: cyclic-reservation_access and reservation-cancellation backpressure. For meeting any set of fairness requirements, the cyclic-reservation access provides throughput efficiency and flexibility in capacity allocation. The reservation-cancellation backpressure minimizes the worst-case access delay. Therefore, we are mainly concerned with CRMA.

As shown in Fig. 1(a), a dual-bus CRMA network consists of two unidirectional buses running in opposite directions. One bus is referred to as bus A, the other as bus B. The headends provide the global access functions of CRMA. Without loss of generality, only the operation of bus A is described in this paper; the bus B case is completely symmetric. In the original CRMA reservation scheme [4], headend A periodically issues *reserve* commands on bus A. Each *reserve* command contains a *cycle_number* and a *cycle_length*, the value of the latter being initially set to zero. As a *reserve* command travels along bus A, the stations do not modify the value of its *cycle_length.* After the *reserve* command reaches headend B, it is retransmitted back to headend A via bus B. From this point, each station starts to add the number of empty slots it requires in this cycle to the *cycle_length*. Finally, when the *reserve* command arrives at headend A, both the *cycle_number* and *cycle_length* are stored in a reservation FIFO (First In First Out) queue as the total number of empty slots requested by all the stations in this cycle. Each time a (*cycle_number*, *cycle_length*) reaches the head of the reservation FIFO queue, headend A generates a new cycle by issuing a *start* command with the *cycle_number* followed by as many empty slots as indicated by the *cycle_length*. When the *start* command travels along bus A, each station waits for the first empty slot and subsequently transmits its packets into as many empty slots as it has reserved for the cycle. To illustrate this point, an example is shown in Fig. 1. First, in Fig. 1(a), stations S_1, S_2, and S_3 reserve 3, 1 and 2 empty slots in cycle *i*, respectively. Next, in Fig. 1(b), when this cycle is serving, stations S_1, S_2, and S_3 will use the reserved consecutive empty slots, beginning with the first empty slot which they meet respectively.



(a) A CRMA high-speed network and its reserve command.

(b) A CRMA high-speed network and its start command.

Fig. 1. A CRMA high-speed network and its reservation scheme.

In the above CRMA reservation scheme, there exists an undesirable feature: once a slot is used, it propagates to the end of the bus even though its packet might have already been read by its destination [1, 7-10]. Consequently, the length of a cycle is always equal to the total number of empty slots reserved by all the stations in the cycle. On the other hand, the phenomenon of traffic locality often occurs in present-day networks, including metropolitan area networks [11-12]. By exploiting the property of traffic locality, several methods based on the concept of *slot reuse* have been proposed to improve the performance of the original CRMA reservation scheme [1, 7-10]. In these slot reuse methods, a slot whose packet has passed its destination is released and becomes an empty slot for further use. Thus, a shorter cycle length can be obtained. In general, a shorter cycle length means better slot utilization as well as shorter access delay. In this paper, we will study the slot reuse problem, or equivalently, the problem of reducing the total number of empty slots generated every cycle.

One of the main slot reuse methods, called the *cycle compression problem (the CCP)*, has been extensively studied in [1, 7-8]. In the CCP, multiple packets are allowed to be served by a common empty slot if any two of them are not *intersected*. A packet (i, j) from station S_i to station S_j is said to be *intersected* with another packet (i', j') if $i \leq i' < j$. However, it has been shown that the CCP is NP-complete [1]. In [13], it was found that the main factor causing the CCP to be NP-complete is the constraint that all the empty slots used by a station in a cycle are required to be consecutive. In fact, if this slot-contiguity constraint is removed, the problem of obtaining the minimal cycle length can then be optimally solved in polynomial time. Based on the interval graph coloring algorithm [14], [13] proposed an optimal time slot assignment algorithm called algorithm OTSA, whose time complexity is $O\left(\left(\sum_{i=1}^{M}\alpha_i\right)^2\right)$, where α_i denotes the number of empty slots reserved by station S_i in the cycle and M is the total number of stations. Algorithm OTSA is centralized and is executed in the headend. To find the empty slots assigned for each station within a cycle, the headend will first collect the reservation requests of each station via the *reserve* command. The headend then uses algorithm OTSA to compute an optimal time slot assignment. Finally the assignment is sent back to each station. When the cycle starts to be served, each station transmits its packets to the empty slots according to the assignment.

