Distributed Dynamic Channel Access Scheduling for Ad Hoc Networks [1]

Lichun Bao

and

J.J. Garcia-Luna-Aceves
School of Engineering BE399
University of California
Santa Cruz, CA 95064
E-mail: {baolc, jj}@soe.ucsc.edu

Version: 03/15/2002

Three types of collision-free channel access protocols for ad hoc networks are presented. These protocols are derived from a novel approach to contention resolution that allows contending entities to elect one or multiple winners for channel access in any given contention context (e.g., a time slot) in a distributed fashion. In multihop wireless networks, the only required information for each entity is the identifiers of its neighbors one and two hops away. The new protocols are shown to be fair and capable of achieving maximal utilization of the channel bandwidth. The delay and throughput characteristics of the contention resolution algorithms are analyzed, and the performance of the three types of channel access protocols is studied by simulations and compared with that of optimal static scheduling algorithms.

Key Words: Ad hoc network, channel access scheduling, medium access control (MAC), distributed computing.

[1] This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Grant No. F30602-97-2-0338 and by the US Air Force/OSR under Grant No. F49620-00-1-0330.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **15 MAR 2002**	2. REPORT TYPE	3. DATES COVERED **00-03-2002 to 00-03-2002**

4. TITLE AND SUBTITLE **Distributed Dynamic Channel Access Scheduling for Ad Hoc Networks**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**34**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

1. INTRODUCTION

Channel access schemes for ad hoc networks can be contention-based or scheduled. The advantage of contention-based schemes is that they are relatively easy to deploy; this has resulted in many contention-based schemes for ad hoc networks being proposed based on carrier sense multiple access with collision avoidance (CSMA/CA), and the success of the IEEE 801.11(b) standard for wireless local area networks [9]. Collision-avoidance schemes are attractive for ad hoc networks, because they attempt to eliminate collisions of data packets, which degrade network performance. However, collision-avoidance schemes cannot prevent collisions of signal packets from near-far phenomena, fading, and capture effects on the channel [14, 16]. In addition, it is difficult to provide quality of service or fairness with these channel access schemes. This points to the need for channel access methods based on scheduling.

Scheduled access schemes prearrange or negotiate a set of timetables for individual nodes or links, such that the transmissions from these nodes or on these links are collision-free within the effective range of the transmissions on the time and frequency axes. TDMA, FDMA, CDMA, SDMA, and their combinations are widely deployed in cellular systems [2, 13]. However, these solutions require a central base station, and the peer-to-peer scheduling needed in ad hoc networks is much harder to solve.

The quest for optimal solutions to channel access scheduling in ad hoc networks (i.e., multihop packet radio networks) often results in NP-hard problems in graph theory (such as k-colorability on nodes or edges) [10, 11, 24]. In some cases, however, the problems can be solved by reducing them to simpler ones for which polynomial algorithms are known to achieve suboptimal solutions using randomized approaches or heuristics based on such graph attributes as the degree of the nodes.

Many solutions have been proposed combining both random and scheduled access approaches [6, 7, 27]. Specifically, a few time slot assignment algorithms were presented by Cidon and Sidi [8], and Pond and Li [22] using a dedicated control segment of the channel to resolve conflicts and broadcast channel reservations. However, the complex resolution of neighbor schedules via message exchanges in the channel consume a considerable portion of the scarce bandwidth and introduce long delays to obtain the correct schedule. Several channel scheduling and reservation protocols have been proposed based on in-band signaling (phased dialogs or RTS/CTS handshakes) before transmissions [28, 30] to secure a temporary schedule for channel access. Because of the in-band signaling required, these protocols suffer from unused time slots when signals collide because of their randomness.

Topology-transparent scheduling methods have been proposed by Chlamtac and others [5, 19] to avoid the need for the in-band signaling of the above "topology-dependent" schemes. The basic idea of the topology-transparent scheduling approach is for a node to transmit in a number of time slots in each time frame. The times (slots) when node i transmits in a frame corresponds to a unique code, such that for any given neighbor k of i, node i has at least one transmission slot during which k and none of k's own neighbors are transmitting. Therefore, within any given time frame, any neighbor of i can receive at least one packet from i collision-free. The limitation of the topology-independent scheduling approaches described to date is that the sender is unable to know which neighbor(s) can correctly receive the packet it sends in a particular slot. This implies that the sender has to send its packet in the various slots in a frame, making the frame length (number of slots) much larger than the number of nodes in a two-hop neighborhood and dependent on the network size, which is less scalable.

