A Dynamic Queue Protocol for Multiaccess Wireless Networks With Multipacket Reception

Qing Zhao, *Member, IEEE,* and Lang Tong, *Fellow, IEEE*

Abstract—**A dynamic medium access control (MAC) protocol is proposed for a finite-user slotted channel with multipacket reception (MPR). This protocol divides the time axis into transmission periods (TPs) where the ith TP is dedicated to the transmission of the packets generated in the $(i-1)$th TP. At the beginning of each TP, the state (active or idle) of each user is estimated based on the length of the previous TP and the incoming traffic load. By exploiting the information on the state of users and the channel MPR capability, the number of users who can simultaneously access the channel in the current TP is chosen so that the expected length of this TP is minimized. As a result, the MPR capability is more efficiently utilized by the proposed protocol as compared to, for example, the slotted ALOHA with optimal retransmission probability. Furthermore, the proposed protocol requires little on-line computation. Its simplicity is comparable to that of slotted ALOHA. It can be applied to random access networks with spread spectrum and/or antenna array.**

Index Terms—**Medium access control (MAC), multipacket reception (MPR), random access network.**

I. INTRODUCTION

IN MULTIACCESS wireless networks where a common channel is shared by a population of users, both the reception capability of the common wireless channel and the efficiency of the medium access control (MAC) protocol affect the network performance. The conventional assumption on the reception capability of the common channel is that a packet is successfully received if and only if there is no concurrent transmissions. Based on such a noiseless collision channel model, MAC protocols are sought to coordinate the transmissions of all users for the efficient utilization of the limited channel reception capability. Numerous protocols, such as ALOHA [1], [23], the tree algorithm [6], the first-come first-serve (FCFS) algorithm [9], and a class of adaptive schemes [5], [12], [13], [17], have been proposed and their performance studied.

The development of spread spectrum multiple access, space-time coding, and new signal processing techniques makes the correct reception of one or more packets in the presence of other simultaneous transmissions possible. While promising improvement in the overall performance of the network, this multipacket reception (MPR) capability also raises important questions. 1) How does the MPR capability at the physical layer affect the performance of existing MAC protocols? (2) How should we design the MAC layer to fully exploit the MPR capability at the physical layer? Many researchers have provided answers to the first question. Being the first random access protocol, the application of ALOHA to networks with MPR capability has been thoroughly studied. In [2], [8], [27], [31] and references therein, slotted ALOHA is applied to networks with capture effect. In [10] and [11], a general model for channels with MPR capability is developed and the performance of slotted ALOHA analyzed for infinite population case. In [4], the impact of MPR on the performance of slotted ALOHA is compared between two types of code-division-vision multiple-access (CDMA) wireless local area networks (LANs)—base station controlled and ad hoc networks—based on a finite population model. Other random access protocols such as the FCFS algorithm and the window protocol [21] have also been extended to networks with capture effect and their performance evaluated [3], [19], [25], [28]. The application of contention free scheme TDMA to networks with MPR capability is another interesting research topic. In [7] and [16], the authors address the use of time-division multiple access (TDMA) in fully connected half-duplex ad hoc networks with MPR provided by multiple independent collision channels. In [24], dynamic time slot allocation is introduced for cellular systems with antenna arrays. Given a set of active users (users with packets to transmit), the proposed dynamic slot allocation scheme assigns an appropriate number of active users to each time slot to utilize the MPR capability provided by the antenna array.

Answers to the second question, however, are scarce in the literature. The Multi-Queue Service Room (MQSR) protocol proposed in [30] is perhaps the first MAC protocol designed explicitly for networks with MPR capability. By optimally exploiting all available information up to the current slot, this protocol grants access to the channel to an appropriate subset of users so that the expected number of successfully received packets is maximized in each slot, leading to the optimal utilization of the channel MPR capability. The difficulty of the MQSR protocol, however, lies in its computational complexity which grows exponentially with the number of users in the network.

In this paper, we propose, for general MPR channels, a MAC protocol that achieves a performance comparable to that of the MQSR protocol, but with a much simpler implementation. Similar to the structure of collision resolution interval in the dynamic tree protocol [5], the proposed scheme, referred to as the

Q. Zhao is with the Department of Electrical and Computer Engineering, University of California, Davis, CA 95616 USA (e-mail: qzhao@ece.ucdavis.edu).

L. Tong is with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: ltong@ece.cornell.edu).

dynamic queue protocol, divides the time axis into transmission periods (TP) where the ith TP is dedicated to the transmission of the packets generated in the $(i-1)$th TP. With such a transmission structure, our knowledge on the state of users at the beginning of the ith TP can be characterized by the probability q_i that a user has a packet to transmit in the ith TP, which depends on the incoming traffic load and the duration of the $(i-1)$th TP. Based on q_i and the channel MPR capability, the size of the access set which contains users who can simultaneously access the channel is chosen for the ith TP so that the expected duration of this TP is minimized, i.e., all packets generated in the $(i-1)$th TP are successfully transmitted within a minimum number of slots. As a consequence, unnecessary empty slots at light traffic and excessive collision events at heavy traffic are avoided simultaneously, leading to an efficient utilization of the channel MPR capability at any incoming traffic load. Furthermore, the optimal size of access set for each TP is obtained from a look-up table. The online implementation of the proposed protocol is as simple as that of slotted ALOHA.

This paper is organized as follows. In Section II, we present the model of a communication network with MPR capability. In Section III, we propose the dynamic queue protocol. In Section IV, we show that the network employing the dynamic queue protocol will eventually reach the steady state regardless of the initial condition of the network. Hence, steady-state performance measures such as throughput and delay can be used to evaluate the performance of the proposed protocol. Finally, we present simulation examples in Section V, where we compare the throughput and delay of the proposed dynamic queue protocol with that of the optimal scheme MQSR [30] and the slotted ALOHA with optimal retransmission probability.

II. MODEL

We consider a communication network with M users who transmit data to a central controller through a common wireless channel. This network model applies to cellular systems where out-of-cell interference is negligible. Users in the network generate data in the form of equal-sized packets. Transmission time is slotted, and each packet requires one time slot to transmit. With probability p, a user independently generates a packet within each slot.

A. MPR Channel

1) Model: As considered in [4], [10], and [11], the common wireless channel is characterized by the probability of having k successes in a slot when there are n concurrent transmissions as denoted by

$$C_{n,k} = \mathrm{P}[k \text{ packets are correctly received} \mid n$$
$$\text{are transmitted}](1 \le n \le M, 0 \le k \le n).$$

The multipacket reception matrix of the channel in a network with M users is then defined as

$$\mathbf{C} = \begin{pmatrix} C_{1,0} & C_{1,1} & & & \\ C_{2,0} & C_{2,1} & C_{2,2} & & \\ \vdots & \vdots & \vdots & & \\ C_{M,0} & C_{M,1} & C_{M,2} & \cdots & C_{M,M} \end{pmatrix}. \quad (1)$$

Let

$$\mathcal{C}_n \triangleq \sum_{k=1}^{n} k C_{n,k} \quad (2)$$

denote the expected number of correctly received packets when total n packets are transmitted. We then define the capacity of an MPR channel as

$$\eta \triangleq \max_{n=1,\cdots,M} \mathcal{C}_n. \quad (3)$$

Note that the channel capacity we define here differs from Shannon capacity. As defined in (3), η is the maximum number of packets that we can expect to successfully receive within one time slot. It is the maximum throughput the MPR channel can offer, independent of MAC protocols. Let

$$n_0 \triangleq \min\{\arg \max_{n=1,\cdots,M} \mathcal{C}_n\}. \quad (4)$$

We can see that at heavy traffic load, n_0 packets should be transmitted simultaneously to achieve the channel capacity η. Noticing that the number of simultaneously transmitted packets for achieving η may not be unique, we define n_0 as the minimum to save transmission power. For MPR channels with n_0 greater than 1, contention should be preferred at any traffic load in order to fully exploit the channel MPR capability.