CRMA is designed mainly for high-speed networks. Although algorithm OTSA is optimal in terms of cycle lengths, its high time complexity makes it impractical to implement. In view of this defect, we propose a new optimal non-continuous slot reuse scheme with lower time complexity. The time complexity depends solely on the total number of stations, and is independent of the total number of slots reserved. Furthermore, our new scheme is operated in a more distributed fashion. In high-speed networks like CRMA, a distributed algorithm is more fault-tolerant and can be realized at a faster speed than a centralized one. Experimental results indicate that our new scheme possesses much shorter cycle lengths, much higher throughput, and much shorter MAC delay than the original CRMA scheme. Hence, our new scheme has a number of advantages over previous methods.

The rest of this paper is organized as follows. In Section 2, some notations and definitions are introduced. In Section 3, a fast optimal non-continuous slot reuse scheme is proposed and its time complexity is analyzed. In Section 4, the simulation models and results are reported. Finally in Section 5, some concluding remarks are given.

2. PROBLEM FORMULATION

Throughout this paper, we assume that the CRMA network, shown in Fig. 1(a), consists of M stations and two unidirectional buses, bus A and bus B. We define a *reserve vector* $x = [i, j)$ as a request of one empty slot between source S_i and destination S_j. Note that $x = [i, j)$ means x includes point i but excludes point j. The reason for adopting such a definition will become clear later. Two reserve vectors, $x = [i, j)$ and $x' = [i', j')$ are said to be *intersected* if $i \le i' < j$. It is clear that a slot used by a reserve vector cannot be reused by another reserve vector if they are intersected. A reserve of α empty slots issued by a certain station is represented by the same α reserve vectors.

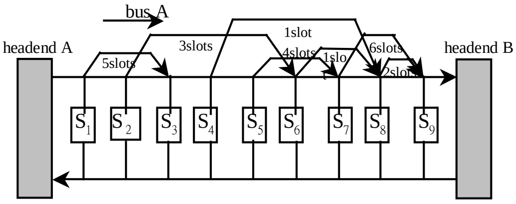
In a cycle, if station S_i reserves α_i empty slots, we use a *reserve set*, $X = \{x_1, x_2, \cdots, x_{\alpha_1}, x_{\alpha_1+1}, x_{\alpha_1+2}, \cdots, x_{\alpha_1+\alpha_2}, x_{\alpha_1+\alpha_2+1}, \cdots, x_{\alpha_1+\alpha_2+\cdots+\alpha_M}\}$, where x_i is a reserve vector, to represent all the reserves made by all the stations. (For convenience, in this paper we allow a set to include the same elements.) In addition, an interval representation of a reserve set X is helpful in understanding our slot reuse scheme. Fig. 2(b) shows a reserve set X representing the CRMA reservation pattern in Fig. 2(a), and Fig. 2(c) shows its corresponding interval representation. For a given reserve set X, we define the number of reserve vectors intersected at station S_i (denoted by $I(X, S_i)$) to be the number of reserve vectors that contain point i. We also define $I_{max}(X) = \underset{1 \le i \le M}{MAX}\{I(X, S_i)\}$. Fig. 2(c) gives the values of $I(X, S_i)$, for $1 \le i \le 9$, and $I_{max}(X)$ for the X in Fig. 2(b).

We define the *length* of the cycle associated with a given reserve set X to be the number of the empty slots which headend A must generate to serve X. A cycle for a given reserve set X is said to be *optimal* if its length is the minimum among all possible cycles for X. A non-continuous slot reuse scheme is said to be *optimal* if it can always generate an optimal cycle for any given reserve set under the condition that the empty slots used by a station in a cycle are not required to be consecutive.

3. A FAST OPTIMAL NON-CONTINUOUS SLOT REUSE SCHEME

In this section, we present a fast optimal non-continuous slot reuse scheme with time complexity $O(M^2)$ for CRMA high-speed networks.