A unified framework for static channel assignment in time, frequency, and code division multiple access called UxDMA was described by Ramanathan [23] to compute a k-coloring of an arbitrary graph within polynomial steps. The heuristic consists of first coloring nodes or edges randomly or sequentially according to vertex degrees, and then conclude with a minimum number of colors, such that a set of constraints on the nodes or links are satisfied. The constraints on the coloring pattern include commonly known interferences, such as direct and hidden-terminal interferences [29]. A limitation of this and similar schemes based on k-colorings of graphs is that, inherently, topology information needs to be collected and frequent schedule broadcasts have to be carried out in dynamic networks, which would consume a significant portion of the scarce wireless bandwidth.

To avoid the repetitious schedule adjustments or redundant multiple transmissions of data packets due to the volatility of wireless network topologies, we propose that local topology be an integral ingredient of the channel-access scheduling for each node of an ad hoc network.

Section 2 shows that the scheduling problems for node-activation and link-activation channel access can be approached as a 2-coloring problem on graphs. It presents a new contention resolution algorithm called neighborhood-aware contention resolution (NCR) which works by each node maintaining the identifiers of its one- and two-hop neighbors, and making a new node or link activation decision during each contention context (e.g., each time slot). Section 3 addresses the performance of NCR, its fairness, and its proper operation. Section 4 describes three channel access protocols based on node-activation and link activation-schemes. Section 5 discusses the neighbor protocol for handling mobility. Section 6 addresses the performance of these protocols by means of simulation experiments. Section 7 concludes the paper.

2. NEIGHBORHOOD-AWARE CONTENTION RESOLUTION

In multihop wireless networks, a single radio channel is spatially reused in different parts of the network, and contending entities are nodes or links (edges) between nodes. Collisions happen in three cases, as illustrated in FIG. 1 [26]. Nodes can avoid such collisions using topology information within two hops.

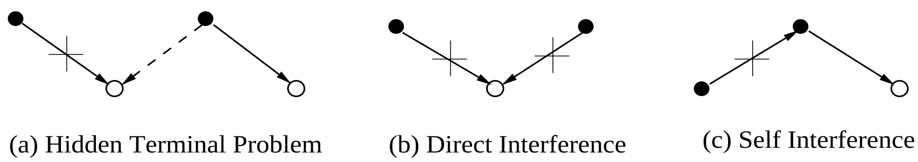

FIG. 1 Examples of Collision Types

We assume that every entity knows the set of its contenders through some appropriate means. One approach is for each node to broadcast periodically the identifiers of its one-hop neighbors, as described in Section 5. We also assume that each contention context is identifiable, which is reasonable in networks based on time-division multiple access or frequency hopping.

We formulate the problem of contention resolution with neighborhood information as follows:

> *Given a set of contenders, M_i, against an entity i in contention context t, how should the precedence of i be arbitrated in the set $M_i \cup \{i\}$, such that every contender yields to i whenever i decides that it is the winner for the common channel?*

To describe our solution to the problem, we assume that the primary operands in mathematical formulas are of fixed length, and the sign '$\oplus$' represents the concatenation operation on its operands. We also denote by $\texttt{MD}(x)$ the message digest function that produces a uniformly distributed random number using byte-string x as the seed [25]. Based on function $\texttt{MD}(x)$, we define the priority of node i in a contention context t as:

$$p_i^t = \texttt{MD}(i \oplus t) \oplus i \ . \tag{1}$$

The algorithm for neighbor-aware contention resolution (NCR) is the following:

NCR (entity i, contention context t):

1. Compute a priority p_k^t for each member k in set $M_i \cup \{i\}$:

$$p_k^t = \texttt{MD}(k \oplus t) \oplus k, k \in M_i \cup \{i\} \ . \tag{2}$$

2. Exit if Eq. (3) is not true.

$$\forall j \in M_i, p_i^t > p_j^t \tag{3}$$

3. Have i access the common channel during t. □

Note that, while the MD function can generate the same number on different inputs, each priority number is unique, because $p_k^t, k \in M_i \cup \{i\}$ is appended with k to the corresponding message digest.

Describing NCR in terms of a two-coloring problem, an entity i gives itself color r if its has the highest priority amongst its contenders in a contention context; otherwise, i colors itself with b. Nodes in color r are active in the corresponding contention context. The color r is extensively used in each contention situation to the maximal degree without collisions.

The description of NCR provided thus far assumes that each node requires the same amount of bandwidth. In practice, traffic and bandwidth demands from different nodes can vary. NCR can be easily improved to accommodate variable bandwidth requirements by assigning multiple *pseudo identities* to each entity.