2) Examples of MPR Channel: The general model for MPR channels given in (1) applies to various systems with spread spectrum, antenna arrays, or sophisticated signal-processing-based packet separation schemes [26]. It also accommodates, as special examples, the conventional collision channel and channels with capture. The reception matrices of the conventional collision channel and channels with capture are given by

$$\begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 0 & 0 & \cdots & 0 \end{pmatrix}, \quad \begin{pmatrix} 1-p_1 & p_1 & 0 & \cdots & 0 \\ 1-p_2 & p_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 1-p_M & p_M & 0 & \cdots & 0 \end{pmatrix} \quad (5)$$

where p_i is the probability of capture given i simultaneous transmissions. With p_1 smaller than 1, this channel model can easily accommodate noisy scenarios.

Another example of an MPR channel is a CDMA system where each transmitted packet is spread by a randomly generated code with length P. At the central controller, the spreading code of each transmitted packet is assumed known, and a bank of matched filters are used as the receiver. We assume that each packet contains L_p bits. A block error control code is used which corrects up to t errors in each received packet. We consider a noisy environment where the variance of the additive white Gaussian noise is denoted by σ^2. For such a network, we can characterize its physical layer with a reception matrix $\mathbf{C}$ constructed as follows. Under the Gaussian assumption on the multiaccess interference from users with equal power, the bit-error rate (BER) p_e of a packet received in the presence of $n-1$ interfering packets is given by [18]

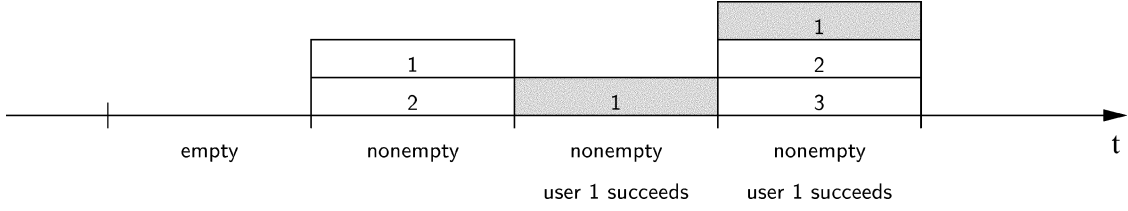$$p_e(n-1) = Q\left(\sqrt{\frac{3P}{n-1+3P\sigma^2}}\right). \quad (6)$$

Fig. 1. Possible outcomes of a slot.

Assuming that errors occur independently in a packet, we then have the packet success probability p_s in the presence of $n - 1$ interfering packets as

$$p_s(n-1) = \sum_{i=0}^{t} B(i, L_p, p_e) \qquad (7)$$

where $B(u, U, s)$ denote the probability mass at the value u of a Binomial random variable with total U trials and a success probability s, i.e.,

$$B(u, U, s) \triangleq \binom{U}{u} s^u (1-s)^{U-u}. \qquad (8)$$

Under the assumption that each matched filter works independently at the receiver, we have

$$C_{n,k} = B\left(k, n, p_s(n-1)\right). \qquad (9)$$

The reception matrix $\mathbf{C}$ serves as an interface between the physical layer and the MAC layer. It characterizes the impact of many physical layer parameters on packet reception at the MAC layer. For example, the correlation property of the spreading codes, receiver design, error control codes, fading characteristics of the channel, power control schemes, and background noise all affect the specific value taken by each entry of $\mathbf{C}$. In this paper, we do not address the construction of $\mathbf{C}$ for various physical layer parameters. The question we seek to answer is, for a given physical layer characterized by a reception matrix $\mathbf{C}$, how to design the MAC layer to fully utilize the reception capability of the physical layer.

B. Central Controller

In our network model, access to the common wireless channel is controlled by the central controller. At the beginning of each slot, the central controller chooses and broadcasts an access set which contains users allowed to access the channel in this particular slot. Users and only users in this access set transmit packets if they have any. At the end of this slot, the central controller observes the channel outcome which contains information on whether this slot is empty and whose packets are successfully received. Here, we assume that the central controller can distinguish without error between empty and nonempty slots. Furthermore, if some packets are successfully demodulated at the end of a slot, the central controller can identify the source of these packets. However, if at least one packet is successfully demodulated at the end of the slot, the central controller does not assume the knowledge whether there are other packets transmitted in this slot but not successfully received. We illustrate this point in Fig. 1, where we consider possible outcomes of a slot:

empty, nonempty with success, and nonempty without success (successfully received packets are illustrated by shaded rectangles). To the central controller, the two events that happened in the third and the forth slots are indistinguishable.

After observing the channel outcome, the central controller acknowledges the sources of successfully received packets (if any). Users who transmit but do not receive acknowledgment assume their packets are lost and will retransmit the next time they are enabled. We assume in this paper that the down-link channel (from the central controller to the users) is error free and the time for acknowledgment and broadcasting the access set is negligible.

Our goal here is to design, for a multiaccess network as specified earlier, a random access protocol that adaptively controls the access set according to the channel MPR capability and the current traffic load. It should achieve efficient channel utilization with a simple online implementation.

III. DYNAMIC QUEUE PROTOCOL

A. Structure of Transmission Period

In the proposed dynamic queue protocol, the time axis is divided into transmission periods (TPs) as illustrated in Fig. 2, where we assume that the network starts at time 0 and one slot lasts one time unit. Each TP is dedicated to the transmission of packets generated in the previous TP and ends when the central controller can assert that all packets generated in the previous TP have been successfully transmitted.

We assume that besides the packet waiting for transmission in the current TP, each user can hold at most one packet newly generated in the current TP and to be transmitted in the next one. Thus, in each TP, each user has at most one packet to transmit.[1] Let q_i denote the probability that a user has a packet to transmit in the ith $(i \geq 1)$ TP. Recall that p denotes the probability that a user generates a packet within one time slot. We have

$$q_i = 1 - (1-p)^{L_{i-1}} \qquad (10)$$

where L_{i-1} $(i \geq 2)$ denotes the length of the $(i-1)$th TP defined as the number of slots it contains; L_0 specifies the network initial condition and is known to the central controller. Thus, q_i carries our knowledge on the state of each user at the beginning of the ith TP. Based on q_i and the channel reception matrix $\mathbf{C}$, the size N_i of the access set which contains users who can simultaneously access the channel in the ith TP is chosen optimally (see Section III-C). Packets generated in the $(i-1)$th