In our new non-continuous slot reuse scheme, each *reserve* command is an array of length $2M+1$, whose first entry is a *cycle_number* and other entries are initially set to zero. Each station S_k puts the destination address of its packets and the number of required empty slots into the $(2k)$th and $(2k + 1)$th entries, respectively. When the *reserve* command arrives at headend A, headend A calculates $I_{max}(X)$ and stores (*cycle_number*, $I_{max}(X)$) in a reservation FIFO queue as the total number of empty slots which headend A must generate in this cycle. Each time a (*cycle_number*, $I_{max}(X)$) reaches the head of the reservation FIFO queue, headend A generates a new cycle by issuing a *start* command with the *cycle_number* followed by $I_{max}(X)$ empty slots. When the *start* command travels along bus A, each station S_k sequentially checks each slot and performs the following three actions:



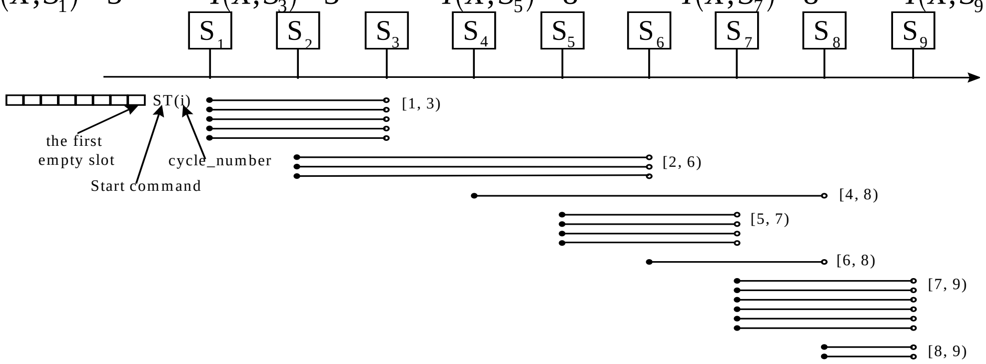
(a) A CMRA reservation pattern R.

$$X = \left\{ \begin{array}{l} [1,3), [1,3), [1,3), [1,3), [1,3), \\ [2,6), [2,6), [2,6), \\ [4,8), \\ [5,7), [5,7), [5,7), [5,7), \\ [6,8), \\ [7,9), [7,9), [7,9), [7,9), [7,9), [7,9), \\ [8,9), [8,9) \end{array} \right\}$$

(b) The reserve set X corresponding to the reservation pattern R in Fig. 2(a).

$$I_{\max}(X) = \underset{1 \le i \le 9}{MAX}\{I(X, S_i)\} = 8$$

$$I(X,S_1)=5 \quad I(X,S_2)=8 \quad I(X,S_3)=3 \quad I(X,S_4)=4 \quad I(X,S_5)=8 \quad I(X,S_6)=6 \quad I(X,S_7)=8 \quad I(X,S_8)=8 \quad I(X,S_9)=0$$



(c) The interval representation of the reservation set X in Fig. 2(b).

Fig. 2. An illustration of our problem formulation.

Action 1: If a slot carries a packet belonging to S_k, S_k copies the packet and immediately releases the slot so that it becomes empty.

Action 2: If a slot is empty (the empty slot may be the one that S_k just released in action 1) and S_k has packets to send, S_k uses the empty slot to immediately transmit its one packet.

Action 3: Otherwise, S_k ignores the slot.

In the following we show that our scheme is optimal and has low time complexity. Lemma 1 is complicated and can be better understood with the help of the examples shown in Figs. 2(c), 3, 4 and 5.

Lemma 1: Given a reserve set X, let X_1 denote the set of remaining reserve vectors after the first slot generated by headend A passes through all the stations. If our scheme is applied, then the value of $I(X_1, S_t)$ of each station S_t with $I(X, S_t) = I_{max}(X)$ will be $I(X, S_t) - 1$. This also implies that $I_{max}(X_1) = I_{max}(X) - 1$.

Proof: There are two possible cases when a station S_k sees the first slot.

Case 1: The first slot carries a reserve vector whose destination is located after station S_k.

In this case, $I(X_1, S_k) = I(X, S_k) - 1$ because the reserve vector which is being carried includes point k. In the example shown in Fig. 2(c), stations S_2, S_5, S_6, and S_7 will experience Case 1.