An entity i may claim $pi_i \geq 0$ pseudo identities, and each pseudo identity of i is defined as the concatenation of i with a number chosen from $\{1, \ \ldots, pi_i\}$. Therefore, the l-th pseudo identity is denoted as $i \oplus l$. To account for stability of bandwidth requests, an upper bound can be placed on the number of pseudo identities available to each entity, so as to allow reasonable granularity of bandwidth allocation.

Consequently, NCR can be modified as follows to support multiple identities for each node:

NCR-MI (entity i, contention context t):

1. Compute the priority numbers on the pseudo identities of each member $k \in M_i \cup \{i\}$, the l-th priority number of which is denoted as $p^t_{k \oplus l}$:

$$\begin{aligned} p^t_{k\oplus l} &= \mathtt{MD}(k \oplus l \oplus t) \oplus k \oplus l, \\ &\quad k \in M_i \cup \{i\}, 1 \leq l \leq pi_k \end{aligned} \tag{4}$$

2. Exit if Eq. (5) is not true.

$$\begin{aligned} &\forall j \in M_i, \exists\, m,\ p^t_{i\oplus m} > p^t_{j\oplus n}, \\ &1 < m < pi_i, 1 < n < pi_j. \end{aligned} \tag{5}$$

3. Have i access the common channel during t. $\square$

The portion of the common channel available to an entity i is

$$q_i = \frac{pi_i}{\sum_{k \in M_i \cup \{i\}} pi_k}. \tag{6}$$

Note that NCR is the special case of NCR-MI with the restriction $\forall k \in M_i \cup \{i\}, pi_k = 1$. For simplicity, the rest of this paper addresses only NCR.

3. BEHAVIOR OF NCR

3.1. Correctness

Once the nodes in an ad hoc network have consistent knowledge of their two-hop neighborhood, NCR achieves the following three goals:

1. Avoid unintentional collisions from simultaneous transmissions.

2. Fair sharing of network bandwidth for each node, so as to avoid the resource starvation problem present in contention-based schemes.

3. Permit constant bandwidth utilization, even under heavy traffic load, so as to keep network data transmission live at all times.

Because it is assumed that contenders have mutual knowledge and t is synchronized, the order of contenders based on the priority numbers is consistent at every participant. When entity i has the highest priority in the set $M_i \cup \{i\}$, each $k \in M_i$ respects the right of i, and allows i to access the common channel collision-free.

NCR basically generates a *permutation* of the contending members, the order of which is decided by the priorities of all participants. Since the priority is a pseudo-random number generated from a seed that changes from time to time, the permutation also becomes random such that i has certain probability to win in each contention context, commensurate to its contention level:

$$q_i = \frac{1}{|M_i \cup \{i\}|} \ . \tag{7}$$

An ad hoc network has a finite number of entities; therefore, NCR always produces one or multiple winners for each contention context because NCR gives a unique priority number to each entity and multiple locally maximal priorities exist in the network. Accordingly, NCR allows live utilization of the common channel.

3.2. Delay And Throughput Analysis

When the arrival rate of the queuing system in a channel access scheduling system is bellow the service rate, we can analyze the delay properties of the queuing system using a steady-state M/G/1 queue with server vacations, where the single server is an entity (node or link).

We assume that data packets arrive at an entity i according to a Poisson process with rate λ_i and are served by first-come-first-serve (FIFO) strategy. Server i takes a vacation that lasts for Z of length one time slot when there is no data packet in the queue; otherwise, i looks for the next available time slot to transmit the first packet waiting in the queue. Because of the randomness in NCR and NCR-MI, the number of time slots that an entity must wait before activation is a geometric distribution with parameter q_i, which is the probability of the entity i winning a contention context (Eqs. (7) and (6)). Therefore, the probability of the service time X_i for a data packet follows $P\{X_i = k\} = (1-q_i)^{k-1} q_i$.

The mean and second moments of random variable X_i are:

$$\overline{X_i} = \frac{1}{q} \ , \ \overline{X_i^2} = \frac{2-q}{q^2} \ .$$

And the mean and second moments of random variable Z (a constant) are: $\overline{Z} = \overline{Z^2} = 1$.