[1]This single buffer assumption is commonly used in MAC protocol design. By considering a user with multiple buffers as multiple users with a single buffer, protocols designed under the single buffer assumption may also be applied to multiple-buffer scenarios.
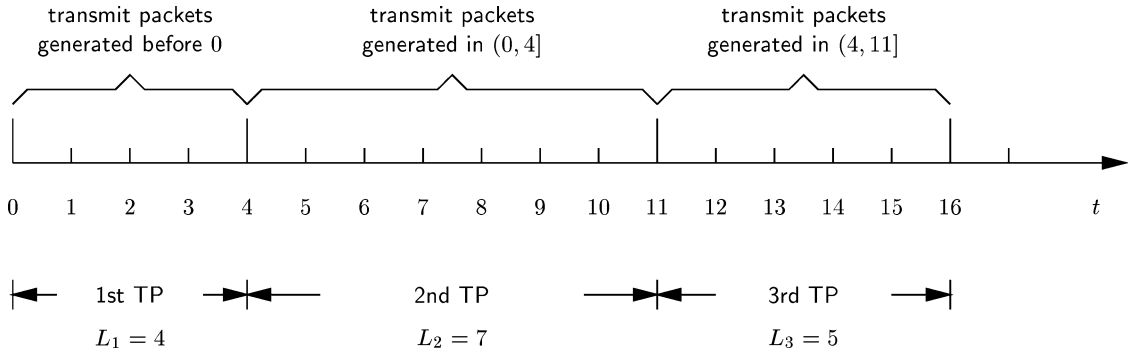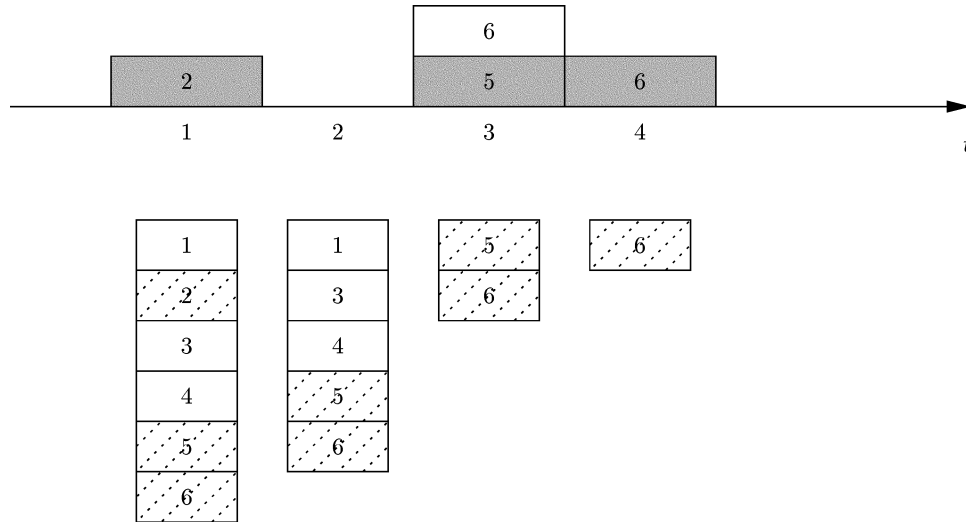
Fig. 2. Structure of transmission period.

Fig. 3. Basic procedure of the dynamic queue protocol.

TP are then transmitted according to the procedure specified in Section III-B.

B. Structure of the Dynamic Queue Protocol

The basic structure of the dynamic queue protocol is a waiting queue which consists of unprocessed users. A user is said unprocessed if its state (whether or not it has a packet to transmit in this TP) is unknown to the central controller. Hence, at the beginning of the ith TP, all M users are waiting in the queue for the transmission of their packets generated in the $(i-1)$th TP. After a user is processed, i.e., either its packet generated in the $(i-1)$th TP has been successfully received or it has been identified by the central controller as inactive (did not generate packet in the $(i-1)$th TP), the central controller removes it from the waiting queue. The current TP ends when the queue becomes empty (all M users are processed).

Specifically, based on q_i given by (10), N_i, the size of the access set for this TP, is chosen. Then, the first N_i users in the queue are enabled to access the channel in the first slot of the ith TP. At the end of this slot, the central controller detects whether this slot is empty or not. If it is empty, all these N_i users are processed and the next N_i users in the queue are enabled in the next slot. On the other hand, if this slot is not empty and $k\,(k \geq 0)$ packets are successfully received, the sources of these k packets are processed and removed from the waiting queue;

the rest $N_i - k$ users along with the next k users in the queue are enabled to access the channel in the next slot. This procedure continues until all M users are processed.

We illustrate in Fig. 3 the basic procedure of the dynamic queue protocol, where we consider the ith TP in a network with $M = 6$. Suppose that each of user 2, 5, and 6 (shaded with dashed lines) has generated a packet in the $(i-1)$th TP, and we choose $N_i = 3$ for the ith TP. As shown in Fig. 3, at the beginning of the ith TP, all 6 users are waiting in a queue to access the channel. In the first slot of this TP, users 1, 2, and 3 are enabled, resulting in a successful transmission by user 2 (shaded rectangles indicate successfully transmitted packets). User 2 is then processed and removed from the waiting queue. Users 1, 3, and 4 are then enabled in the second slot of this TP, resulting in an empty slot in which all these three users are processed. Finally, the only two users left in the queue, namely, users 5 and 6, access the channel in the third slot. At the end of the fourth slot, all users are processed, leading to the end of the ith TP and the beginning of the $(i+1)$th TP.

With this structure, the only parameter to be designed is N_i, the size of the access set for the ith $(i \geq 1)$ TP, which we discuss in Section III-C.

We point out that the order of a user in the waiting queue affects its average packet delay. While all users generate packets simultaneously, users in the front of the queue access the channel before users in the end of the queue unless $N_i = M$.
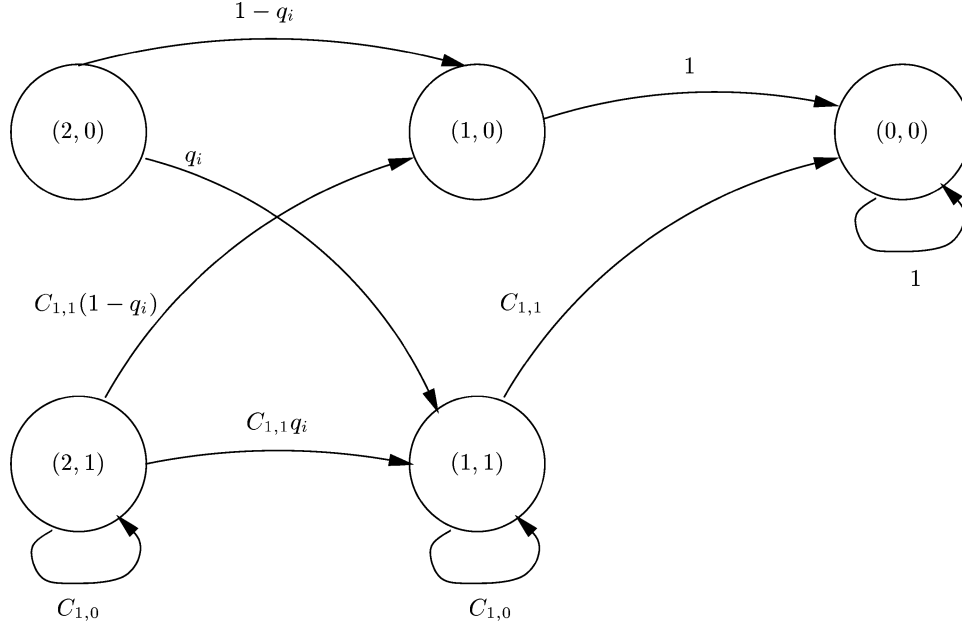
Fig. 4.   State transition diagram.

If priority among users is desirable, users with higher priority should be in the front of the queue in each TP. Otherwise, the order of a user in the queue needs to be randomized at the beginning of each TP to ensure fairness. In Section V, simulation examples show that the average delay for the last user in the queue can be twice as large as that for the first user at medium and heavy traffic load.

C. Optimal Access Set

The optimal size N_i of the access set for the ith TP is chosen so that the expected length of this TP is minimized, i.e., the expected number of slots for processing all M users, each with probability q_i having a packet, is minimized. Specifically, N_i is determined by

$$N_i = \arg \min_{N=1,\cdots,M} E[L_i|q_i, N] \tag{11}$$

where $E[L_i \mid q_i, N]$ is the expected length of the ith TP when each user with probability q_i has a packet to transmit and the size of the access set is N.