Case 2: The first slot is an empty slot. (This implies either that the first slot has not yet carried a reserve vector (situation 2a) or that the slot carried a reserve vector whose destination is located before S_k (situation 2b) or at S_k (situation 2c)).

The reason why the first slots in situations 2b and 2c are empty is due to action 1. In Fig. 2(c), station S_1 will experience situation 2a. Station S_4 will experience situation 2b. Stations S_3, S_8 and S_9 will experience situation 2c.

For Case 2, we have two subcases:

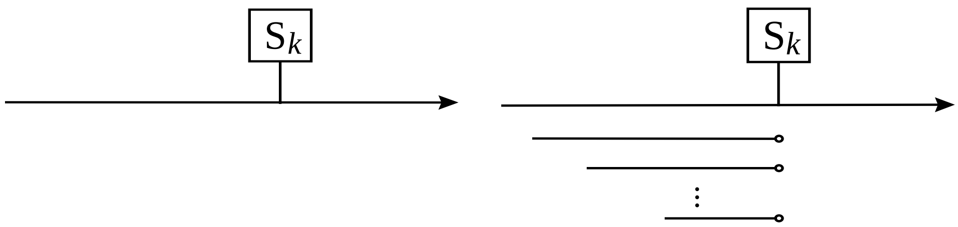
Case 2′**:** Station S_k has reserve vectors to send.

According to action 2, S_k must put its one reserve vector into the first slot. Therefore, $I(X_1, S_k) = I(X, S_k) - 1$. In Fig. 2(c), stations S_1, S_4, and S_8 will experience case 2′.

Case 2″**:** Station S_k has no reserve vectors to send.

According to action 3, S_k ignores the slot and $I(X_1, S_k) = I(X, S_k)$. However, in such a situation, $I(X, S_k) < I_{max}(X)$. Let us consider the following two possible situations.

Case 2″**-I**: There are no reserve vectors that contain point k (see Figs. 3(a) & 3(b)).



(a) No reserve vectors. (b) No reserve vectors containing point k.

Fig. 3. Case 2″-I of the proof of Lemma 1.

In this subcase, $I(X, S_k) = 0$. In Fig. 2, station S_9 will experience case 2″-I.

Case 2″**-II**: There are reserve vectors that contain point k (see Fig. 4).

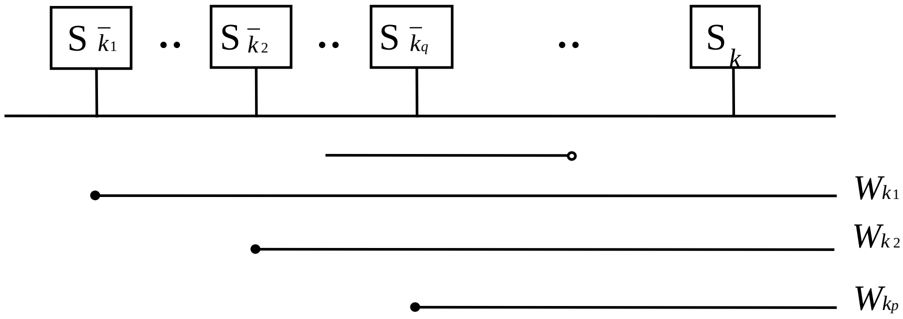


Fig. 4. Case 2″-II of the proof of Lemma 1.

Let these reserve vectors that contain point k be $w_{k_1}, w_{k_2}, \cdots, w_{k_p}$, where w_{k_j}, for $1 \le j \le P$, denotes a reserve vector. Let $S_{\bar{k}_1}, S_{\bar{k}_2}, \cdots, S_{\bar{k}_q}$ be the source stations of $w_{k_1}, w_{k_2}, \cdots, w_{k_p}$. Let $S_{\bar{k}_q}$ be the station that is nearest to S_k among $S_{\bar{k}_1}, S_{\bar{k}_2}, \cdots, S_{\bar{k}_q}$. Recall that in the current case (case 2), S_k does not see the first slot carrying a reserve vector. Therefore, $S_{\bar{k}_q}$ must see that the first slot carries a reserve vector with destination between $S_{\bar{k}_q}$ and S_k (see Fig. 4). Thus, $I(X, S_{\bar{k}_q}) \ge I(X, S_k) + 1$, and consequently, $I(X, S_k) < I_{max}(X)$.