So that the extended Pollaczek-Kinchin formula [4] for M/G/1 system with vacations readily yields the average waiting time in the queue at entity i:

$$W_i = \frac{\lambda \overline{X_i^2}}{2(1-\lambda\overline{X_i})} + \frac{\overline{V^2}}{2\overline{V}}$$

Adding the average service time to the queuing delay, we obtain the overall delay in the system:

$$T_i = W_i + \overline{X_i} = \frac{2 + q_i - 2\lambda}{2(q_i - \lambda)} \ . \tag{8}$$

When $\lambda_i = 0$, the least expected system process latency is:

$$T_i = 1/q_i + 0.5 \tag{9}$$

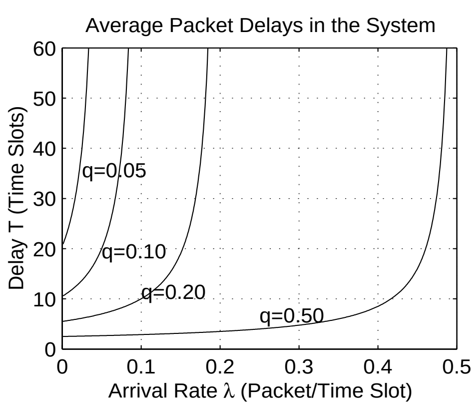

FIG. 2 Average System Delay of Packets

Depending on whether the entity is a node or a link, the probabilities of the entity winning a contention context are different, so are the delays of data packets going through that entity. FIG. 2 shows the average delay of a packet in the queuing system at an entity i with different channel access probability q_i and arrival rate λ_i. To keep the queuing system in a steady state, it is necessary that $\lambda_i < q_i$.

Because of the collision freedom of NCR, the common channel can serve certain load up to the maximum channel capacity. That is, the throughput over the common channel is the summation of arrival rates at all competing entities as long as the queuing system at each entity remains in equilibrium on the arrival and departure events. Accordingly, the system throughput S from each and every entity k that competes for the common channel is:

$$S = \sum_k \min(\lambda_k, q_k) \tag{10}$$

where q_k is the probability that k may be activated, and λ_k is the data packet arrival rate at k.

4. CHANNEL ACCESS PROTOCOLS

4.1. Topology Assumptions

For simplicity, we abstract the topology of a packet radio network as an undirected graph $G = (V, E)$, where V is the set of nodes, each mounted with an omnidirectional radio transceiver and assigned a unique identifier (ID), and $E \subseteq V \times V$ is the set of links between nodes. Unless specified otherwise, a link $(u, v) \in E$ indicates node u and v are within the transmission range for each other, so that they can exchange packets via the common channel, in which case the two nodes are called *one-hop neighbor*s. Two distinct nodes that are not one-hop neighbors but share a common one-hop neighbor are called *two-hop neighbor*s to each other. The set of one-hop and two-hop neighbors of node i is denoted by N_i^1 and N_i^2, respectively.

Contentions at a node i should be resolved on the subgraph covered within two hops from the node i, i.e., $N_i^1 \cup N_i^2$, depending on node or link activation schemes and signal coding methods used in the specific channel access protocols.

The following channel access protocols are described assuming that nodes already know their neighborhood, i.e., that they have exchanged the necessary information about their two-hop neighborhood. In addition, these protocols are based on a distributed time division multiplexing scheme, which makes channel access decisions based on the time slot boundaries. However, we do not address the time synchronization issue.

4.2. Node Activation Protocol

We first describe a node-activation multiple access (NAMA) protocol, in which the contender set of node i is the set of one-hop and two-hop neighbors, i.e., $M_i = N_i^1 \cup N_i^2$. For each time slot t, NAMA decides the activation of node i based on following algorithm:

NAMA (node i, time slot t):

1. Compute the priority p_k^t of every node k in the set $M_i \cup \{i\}$ using Eq. (2).
2. Exit if Eq. (3) does not hold for node i.
3. Have i access the common channel during time slot t. □

NAMA is suitable for broadcast and multicast services in ad hoc networks, where the receivers of a packet transmission include all or some of the transmitter one-hop neighbors. This is because each activated node has the highest priority among the two-hop neighborhood of the node, and is able to deliver a packet to all its one-hop neighbors without interference from its two-hop neighbor.

4.3. Link Activation Protocol

The link activation multiple access (LAMA) protocol is a time-slotted code division medium access scheme using direct sequence spread spectrum (DSSS), together with NCR.

In DSSS, code assignment can be based on a transmitter-oriented, a receiver-oriented or a per-link oriented coding scheme [15, 18, 21]. In LAMA, we opt for a receiver-oriented code assignment, which is suitable for unicasting using a link-activation scheme. Although many collision resolution protocols have relied on code assignment algorithms to eliminate packet collisions [3, 17]. In contrast, the code assignment for LAMA is random and does not consider contention resolution. Instead, the contentions for transmission on the code of the intended receiver are resolved by additional computations using NCR.