In order to determine N_i, we calculate $E[L_i \mid q_i, N]$ as the absorbing time of a finite-state discrete Markov chain. It can be shown that the number of unprocessed users at the beginning of a slot along with the number of packets that will be transmitted in this slot forms a Markov chain. Specifically, at the beginning of a slot in the ith TP, the network is in state

(j, k) if there are j $(j = 0, \cdots, M)$ unprocessed users and k $(k = 0, \cdots, \min\{N, j\})$ packets to be transmitted in this slot when the size of access set is chosen to be N. A state diagram of this Markov chain for $M = 2$ and $N = 1$ is illustrated in Fig. 4. With probability q_i, the first user in the queue has a packet to transmit in the ith TP. Thus, with probability q_i, the Markov chain starts with state (2,1), and with probability $1 - q_i$, it starts with state (2,0). Take state (2,1) for example. With probability $C_{1,0}$, the transmission by the first user in the queue does not succeed. The chain then stays in (2,1). With probability $C_{1,1}q_i$, the transmission by the first user succeeds and the second user in the queue has a packet. The chain then jumps to state (1,1). With probability $C_{1,1}(1 - q_i)$, the chain jumps to state (1,0).

In general, the transition probability from state (j, k) to state (l, m) is given by (12), located at the bottom of the page. The initial condition of this Markov chain is given by

$$P[X_0 = (M, k)] = B(k, N, q_i), \quad k = 0, \ldots, N \tag{13}$$

where X_0 denote the initial state of the Markov chain. With state (0,0) defined as the absorbing state, $E[L_i \mid q_i, N]$ is the absorbing time of this Markov chain, which is defined as the expected number of transitions until the first hit of state (0,0). Define $e_{(j,k)}$ as the expected remaining time until absorption given that the current state is (j, k). Let

$$\mathbf{e} \triangleq \left[e_{(M,0)}, \cdots, e_{(M,N)}, e_{(M-1,0)}, \cdots, e_{(1,0)}, e_{(1,1)} \right]^t. \tag{14}$$

$$r_{(j,k),(l,m)} = \begin{cases} B\left(m, \min\{N, l\}, q_i\right) \\ \quad (\text{if } k = 0, l = \max\{j - N, 0\}, 0 \le m \le \min\{N, l\}) \\ C_{k,j-l}B\left(m - k + j - l, \min\left\{j - l, \max\{j - N, 0\}\right\}, q_i\right) \\ \quad (\text{if } 1 \le k \le \min\{N, j\}, j - k \le l \le j, k - (j - l) \le m \le \min\{k, l\}) \\ 0 \quad (\text{otherwise}) \end{cases} \tag{12}$$

The Dynamic Queue Protocol

In the ith ($i \geq 1$) TP,

1. Let W denote the number of unprocessed users. Set $W = M$.

2. Compute q_i as given by (10).

3. Choose N_i based on q_i from the look-up table.

4. The first N_i users form the access set.

5. Users in the access set access the channel in the current slot.

6. At the end of this slot,

 • if the slot is empty, update W as $W = W - A$, where A is the size of the access set; remove all users from the access set; the next $\min\{N_i, W\}$ users in the queue form the access set;

 • if the slot is not empty and k packets are successfully received, remove the source of these k packets from the access set; update W as $W = W - k$; the next $\min\{k, W\}$ users in the queue join the access set.

7. Repeat Step 5 and Step 6 until $W = 0$. This starts the $(i+1)$th TP.

Fig. 5. Dynamic queue protocol.

We then have (see [14] for analysis on the absorbing time of Markov chains)

$$(\mathbf{I} - \mathbf{P})\mathbf{e} = \mathbf{1}, \tag{15}$$

where $\mathbf{P}$ is the transition probability matrix [after removing state (0,0)] with entries specified by (12), $\mathbf{I}$ and $\mathbf{1}$ denote, respectively, an identity matrix and a vector with all entries equal to 1. From (15), we can solve for $e_{(M,k)}$ for $k = 0, \cdots, N$. Thus, considering the initial condition of the Markov chain given by (13), we can calculate $E[L_i \mid q_i, N]$ as

$$E[L_i | q_i, N] = \sum_{k=0}^{N} B(k, N, q_i) e_{(M,k)}. \tag{16}$$

With $E[L_i \mid q_i, N]$ computed for all possible N, the optimal size N_i of the access set for the ith TP can be easily obtained from (11).

We point out that the optimal size of the access set can be computed off line. By varying q_i from 0 to 1, we can construct a table that specifies the interval of q_i in which a size $N \in \{1, \cdots, M\}$ of the access set is optimal (a typical look-up table is illustrated in Fig. 7). Thus, when the network starts, the optimal size of the access set for each TP can be obtained from this table; little online computation is required to implement the dynamic queue protocol.

The basic procedure of the dynamic queue protocol is given in Fig. 5.

D. Connections With Existing MAC Protocols

In this section, we draw connections between the proposed dynamic queue protocol and existing MAC schemes. In particular, we consider the dynamic tree [5] protocol proposed for the noiseless collision channel and the MQSR protocol [30] proposed for MPR channels.

1) Dynamic Tree Protocol: The similarity between the dynamic queue and the dynamic tree protocol is the structure of transmission period. Although the terminology of collision resolution interval (CRI) is used in the dynamic tree protocol, both protocols have the property that newly generated packets can not be transmitted until the current TP (CRI) ends. This ensures that a single parameter q_i is sufficient to characterize our knowledge on the state of users at the beginning of each TP (CRI).

The main difference between these two protocols lies in their schemes of determining the access set for each slot in a particular TP (CRI). Proposed exclusively for the noiseless collision channel, the dynamic tree protocol utilizes the binary tree algorithm [6] for determining the access set for each slot. However, like other splitting algorithms such as FCFS [9], the tree algorithm relies on three assumptions that do not hold in a general MPR channel. First, the tree algorithm relies on the property that a successful transmission in the noiseless collision channel implies that other users in the access set do not have packets. Thus, in a slot with a success of one user, all users in the access set are processed. This, however, is not true in a general MPR channel (see Fig. 1) where any user in the access set from whom we do not receive a packet in a nonempty slot is unprocessed. If we insist on the tree structure in protocols designed for MPR channels, the access set may have to be unnecessarily shrunk in order to enable the unprocessed users left in the access set after a slot with success. This is due to the boundaries among nodes on the same level of a tree. The second assumption made by the dynamic tree protocol is that packet collisions can only be resolved by splitting of users. However, the MPR capability opens new options for resolving a collision. Splitting is not always necessary or even sensible. Take, for example, an MPR channel with $C_{2,0} = \epsilon$ and $C_{2,2} = 1 - \epsilon$ where $0 < \epsilon \ll 1$. When two packets are simultaneously transmitted and none successfully received, in stead of splitting, it is more sensible to enable the same set of users again. Actually, it has been shown in [20] that when the channel has a moderate level of MPR capability, immediate retransmission without random backoff (one form of splitting users) is optimal. Finally, the dynamic tree protocol does not

take noise into account. Any nonempty slot without success is assumed to be a consequence of collision and, hence, leads to a splitting of users. This causes unnecessary empty slots in the case that only one user in the access set transmits but does not succeed because of noise.

In the dynamic queue protocol, in stead of a tree, a queue structure is utilized for determining the access set for each slot. The boundaries among users are eliminated by the queue structure. By letting the next k users in the waiting queue join the access set after a slot with k successes, we can keep the size of the access set to be N_i which has been chosen optimally (in terms of minimizing $E[L_i]$). Furthermore, the same N_i users are enabled after a nonempty slot with no success, where N_i has been chosen according to the channel MPR capability. This enables us to exploit the MPR capability for collision resolution (in the case of more than one active user) and avoid unnecessary splitting of users (in the case of one active user in a noisy environment). Note that $E[L_i \mid q_i, N] = \infty$ if $\mathcal{C}_N = 0$. We thus have

$$N_i \in \{N : 1 \leq N \leq M, \mathcal{C}_N > 0\} \qquad (17)$$

i.e., the optimal size of access set always enables us to resolve packet collisions via the channel MPR capability.