In summary, we conclude that the value of $I(X_1, S_t)$ of each station S_t with $I(X, S_t) = I_{max}(X)$ will be $I(X, S_t) - 1$ after the first slot generated by headend A goes through all the stations. Obviously, by the definition of $I_{max}(X)$, this implies that $I_{max}(X_1) = I_{max}(X) - 1$. As an example, Fig. 5 shows the results after the first slot goes through all the stations when our scheme is applied to Fig. 2(c). ■

Lemma 2: Given a reserve set X, if our scheme is applied, then exactly $I_{max}(X)$ empty slots must be generated to serve X.

Proof: After the first slot generated by headend A goes through all the stations, some reserve vectors in X will have been served. We denote the set of remaining reserve vectors by X_1. According to Lemma 1, $I_{max}(X_1) = I_{max}(X) - 1$. Similarly, let X_2 denote the set of remaining reserve vectors after the second slot goes through all the stations. According to Lemma 1, $I_{max}(X_2) = I_{max}(X_1) - 1$. Apparently, after the $I_{max}(X)$th slot goes through all the stations, $I_{max}(X_{Imax(X)}) = 0$, which means all the reserve vectors in X have been served. ■

Theorem 1: Given a reserve set X, our scheme can optimally serve X in $O(M^2)$ time.

$$I_{\max}(X_1) = \underset{1 \le i \le 9}{MAX}\{I(X_1, S_i)\} = 7 = I_{\max}(X) - 1$$

$$I(X_1, S_2) = 7 \quad I(X_1, S_4) = 3 \quad I(X_1, S_6) = 5 \quad I(X_1, S_8) = 7$$

$$I(X_1, S_1) = 4 \quad I(X_1, S_3) = 3 \quad I(X_1, S_5) = 7 \quad I(X_1, S_7) = 7 \quad I(X_1, S_9) = 0$$

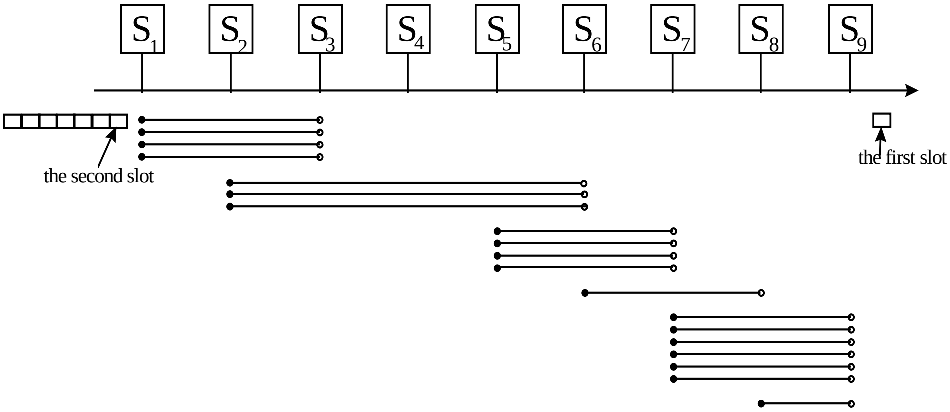


Fig. 5. The results after the first slot goes through all the stations when our scheme is applied to Fig. 2(c).

Proof: It is obvious that at least $I_{max}(X)$ empty slots must be generated to serve X. From Lemma 2, we know that if our scheme is applied, then exactly $I_{max}(X)$ empty slots must be generated to serve X. Therefore, our scheme is optimal.