We assume that a pool of well-chosen quasi-orthogonal pseudo-noise codes, the set of which is denoted as $C_{pn} = \{c^k\}$, are available for each node to choose from, where $k = 0, \ \ldots, |C_{pn}| - 1$. A receiver i is assigned a pseudo-noise code c_i from C_{pn} by the following hashing function, which utilizes the message digest function used in Eq. (1):

$$c_i = c^k, \ k = \mathtt{MD}(i \oplus t) \ \mathtt{mod} \ |C_{pn}| \tag{11}$$

Because we have only a limited number of pseudo-noise codes to assign, it is possible for multiple nodes to share the same code. If we denote the set of i's one-hop neighbors assigned with code c as $n^1_{i,c}$, our goal in LAMA is to decide whether node i can activate a link on a code c and send a packet to one of the receivers in $n_{i,c}^1$ during time slot t. Therefore, the set of contenders to node i includes the one-hop neighbors of i and the one-hop neighbors of nodes in the set $n_{i,c}^1$ excluding node i itself, as shown in Eq. (12).

$$M_i = N_i^1 \cup \left(\bigcup_{k \in n_{i,c}^1} N_k^1 \right) - \{i\}. \tag{12}$$

The resulting link activation algorithm is the following:

LAMA (node i, time slot t, code c):

1. Compute the priority p_k^t of every node $k \in M_i \cup \{i\}$ using Eq. (2).

2. If Eq. (3) holds, have i activate link $(i, j), j \in n_{i,c}^1$ in time slot t. □

LAMA ensures that the transmissions on a given code in any time slot are always collision-free at the receivers. Because multiple receivers may be waiting on the code, the transmitter can choose to deliver multiple packets or multicast packets to the receiving neighbors.

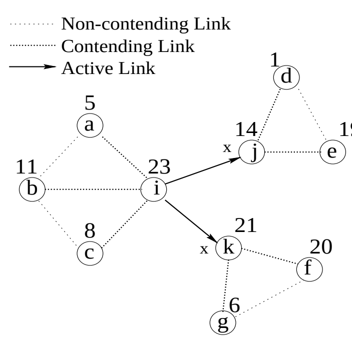

FIG. 3 An Example of Contending Resolution

FIG. 3 exemplifies a contention situation at node i during time slot t. The topology is an undirected graph. The number beside each node represents the current priority of the node. Node j and k happen to have the same code x.

To determine if node i can activate links on code x, we compare the priorities of the nodes according to LAMA. Node i has the highest priority within one-hop neighbors, and higher priority than j and k as well as their one-hop neighbors. Therefore, i can activate either (i, j) or (i, k) in the current time slot t, depending on back-logged data flows at i. In addition, node e may activate link (e, d) if d is assigned a code other than code x.

4.4. Pairwise Link Activation Protocol

The pairwise-link activation multiple access (PAMA) protocol is based on link activation, using a time-slotted code-division multiplexing scheme with DSSS. The difference between PAMA and LAMA is that a code is assigned for a given transmitter-receiver pair, and computed every time slot. Unlike NAMA and LAMA, in which contending entities are nodes, links are the entities competing for channel access in PAMA. Links are directed in PAMA to signify transmission directions. Each undirected physical link is represented by two directed links in the opposite directions.

PAMA assumes a pool of quasi-orthogonal pseudo-noise codes, $C_{pn} = \{c^k\}$. A pseudo-noise code c_u from C_{pn} is assigned to a directional link (u, v) at time slot t according to the following hashing function:

$$c_{(u,v)} = c^k, \; k = \mathtt{MD}(u \oplus t) \; \mathtt{mod} \; |C_{pn}|. \tag{13}$$

Note that Eq. (13) does not involve v in the code assignment. This is because of two reasons: (a) a node can activate only one link at a time, and (b) other nodes do not have to know both u and v to decide the code assignment of link (u, v).

PAMA decides whether a directed link (u, v) can be activated by node u in time slot t using the two-hop neighbor information. The set of contenders to link (u, v)

are the incident links of u and v excluding (u,v) itself, i.e.,

$$\begin{aligned} M_{(u,v)} = \quad & \{(x,y) \mid (x,y) \in E, x \in \{u,v\}\ \} \cup \\ & \{(x,y) \mid (x,y) \in E, y \in \{u,v\}\ \} - \{(u,v)\}. \end{aligned}$$

The resulting link activation algorithm is the following:

PAMA (link (u,v), time slot t):