2) MQSR Protocol: As their names suggest, both the MQSR and the dynamic queue protocol utilizes a queue structure for determining the access set for each slot. The difference between them is the amount of information they exploit for choosing the size of access set.

In the MQSR protocol, in order to maximize per-slot throughput, the size of access set for each slot is chosen by exploiting all the information that is available at the beginning of this slot. Since the outcome of each slot provides information on the state of users, the size of access set is updated at the beginning of each slot in order to incorporating the newly available information. However, this update of the size of access set on a slot-by-slot basis results in the high computational complexity of the MQSR protocol.

With the structure of transmission period, the dynamic queue protocol utilizes only the information available at the beginning of each TP for determining the size of access set. Once the size of the access set is chosen at the beginning of a TP, it is used in all slots in this TP, except when the number of unprocessed users left in the waiting queue is smaller than the size of the access set. Information obtained from the outcome of each slot within a TP is used only for determining whether a user is processed or not, but not for updating the size of the access set. It turns out that by fixing the size of access set for the whole TP, determining it becomes as simple as looking up a table. The price we paid for this simple implementation is performance. Nevertheless, extensive simulations demonstrate that the performance of the dynamic queue protocol is comparable to that of the optimal MQSR protocol (see Fig. 8, 11).

IV. STEADY-STATE PERFORMANCE ANALYSIS

Our main concern with MAC protocols is their long-term behavior (when the initial condition of the network becomes irrelevant). Thus, steady-state performance measures such as throughput and average delay are commonly used for evaluating a MAC protocol. In order to evaluate the throughput and delay performance of the dynamic queue protocol, we need to show that the network employing the proposed protocol will eventually reach a steady state regardless of the initial condition L_0. In this section, we first study the existence and the uniqueness of the steady state of the dynamic queue protocol. Throughput and average delay provided by the proposed protocol is then analyzed.

A. Existence of Steady State

Given the channel reception matrix $\mathbf{C}$ and the incoming traffic load p, the optimal size N_i of the access set for the ith TP is a function of L_{i-1}, i.e., $N_i = f_{\mathbf{C},p}(L_{i-1})$. In general, $f_{\mathbf{C},p}(\cdot)$ is a monotonic decreasing[2] function as illustrated in Fig. 6. It is completely determined by $\mathbf{C}$ and p and can be computed offline. Suppose that the range of $f_{\mathbf{C},p}(\cdot)$ is $\{\underline{n}_1, \ldots, \underline{n}_J\}$ with $\underline{n}_1 > \underline{n}_2 > \ldots > \underline{n}_J$. We then define

$$\underline{l}_j \triangleq \min \{l : f_{\mathbf{C},p}(l) = \underline{n}_j\}, \quad j = 1, \cdots, J. \qquad (18)$$

It can be shown [29] that $\{L_i\}_{i=0}^{\infty}$ is a homogeneous Markov process with infinite state space $\mathcal{Z}^+$ ($\mathcal{Z}^+$ denotes the set of positive integers) and transition probability

$$p_{l,m} \triangleq P[L_i = m \mid L_{i-1} = l]. \qquad (19)$$

The steady state of a network using the dynamic queue protocol is then defined as the stationary distribution of $\{L_i\}_{i=0}^{\infty}$. Before using steady-state performance measures such as throughput and average delay, questions about the existence and uniqueness of the network steady states must be resolved.

Theorem 1: Suppose that $f_{\mathbf{C},p}(\cdot)$ is a monotone decreasing function with range $\{\underline{n}_1, \cdots, \underline{n}_J\}$. Let

$$\mathcal{Q}_1 = \left\{1, 2, \cdots, \left\lceil \frac{M}{\underline{n}_w} \right\rceil - 1\right\}, \quad \mathcal{Q}_2 = \left\{\left\lceil \frac{M}{\underline{n}_w} \right\rceil, \left\lceil \frac{M}{\underline{n}_w} \right\rceil + 1, \cdots\right\} \qquad (20)$$

be a partition of the state space $\mathcal{Z}^+$, where

$$w \triangleq \max \left\{j : \left\lceil \frac{M}{\underline{n}_j} \right\rceil \geq \underline{l}_j, 1 \leq j \leq J\right\}. \qquad (21)$$

Consider a noisy environment with $0 < C_{1,0} < 1$. We have, for $p \in (0, 1)$:

T1.1) all states in $\mathcal{Q}_1$ are transient;

T1.2) if the initial distribution of $\{L_i\}_{i=0}^{\infty}$ is such that $P[L_0 \in \mathcal{Q}_2] = 1$, then $\{L_i\}_{i=0}^{\infty}$ is ergodic;

T1.3) $\{L_i\}_{i=0}^{\infty}$ has a limiting distribution $\{\pi_l\}_{l \in \mathcal{S}}$ satisfying

$$\pi_l \begin{cases} > 0, & \text{if } l \in \mathcal{Q}_2 \\ = 0, & \text{if } l \in \mathcal{Q}_1 \end{cases}. \qquad (22)$$

The proof of Theorem 1 can be found in [29]. Theorem 1 shows that a network which employs the dynamic queue pro-

---

[2]A heuristic argument for $f_{\mathbf{C},p}(\cdot)$ being monotonic decreasing is as follows. In order to process all packets generated in the $(i-1)$th TP within a minimum number of slots, n_0 [as defined in (4)] packets should be transmitted simultaneously in each slot. With a smaller L_{i-1}, the probability that a user has a packet to transmit in the ith TP is smaller. Hence, the access set for the ith TP need to be enlarged so that the total number of packets held by users in the access set approaches n_0.
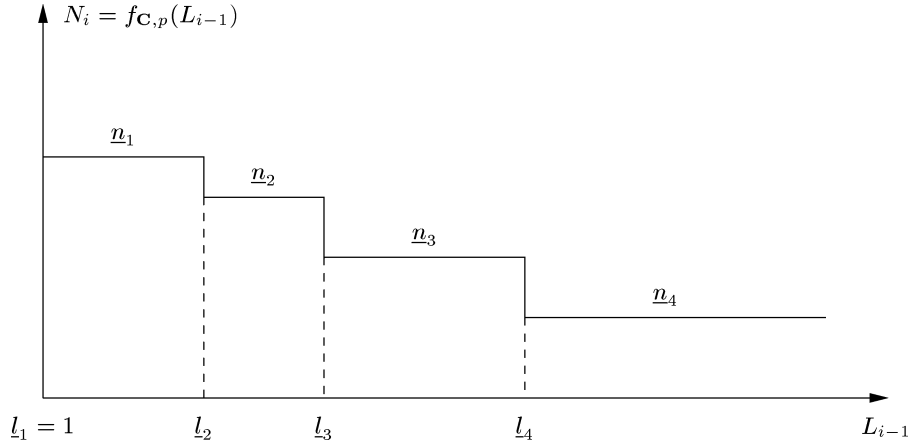
Fig. 6. N_i as a function of L_{i-1}.

tocol will eventually reach a unique steady state. Thus, we can use measures such as throughput and average delay to study the long term behavior of the dynamic queue protocol.