Next, consider the time complexity of our scheme. When headend A receives a *reserve* command, it first sets $I(X, S_k) = 0$, for $1 \le k \le M$, and then sequentially checks each pair of entries, $(j, j + 1)$, for $j = 2, 4, 6, \ldots, 2M$. If the (j)th and $(j + 1)$th entries are d and α, respectively, then $I(X, S_{\frac{j}{2}}) = I(X, S_{\frac{j}{2}}) + \alpha$, …, $I(X, S_{d-1}) = I(X, S_{d-1}) + \alpha$ (this requires $O(M^2)$ time in the worst case). Finally, from $I(X, S_k)$, for $1 \le k \le M$, headend A can calculate $I_{max}(X)$ in $O(M)$ time. As a result, the time complexity of our scheme is $O(M^2)$. ■

4. COMPUTER SIMULATIONS

In this section, we will compare our non-continuous slot reuse scheme with the original CRMA scheme in terms of average cycle length, average throughput, and average MAC delay.

As to the probability $P_{i,j}$ that a station S_i issues a reservation to a destination S_j, two different traffic models are considered in the simulations. They are similar to those of [1, 7-8, 10, 13]. We first consider the uniform distribution traffic model in which $P_{i,j} = 2/(M \times (M - 1))$ if $i < j$ and zero otherwise. Then, the phenomenon of traffic locality is considered in the second model. That is,

$$P_{i,j} = \begin{cases} 0 & i = j \\ \dfrac{(1-p_1)^{|j-i|-1} \times p_1}{2-(1-p_1)^{M-i}-(1-p_1)^{i-1}} & i \neq j \end{cases}.$$

This equation represents a normalized geometric distribution where p_1 denotes the level of locality of the traffic [10]. That is, a larger p_1 value denotes the situation where each station has more packets for its nearby stations than for far-away stations. Consequently, slots can be reused many times more.

In our simulations the cycle length l_i of cycle i is defined to be the total number of empty slots generated by headend A in cycle i. The total number of empty slots reserved by all the stations in cycle i is denoted as TN_i. The MAC delay of packet j at station S_i is denoted by MD_{ij}. More precisely, MD_{ij} is the time (measured in "time slots") starting from when packet j is confirmed by headend A and ending when it is served. Thus, if the number of cycles is n, then the average cycle length ACL = $\left(\sum_{i:=1}^{n} l_i\right) \Big/ n$. The average throughput AT = $\left(\sum_{i:=1}^{n} (TN_i) \big/ l_i\right) \Big/ n$. The average MAC delay AMD = $\left(\sum_{i:=1}^{M}\sum_{j:=1}^{\alpha_i} MD_{ij}\right) \Big/ \sum_{i:=1}^{M} \alpha_i$.

In both simulation models, the following three cases are considered: (1) Two different network sizes are studied, namely, M = 20 and M = 30. (2) The total arrival rate λ of reservations made in a cycle is investigated from 0.2 to 1.0 with a step of 0.2. (3) The number of empty slots issued in each reservation is obtained from the following uniform distributions: one with a mean EB of 3, and the other with a mean EB of 6. Due to limited space, only the results for M = 20 and EB = 3 are presented.

Figs. 6 and 7 indicate that our scheme has a shorter cycle length than the original CRMA scheme for both models. Especially for larger values of p_1 our scheme produces much shorter average cycle lengths. This demonstrates that the phenomena of traffic locality can be exploited to achieve better slot reuse. Figs. 8 and 9 show that our scheme offers higher throughput. It is interesting to note that a larger network results in a greater improvement in throughput. This is because used slots can be reused by more nodes in a larger network. Finally, Figs. 10 and 11 present the simulation results on the average MAC access delay. Undoubtedly, our scheme makes a considerable improvement on the original CRMA scheme, especially when the total arrival rate is large. This is because a shorter cycle length will cause a packet to be served more quickly.

5. CONCLUSIONS

In this paper, we have discussed the problem of cycle length reduction in CRMA high- speed networks. Based on the assumption that the empty slots used by a station within a given cycle may not be consecutive, we have constructed a fast optimal non-continuous slot reuse scheme with time complexity $O(M^2)$, where M is the number of stations, to solve the problem. Compared with [13], the time complexity of our new

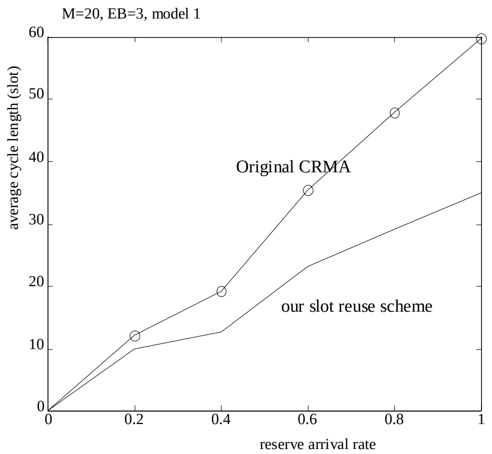


Fig. 6. Simulation results on cycle lengths for model 1.