1. Compute the priority $p^t_{(x,y)}$ of every link (x,y) belonging to $M_{(u,v)} \cup \{(u,v)\}$ using (14):

$$p^t_{(x,y)} = \mathtt{MD}(x \oplus y \oplus t) \tag{14}$$

2. Exit if Eq. (15) is not true.

$$\forall (x,y) \in M_{(u,v)},\ p^t_{(u,v)} > p^t_{(x,y)} \tag{15}$$

3. Compute the priorities of every incident (incoming and outgoing) link of u's one-hop neighbors using (14). Introduce set L_u to be the subset of incoming links of u's one-hop neighbors, each element of which has the highest priority among the incident links of the corresponding one-hop neighbor. That is:

$$\begin{aligned} L_u = \{(x,y) \mid y \in N^1_u,\ y \neq v, \texttt{and}\ x \in N^1_y,\ x \neq u, \\ \texttt{and}\ \forall z \in N^1_y,\ p^t_{(x,y)} > p^t_{(z,y)}\ \texttt{and}\ p^t_{(x,y)} > p^t_{(y,z)}\} \end{aligned} \tag{16}$$

4. For each link $(x,y) \in L_u$, compute the code assignment c_x as well as c_u using (13). If condition $c_u = c_x$ holds for link (x,y), then exit if:

   (a) $x \in N^1_u$ and link (x,y) has the highest priority among the incident links of node x;

(b) $x \notin N_u^1$.

5. Have node u activate link (u, v) during t. □

The first two steps in PAMA determine the eligibility of link (u, v) for activation, while step 3 and 4 avoid possible hidden terminal conflicts on u's one-hop neighbors. Step 3 chooses the candidate incoming link of each one-hop neighbor for activation, and then step 4 tries to avoid interference to u's one-hop neighbors if link (u, v) is ever activated. The transmitter is the one that tries to avoid collisions on its one-hop neighbors. Two conditions are considered in step 4. Case 4a occurs when the end points of the active incoming link are both one-hop neighbors of u, such that u has complete knowledge about their contention situation. Case 4b occurs when u knows only a partial contention situation of the active incoming link, such that u gives up (u, v) activation if only the active incoming link has the same code assignment.

PAMA is suitable only for unicast packet transmissions, because the link activations are purely based on pairs of nodes in the network.

FIG. 4 shows a sample network, where the number next to each link is the priority of that link, and the number beside each node is the code assignment of that node. Link (a, b) and (u, v) (indicated by solid lines) are both candidates for activation on code 5 according to step 2. However, step 4 deactivates u to avoid a collision at b by u and a.

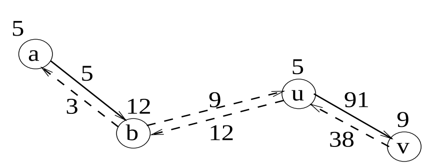

FIG. 4 Collision Avoidance in PAMA

5. NEIGHBOR PROTOCOL

In mobile ad hoc networks, the two-hop neighbor information needed by topology-dependent scheduling protocols is not readily available to each node. Because no neighbor information can be assumed to schedule the exchange of neighbor information, the neighbor protocol utilizes a random access approach for the transmissions of information. Due to the broadcast nature of NAMA, the neighbor protocol can also take advantage of the data packet transmissions to propagate the neighbor information. For simplicity, we only discuss the neighbor protocol for NAMA.

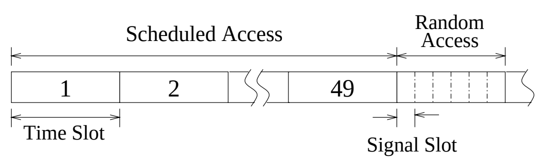

FIG. 5 Time Division Scheme in NAMA

As FIG. 5 illustrates, a special periodic time slot is allocated after every a number of scheduled-access time slots for sending out signals. The number of regular data slots and the special signal slot amount to fifty in the figure, which comprise a period in the time division scheme. The signal slot is further divided into multiple mini-slots for signals.

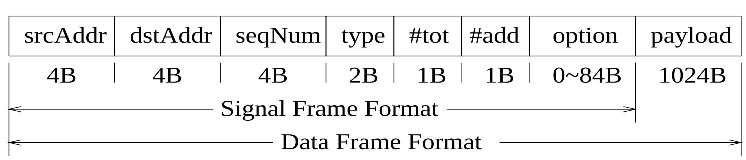

FIG. 6 Formats for Signal and Data Frames in NAMA

FIG. 6 shows the formats of a regular data frame and a signal frame, adopted in the implementation of NAMA. A signal is a special data frame that has every field of a data packet but the payload. The field *srcAddr* and *dstAddr* contain the source and destination addresses (each requires four bytes – 4B). The field *seqNum* and *type* provide the current packet sequence number and packet type, which indicates unicast, broadcast or signal data type. The field *#tot* is used by the neighbor protocol to suggest the total number of neighbor updates in the *option* field, of which *#add* updates are the added or refreshed neighbors.