B. Throughput and Packet Delay

The throughput U is defined as the average number of packets successfully transmitted within one time slot. The average packet delay D is defined as the average number of slots from the time a packet is generated to that it is successfully transmitted. Based on the ergodicity of $\{L_i\}_{i=0}^\infty$, we have,[3] at an arbitrary traffic load p

$$U = \frac{\sum_{l \in \mathcal{Q}_2} \left(1 - (1-p)^l\right) M \pi_l}{\sum_{l \in \mathcal{Q}_2} l \pi_l}, \qquad (23)$$

$$D \leq \sum_{l,m \in \mathcal{Q}_2} \left(l + m - \frac{p \sum_{k=1}^l k(1-p)^{k-1}}{1 - (1-p)^l} + 0.5 \right)$$
$$\times \pi_l p_{l,m}, \qquad (24)$$

where $\{\pi_l\}_{l \in \mathcal{Z}^+}$ and $\{p_{l,m}\}_{l,m \in \mathcal{Z}^+}$ denote, respectively, the limiting distribution and transition probability of the Markov process $\{L_i\}_{i=0}^\infty$. In general, these two quantities are difficult to obtain even numerically. However, at heavy traffic load $p = 1$, both U and D can be studied analytically, as shown in the following.

At $p = 1$, we have $q_i = 1$ for any i. It then follows that $\{L_i\}_{i=0}^\infty$ is an i.i.d. sequence. The throughput and average delay for $p = 1$ are given by

$$U = \frac{M}{\min_{N=1,\cdots,M} E[L_i \mid q_i = 1, N]} \qquad (25)$$

$$D \leq 2 \min_{N=1,\cdots,M} E[L_i \mid q_i = 1, N]. \qquad (26)$$

As shown in Section III-C, $E[L_i \mid q_i = 1, N]$ can be obtained by analyzing the absorbing time of a finite state Markov chain as illustrated in Fig. 4. With $q_i = 1$ for all i, we can simplify the state of this Markov chain to the number j of unprocessed users. The transition probability then becomes

$$r_{j,l} = \begin{cases} C_{\min\{N,j\},j-l} & \text{if } 0 \leq l \leq j \\ 0 & \text{otherwise} \end{cases}. \qquad (27)$$

[3]A detailed derivation can be found in [29].

The initial condition of this Markov chain is given by $P[X_0 = M] = 1$. With state 0 defined as the absorbing state, $E[L_i \mid q_i = 1, N]$ can be obtained as

$$E[L_i \mid q_i = 1, N] = e_M \qquad (28)$$

where e_M is the absorbing time of the Markov chain. With $E[L_i \mid q_i = 1, N]$ computed for all possible N, the throughput and an upper bound on the average delay at $p = 1$ can be easily obtained from (25) and (26).

V. SIMULATION EXAMPLES—MPR VIA SPREAD SPECTRUM

We consider here a M-user CDMA network with randomly generated spreading codes. The reception matrix $\mathbf{C}$ of the network is given by (6), (7), and (9).

A. Throughput

In this example, we compare the throughput performance of the dynamic queue protocol with that of the MQSR protocol and the slotted ALOHA with optimal retransmission probability. We considered a network with $M = 10$. The packet length L_p, spreading gain P, and the number of correctable errors in a packet were, respectively, 200, 6, and 2. The noise variance was given by $10 \log_{10}(1/\sigma^2) = 10$ dB. The capacity of the MPR channel in such a network is 1.7925, which can be achieved by transmitting $n_0 = 2$ packets in each slot.

We first construct the look-up table that specifies the q_i intervals in which a possible size (from 1 to 10) of access set is optimal. The result is shown in Fig. 7. This result demonstrates clearly the trend that the heavier the traffic is (larger q_i), the smaller the access set should be, as intuition suggests. Note that the optimal size of access set equals to n_0 which is greater than 1 at the heaviest traffic load ($q_i = 1$), indicating that contention is preferable at any traffic load for this MPR channel. Fig. 7 also shows that a size of access set is optimal for a range of traffic load. This implies that the proposed protocol is robust to estimation errors on the packet arrival rate p.

In Fig. 8, the throughput performance of the dynamic queue protocol at different incoming traffic load p is compared to that of the MQSR protocol [30] and the delayed first transmission ALOHA with optimal retransmission probability. Here, we intentionally favored the slotted ALOHA by letting it choose
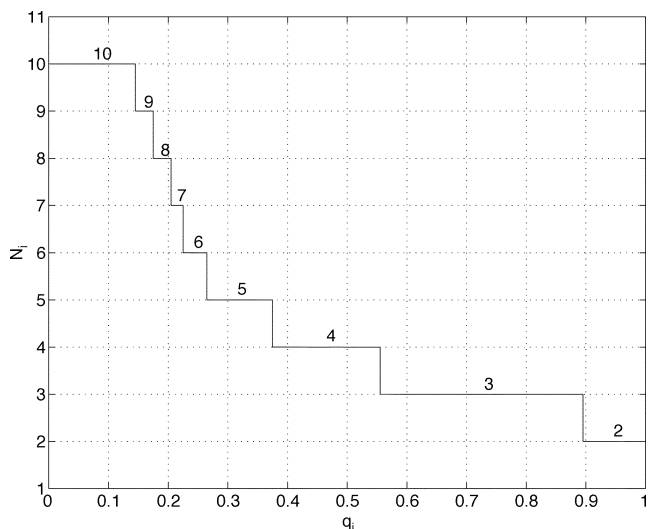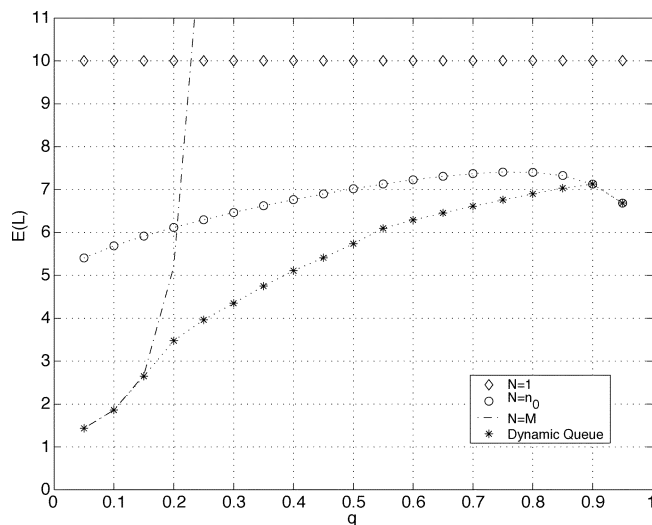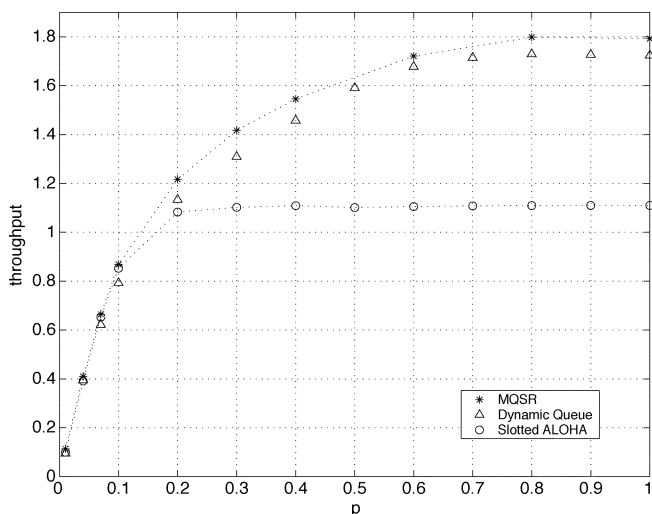
Fig. 7. Optimal size of access set.