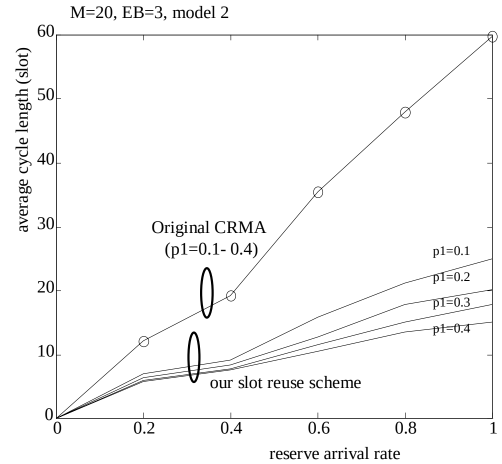


Fig. 7. Simulation results on cycle lengths for model 2.

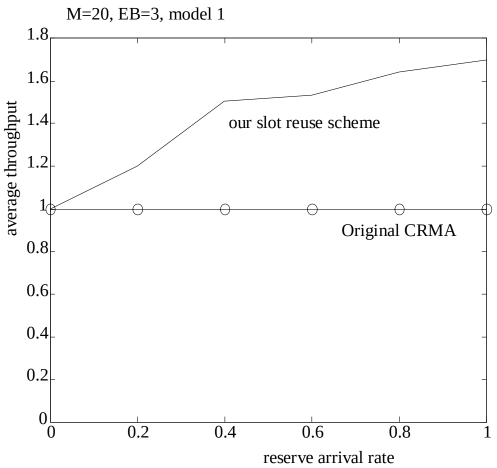


Fig. 8. Simulation results on throughput for model 1.

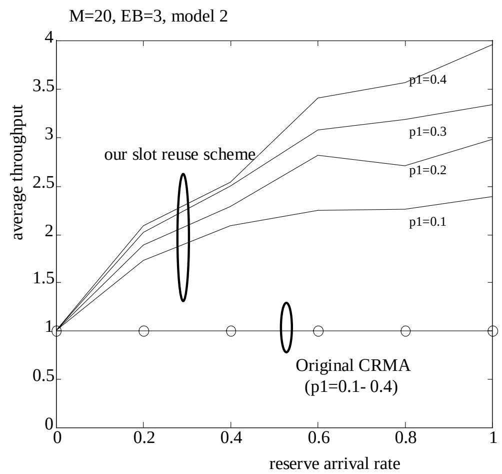


Fig. 9. Simulation results on throughput for model 2.

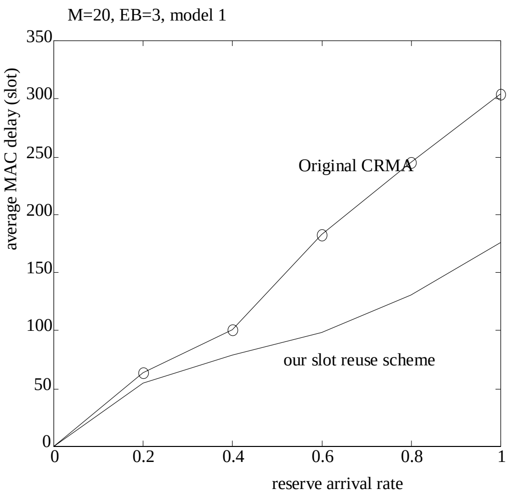


Fig. 10. Simulation results on MAC access delays for model 1.