Assuming that the bandwidth of the channel is 2 Mbps [1], and the sizes of different fields are as in FIG. 6, the signal slot can contain six signal mini-slots, accounting for both radio propagation latency and signal processing latencies.

Signals are used by the neighbor protocol for two purposes. One is for a node to say "hello" to its one-hop neighbors periodically to maintain connectivity. The other is to send neighbor updates when a neighbor is added or deleted.

A node waits for some mini-slots before transmitting its signals, so that the probability of collisions with other neighbors in a mini-slot is reduced. The interval between sending signals in the signal slots is determined by the number of two-hop neighbors. We consider two-hop instead of one-hop neighbors, because their transmissions can all collide with that of a given node. This situation was formulated as an occupancy problem in combinatorial mathematics [12, 20], which pursues the probability of having m empty cells after randomly placing r balls into n cells, where r corresponds to the interval, and n corresponds to the number of two-hop neighbors of a node. We directly use the result on the probability of leaving exactly m cells empty, which is:

$$p_m(r,n) = n^{-r} \begin{pmatrix} \text{n} \\ \text{m} \end{pmatrix} \sum_{v=0}^{n-m} (-1)^v \begin{pmatrix} \text{n-m} \\ \text{v} \end{pmatrix} (n-m-v)^r \qquad (17)$$

Given the average number of two-hop neighbors n in a network, we search for such an r that $p_0(r,n) > 0.99$, which promises 99% probability of having every two-hop neighbor transmit at lease once during the pursued interval collision-free.

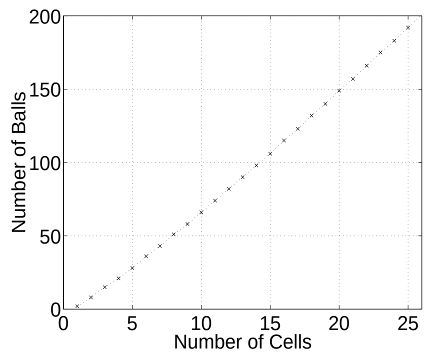

FIG. 7 Number of Balls vs. Number of Cells Such That $p_0(r, n) > 0.99$

FIG. 7 shows the minimum numbers of balls (interval) to allow $p_0(r, n) > 0.99$, given different numbers of cells (two-hop neighbors). In practice, we set the interval to 150, which allows about 20 neighbors within two hops for each node. The 150 signal mini-slot interval corresponds to a time period of about 5.6 seconds between consecutive "hello" signals from the same node.

In addition, a jitter of ± 25 mini-slots is added to the interval number to avoid signal transmission synchronization.

Because of NAMA's broadcasting capability, regular data frames in the scheduled access time slots also have the same fields to propagate neighbor information, as shown in FIG. 6.

Neighbor updates are generated in the following three situations:

1. A new neighbor is detected at a node and the whole one-hop neighbor set of the node needs to be propagated to the new neighbor.

2. The one-hop neighbor set is refreshed, which occurs periodically.

3. A neighbor is lost after a period of silence from that neighbor, and a neighbor deletion update is sent.

In FIG. 6, the difference between field *#tot* and *#add* indicates the number of deletion updates in the *option* field. A two-hop neighbor can be deleted at a node if the node does not hear update about the two-hop neighbor from the common one-hop neighbor for a certain period of time.

6. PERFORMANCE EVALUATION

6.1. Expected Performance Differences

In NAMA, contending members are nodes within two hops. Therefore, the average number of contending nodes in each time slot is

$$|N_i^1 \cup N_i^2|$$

In LAMA, contentions happen on each code. When a node i tries to transmit on a code to one of its neighbors, contention occurs from both i's one-hop neighbors and the one-hop neighbors of the receivers possessing the code. Hence, the average number of contenders to i on a code c is:

$$\left| N_i^1 \cup \left(\bigcup_{j \in n_{i,c}^1} N_j^1 \right) \right| - 1$$

according to Eq. (12).

PAMA is more sensitive to neighbor connections than the other two protocols because two-hop neighbors as well as links between two-hop neighbors become the contention sources. The contenders of a link in PAMA are about twice as many as that of LAMA, because of the directional treatment of links in PAMA.