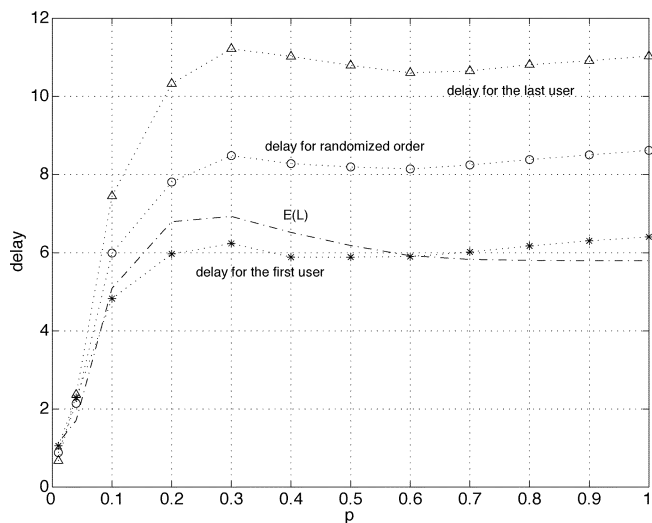Fig. 8. Throughput comparison.

Fig. 9. Expected length of a TP.

Fig. 10. Average packet delay.

the optimal retransmission probability. Comparing the performance of the dynamic queue protocol with that of the slotted ALOHA with optimal retransmission probability, we see a 55% throughput gain at medium and heavy traffic load. Compared to the MQSR protocol which aims to determine the access set for each slot by optimally exploiting all available information, the dynamic queue protocol achieved comparable performance with a much simpler implementation. Note that the throughput provided by the dynamic queue protocol at heavy traffic load approached to the channel capacity 1.7925.

B. Average Delay

Here, we study the delay performance of the dynamic queue protocol in the CDMA network specified in Section V-A. We first study the expected length of a TP in the dynamic queue protocol, which is closely related to the average packet delay. In Fig. 9, the expected length of a TP is plotted as a function of q, the probability that a user has a packet to transmit in this TP. The expected length of TP achieved by the dynamic queue protocol is compared with those provided by schemes with fixed size of

access set $N = 1$, $N = n_0$, and $N = M$. Fig. 9 shows that $N = 1$ and $N = n_0$ yielded inferior performance for small q due to excessive empty slots while $N = M$ gave inferior performance for large q due to excessive packet collision. The TDMA scheme $(N = 1)$ performed worse than the dynamic queue protocol even for large q. This is due to the fact that the channel MPR capability was not fully utilized when $N = 1$. Compared with schemes with fixed size of access set, the advantage of dynamically changing the size of access set according to the traffic load q is obvious.

The average packet delay provided by the dynamic queue protocol as a function of the incoming traffic load p is shown in Fig. 10. We consider two cases—fixed order of users and randomized order of users. From Fig. 10, we see that in the case of fixed order of users, the average delay for the last user in the queue could be twice as large as that for the first user in the queue at medium and heavy traffic load, while at light traffic load, they were about the same. The reason for this is that at medium and heavy traffic load, $N_i < M$ for most transmission periods. In this case, the first user in the queue always access the channel before the last one. While at light traffic load when
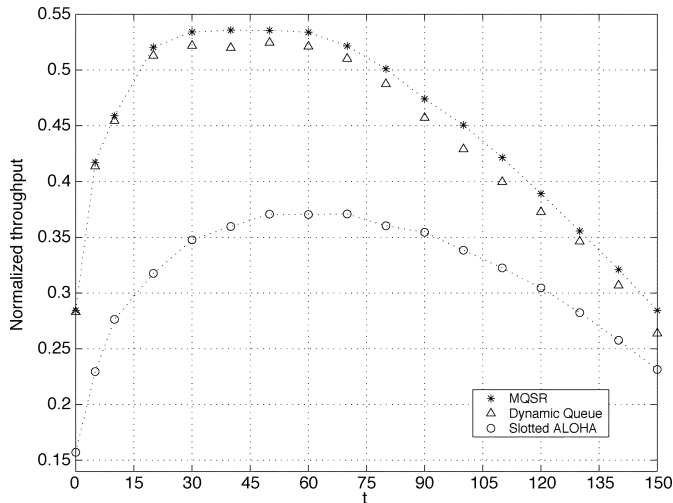
Fig. 11. Normalized throughput at $p = 1$.

we usually have $N_i = M$, all users access the channel simultaneously, resulting in the same average delay for the first and last user in the queue. For the case of randomizing the order of users, the average packet delay for a user was about the average of the delay for the first user and the delay for the last user in the case of fixed order of users. From Fig. 10, we can also see that the average length of a TP $E[L]$ could be a good estimate of the packet delay for the first user in the queue. At medium traffic load $(0.1 \leq p \leq 0.6)$, the average delay for the first user in the queue was slightly smaller than $E[L]$, while at heavy traffic load $(p > 0.6)$, it was the other way around. The reason for this is that when the traffic is heavy, with high probability, a user will generate a packet within the first several slots in a TP. Even for the first user in the queue, it has to wait for almost a whole TP before this packet can be transmitted. For the last user in the queue, its average delay will approach the length of two transmission periods at heavy traffic load, as confirmed by Fig. 10.

C. Normalized Throughput

While the techniques of spread spectrum and error control strengthen the channel reception capability, they consume bandwidth. In this example, we study the normalized throughput of the dynamic queue protocol, where we define the normalized throughput as the average number of information bits successfully transmitted per second per hertz [22]. We assume here a binary phase-shift keying (BPSK) modulation. Given the network throughput U, spreading gain P, packet length L_p, coding rate r_c, and symbol duration T_s, the average number of successfully transmitted information bits per slot is $L_p r_c U$; the duration of each time slot is $L_p T_s$ and the bandwidth (P/T_s). Hence, the normalized throughput U_n is given by

$$U_n = \frac{L_p r_c U}{L_p T_s \frac{P}{T_s}} = \frac{r_c}{P} U. \tag{29}$$

As shown in [15], the maximum coding rate r_c can be computed from the number t of correctable errors as follows:

$$r_c = 1 + \alpha \log_2(\alpha) + (1 - \alpha) \log_2(1 - \alpha), \text{ where } \alpha = \frac{2t + 1}{L_p}. \tag{30}$$

We compare the normalized throughput of the dynamic queue protocol with that of the MQSR protocol and the slotted ALOHA with optimal retransmission probability. We choose $p = 1$ for the reason that all three protocols yield maximum throughput at this heaviest traffic load. The network parameters were chosen as $M = 200$, $L_p = 1000$, $P = 10$, and $\sigma^2 = 0$. The normalized throughput of the dynamic queue, the MQSR, and the slotted ALOHA with optimal retransmission probability at $p = 1$ was theoretically calculated and plotted in Fig. 11 as t, the number of correctable errors within one packet, varies from 0 to 150. From Fig. 11, we again observe that the dynamic queue protocol performed comparably to the optimal MQSR protocol and significantly better than the slotted ALOHA with optimal retransmission probability. Note that the MQSR protocol achieves the channel capacity at $p = 1$ which has been shown theoretically in [30]. A comparable performance to it implies that the throughput provided by the dynamic queue protocol approaches to the channel capacity at heavy traffic load. Furthermore, Fig. 11 shows that to achieve the best bandwidth efficiency, we should choose a block error control code which corrects up to $t = 30$ errors out of a packet with 1000 b for the dynamic queue and the MQSR protocol. For the slotted ALOHA with optimal retransmission probability, however, we should choose $t = 60$. A block error control code with stronger correction capability is, in general, more difficult to design.

VI. CONCLUSION

We have proposed in this paper the dynamic queue protocol for multiaccess networks with MPR capability. According to the traffic load and the channel MPR capability, this protocol adaptively controls the number of users who gain access to the channel in the same slot. As a consequence, unnecessary empty slots at light traffic and excessive collision events at heavy traffic are avoided simultaneously, leading to efficient channel utilization at any incoming traffic load. Furthermore, the proposed protocol is particularly attractive in its simple implementation.

In this paper, we have assumed an error-free downlink channel. How to adapt the dynamic queue protocol to scenarios where the feedback channel is prone to error and its performance in such cases are interesting research topics for the future. Random access protocol design for cellular systems with severe intercell interference also deserves careful study.