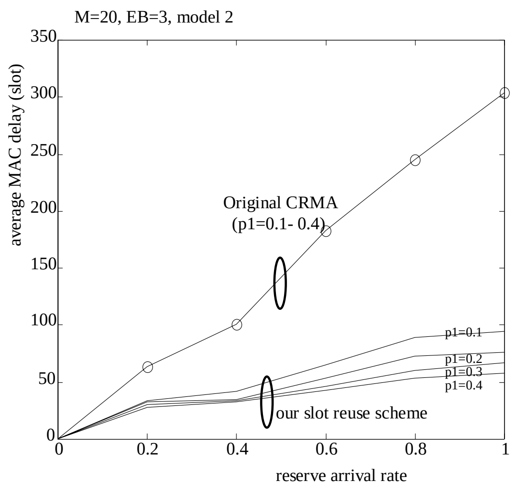


Fig. 11. Simulation results on MAC access delays for model 2.

scheme is lower. It depends only on the total number of stations, and is independent of the total number of slots reserved. Furthermore, our new scheme is performed in a distributed fashion rather than in a centralized manner as in [13], thereby improving both speed and reliability. The experimental results also show that our scheme has much shorter cycle lengths, much higher throughput, and much shorter MAC delay.

REFERENCES

1. N. F. Huang, C. S. Chang, and G. K. Ma, "A cycle compression algorithm for CRMA high-speed networks," *IEEE ICC'93*, 1993, pp. 1363-1367.
2. IEEE Project 802.6, Proposed Standard, "Distributed queue dual bus metropolitan area network," Doc. P802.6/D12, 1990.
3. H. R. Muller, M. M. Nassehi, J. W. Wang, E. Zurfluh, W. Bux, and P. Zafiropulo, "DQMA and CRMA: new access schemes for Gbit/s LANs and MANs," in *Proceedings of IEEE Infocom'90*, 1990, pp. 185-191.
4. M. M. Nassehi, "CRMA: an access scheme for high-speed LANs and MANs," in *Proceedings of SUPERCOMM/ICC '90*, 1990, pp. 1697-1702.
5. M. M. Nassehi, "Cyclic-reservation multiple-access scheme for Gbit/s LANs and MANs based on dual bus configuration," in *Proceedings of EFOC/LAN '90*, 1990, pp. 246-251.
6. J. W. Wong, "Throughput of DQDB networks under heavy load," in *Proceedings of EFOC/LAN '89*, 1989, pp. 146-151.
7. N. F. Huang and C. C. Chiou, "An efficient slot-reuse scheme for CRMA high-speed networks," *Journal of High Speed Networks* Vol. 2, No. 3, 1993, pp. 259-286.
8. N. F. Huang and C. C. Chiou, "A new reservation scheme for CRMA high-speed networks," *Computer Communications*, Vol. 17, No. 6, 1994, pp. 396-406.
9. O. Sharon and A. Segall, "A simple scheme for slot reuse without latency for a dual bus configuration," *IEEE/ACM Transactions on Networking*, Vol. l, No. 1, 1993, pp. 96-104.
10. O. Sharon and A. Segall, "On the efficiency of slot reuse in the dual bus configuration," *IEEE/ACM Transactions on Networking*, Vol. 2, No. 2, 1994, pp. 89-100.
11. R. Jain, "Characteristics of destination address locality in computer network," DEC-TR-592, 1990.
12. J. Mogul, "Network locality at the scale of processes," in *Proceedings of ACM SIGCOMM '91, Conference on Communications, Architectures, and Protocols*, 1991, pp. 273-284.
13. P. R. Sheu and C. M. Lin, "An optimal time slot assignment algorithm for CRMA high-speed networks," *Computer Communications*, Vol. 21, No. 7, 1998, pp. 662-666.
14. S. Olariu, "An optimal greedy heuristic to color interval graphs," *Information Processing Letters*, Vol. 37, 1991, pp. 21-25.

Pi-Rong Sheu (許丕榮) received his Ph.D. degree in Computer Science from National Tsing Hua University, Taiwan, Republic of China, in 1992. Since 1994, he has been an associate professor in the Department of Electrical Engineering at National Yunlin University of Science and Technology, Taiwan, Republic of China. His research interests include high-speed networks, multimedia networking, and computer algorithms.

Charng-Maw Lin (林長茂) received his M.S. degree in Electrical Engineering from National Yunlin University of Science and Technology, Taiwan, Republic of China. His research interests include wireless communications, system integration, and communication networks.