REFERENCES

[1] N. Abramson, "The Aloha system—Another alternative for computer communications," in *Proc. Fall Joint Computer Conf., AFIPS*, 1970, p. 37.

[2] ——, "The throughput of packet broadcasting channels," *IEEE Trans. Commun.*, vol. COM-25, pp. 117–128, Jan. 1977.

[3] D. E. Ayyildiz and H. Delic, "Adaptive random access algorithm with improved delay performance," *Int. J. Commun. Syst.*, vol. 14, pp. 531–539, 2001.

[4] J. Q. Bao and L. Tong, "A performance comparison between ad hoc and centrally controlled CDMA wireless LANs," *IEEE Trans. Wireless Commun.*, vol. 1, pp. 829–841, Oct. 2002.

[5] J. I. Capetanakis, "Generalized TDMA: The multi-accessing tree protocol," *IEEE Trans. Commun.*, vol. 27, pp. 1476–1484, Oct. 1979.

[6] ——, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inform. Theory*, vol. 25, pp. 505–515, Sept. 1979.

[7] I. Chlamtac and A. Farago, "An optimal channel access protocol with multiple reception capacity," *IEEE Trans. Comput.*, vol. 43, pp. 480–484, Apr. 1994.

[8] G. del Angel and T. L. Fine, "Randomized power control strategies for optimization of multiple access radio systems," in *Proc. 38th Allerton Conf. Communication, Control Computing*, Oct. 2000.

[9] R. Gallager, "Conflict resolution in random access broadcast networks," in *Proc. AFOSR Workshop Communication Theory and Applications*, Sept. 1978, pp. 74–76.

[10] S. Ghez, S. Verdú, and S. C. Schwartz, "Stability properties of slotted Aloha with multipacket reception capability," *IEEE Trans. Automat. Contr.*, vol. 33, pp. 640–649, July 1988.

[11] ——, "Optimal decentralized control in the random access multipacket channel," *IEEE Trans. Automat. Contr.*, vol. 34, pp. 1153–1163, Nov. 1989.

[12] M. G. Hluchyj, "Multiple access window protocol: Analysis for large finite populations," in *Proc. IEEE Conf. Decision Control*, New York, 1982, pp. 589–595.

[13] M. G. Hluchyj and R. G. Gallager, "Multiaccess of a slotted channel by finitely many users," in *Proc. Nat. Telecommunications Conf.*, New Orleans, LA, Aug. 1981, pp. D4.2.1–D4.2.7.

[14] D. L. Isaacson, *Markov Chains, Theory and Applications*. New York: Wiley, 1976.

[15] R. K. Morrow Jr. and J. S. Lehnert, "Packet throughput in slotted ALOHA DS/SSMA radio systems with random signature sequences," *IEEE Trans. Commun.*, vol. 40, pp. 1223–1230, July 1992.

[16] S. Kim and J. Yeo, "Optimal scheduling in CDMA pakcet radio networks," *Comput. Oper. Res.*, vol. 25, pp. 219–227, Mar. 1998.

[17] L. Kleinrock and Y. Yemini, "An optimal adaptive scheme for multiple access broadcast communication," in *Proc. Int. Conf. Communications*, June 1978, pp. 7.2.1–7.2.5.

[18] J. Lehnert and M. Pursley, "Error probabilities for binary direct-sequence spread-spectrum communications with random signature sequences," *IEEE Trans. Commun.*, vol. COM-35, pp. 87–98, Jan. 1987.

[19] D. F. Lyons and P. Papantoni-Kazakos, "A window random access algorithm for environments with capture," *IEEE Trans. Commun.*, vol. 37, pp. 766–770, July 1989.

[20] V. Naware and L. Tong, "Smart antennas, dumb scheduling for medium access control," in *Proc. CISS*, Baltimore, MD, Mar, 2003.

[21] M. Paterakis and P. Papantoni-Kazakos, "A simple window random access algorithm with advantageous properties," *IEEE Trans. Inform. Theory*, vol. 35, pp. 1124–1130, Sept. 1989.

[22] A. Polydors and J. Sylvester, "Slotted random access spread-spectrum networks: An analytical framework," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 989–1002, July 1987.

[23] L. G. Roberts, "Aloha packet system with and without slots and capture," in *ASS Note 8*. Stanford, CA: Stanford Res. Inst., Adv. Res. Projects Agency, Network Inform. Center, 1972.

[24] F. Shad, T. D. Todd, V. Kezys, and J. Litva, "Dynamic slot allocation (DSA) in indoor SDMA/TDMA using a smart antenna basestation," *IEEE/ACM Trans. Networking*, vol. 9, pp. 69–81, Feb. 2001.

[25] M. Sidi and I. Cidon, "Splitting protocols in presence of capture," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 295–301, Mar. 1985.

[26] L. Tong, Q. Zhao, and G. Mergen, "Multipacket reception in random access wireless networks: From signal processing to optimal medium access control," *IEEE Commun. Mag.*, vol. 39, pp. 108–112, Nov. 2001.

[27] C. Vanderplas and J. P. M. Linnartz, "Stability of mobile slotted ALOHA network with Rayleigh fading, shadowing, and near-far effect," *IEEE Trans. Veh. Technol.*, vol. 39, pp. 359–366, Nov. 1990.

[28] B. Yucel and H. Delic, "Mobile radio window random access algorithm with diversity," *IEEE Trans. Veh. Technol.*, vol. 49, pp. 2060–2070, Nov. 2000.

[29] Q. Zhao and L. Tong. (2002) "A dynamic queue protocol for multiaccess wireless networks with multipacket reception". Tech. Rep. ACSP-TR-11-02-02. [Online](http://people.ece.cornell.edu/ltong)

[30] ——, "A multi-queue service room MAC protocol for wireless networks with multipacket reception," *IEEE/ACM Trans. Networking*, vol. 11, pp. 125–137, February 2003.

[31] M. Zorzi, "Mobile radio slotted ALOHA with capture and diversity," *Wireless Networks*, vol. 1, pp. 227–239, May 1995.

Qing Zhao (S'99-M'02) received the B.S. degree from Sichuan University, Chengdu, China, in 1994, the MS. degree from Fudan University, Shanghai, China, in 1997, and the Ph.D. degree from Cornell University, Ithaca, NY, in 2001, all in electrical engineering.

From 2001 to 2003, she was a Communication System Engineer with Aware, Inc., Bedford, MA. In 2003, she became a Postdoctoral Research Associate with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY. In 2004, she joined the Department of Electrical and Computer Engineering at the University of California, Davis, where she is currently an Assistant Professor. Her research interests include signal processing, communication systems, wireless networking, and information theory, as well as adaptive signal processing for communications, design and analysis of wireless and mobile netowrks, fundamental limits on the performance of large scale *ad hoc* and sensor networks, and energy constrained signal processing and networking techniques.

Dr. Zhao received the IEEE Signal Processing Society Young Author Best Paper Award in 2000.

Lang Tong (S'87–M'91–SM'01–F'04) received the B.E. degree from Tsinghua University, Beijing, China, in 1985, and the M.S. and Ph.D. degrees in electrical engineering from the University of Notre Dame, Notre Dame, IN, in 1987 and 1990, respectively.

He was a Postdoctoral Research Affiliate at the Information Systems Laboratory, Stanford University, Stanford, CA, in 1991. Currently, he is a Professor in the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY. His research interests include statistical signal processing, adaptive receiver design for communication systems, signal processing for communication networks, and information theory.

Dr. Tong received the Young Investigator Award from the Office of Naval Research in 1996, and the Outstanding Young Author Award from the IEEE Circuits and Systems Society.