Above all, the density of packet radios placed in an ad-hoc network and the transmission range of the radios determine contention levels in these protocols. Suppose that the network nodes are uniformly distributed on an infinite plane with density ρ, and all nodes have the same effective transmission range r. A node in NAMA has approximately $4\rho\pi r^2 - 1$ contending nodes with regard to two-hop neighbors. In LAMA, a node would have around $2\rho\pi r^2 - 1$ contending nodes for activating a link, considering the two endpoints of the link, and assuming one-hop neighbors of the endpoints are always assigned distinctive codes. While in PAMA, the number of contending links of each link activation is $4\rho\pi r^2 - 2$, because of the directional treatment of links.

If we examine the number of active links when a node may transmit packet in the current time slot, NAMA can activate all of its incident links, and LAMA can activate a subset of its incident links, while PAMA can activate a single incident link at all times. In the case of unicasting, PAMA sustains the highest throughput, because of a better spatial reuse of the channel, as shown by the results of simulation experiments described subsequently.

To better understand the performance of dynamic scheduling protocols, we compare them with the optimal static scheduling algorithm UxDMA that uses global topology information [23].

The unified framework UxDMA defines a parametric algorithm to derive a coloring on a graph according to the network topology and the type of entities. The number of colors used on the graph indicates the efficiency of the algorithm. In a time division multiple access scheme, the number of colors utilized determines the length of a time frame, during which every entity is activated once in a time slots of the time frame.

A set of atomic constraints, which serves as input to the UxDMA algorithm, enumerates all kinds of node and link relations that may result in collisions if the related entities are assigned the same color and activated at the same time during channel access. Accordingly, we select an appropriate subset of the constraints for each of our scheduling protocols, as shown in Table 1, corresponding to UxDMA for NAMA, LAMA and PAMA, respectively.

TABLE 1
Constraint Sets Corresponding to NAMA, LAMA and PAMA

Protocol	Type of Entities	Constraint Set
UxDMA-NAMA	Node	$\{V_{tr}^0, V_{tt}^1\}$
UxDMA-LAMA	Link	$\{E_{rr}^0, E_{tr}^0, E_{tr}^1\}$
UxDMA-PAMA	Link	$\{E_{rr}^0, E_{tt}^0, E_{tr}^0, E_{tr}^1\}$

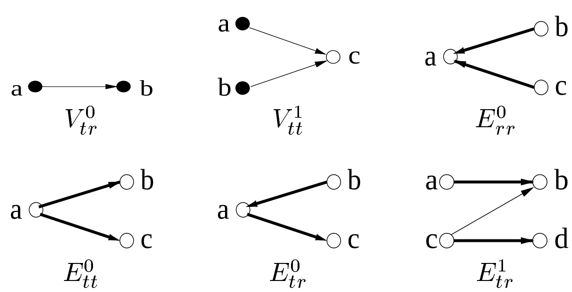

FIG. 8 Constraints Used by UxDMA for NAMA, LAMA and PAMA

FIG. 8 illustrates the prohibited schemes of node and link activations as referenced by UxDMA for NAMA, PAMA and LAMA, where solid dots and thick lines mean simultaneous node and link activations, respectively, thin lines indicate interferences. However, because both LAMA and PAMA use code-division channel access, constraint E_{tr}^1 is allowed when the *transmission* codes are different for node a and c in PAMA and *reception* codes are different for node b and d in LAMA.

An optimal ordering, PMNF (Progressive Minimum Neighbors First) heuristic, has been applied in each computation of the colorings on the graphs in UxDMA, so that the colorings "perform quite close to optimum" [23].

6.2. Simulation Results

We simulate NAMA, LAMA, PAMA and the corresponding UxDMA algorithms in static topologies. The performance of the protocols are studied in two scenarios: fully connected networks with different numbers of nodes, and multihop networks with different radio transmission ranges. The packet arrival and departure events are modeled as M/G/1 queuing systems with vacations. The delay of packets at each node and the throughput of the network are collected in each simulation.

The simulations are guided by the following parameters and behaviors:

- The signal propagation in the channel follows the free-space model and the effective range of radio is determined by the power level of the radio. All radios have the same transmission range.
- The bandwidth of a radio transmission is 2 Mbps.
- A time unit in the simulation equals one time slot. A time slot lasts 4.5 milliseconds including the guard time, which is long enough to transmit a 1124-byte packet.
- Each node has an unlimited buffer for data packets.
- In LAMA and PAMA, 30 pseudo-noise codes are available for code assignments, i.e., $|C_{pn}| = 30$.
- All nodes have the same packet arrival rate λ_i in each simulation. Unless otherwise specified, the destinations of the generated packets are evenly distributed on all outgoing links.
- Packets are served in First-In First-Out (FIFO) order.

- The duration of the simulation is 100000 time slots (equal to 450 seconds) slots), long enough to compute the metrics of interests.

6.2.1. Fully Connected Scenario:

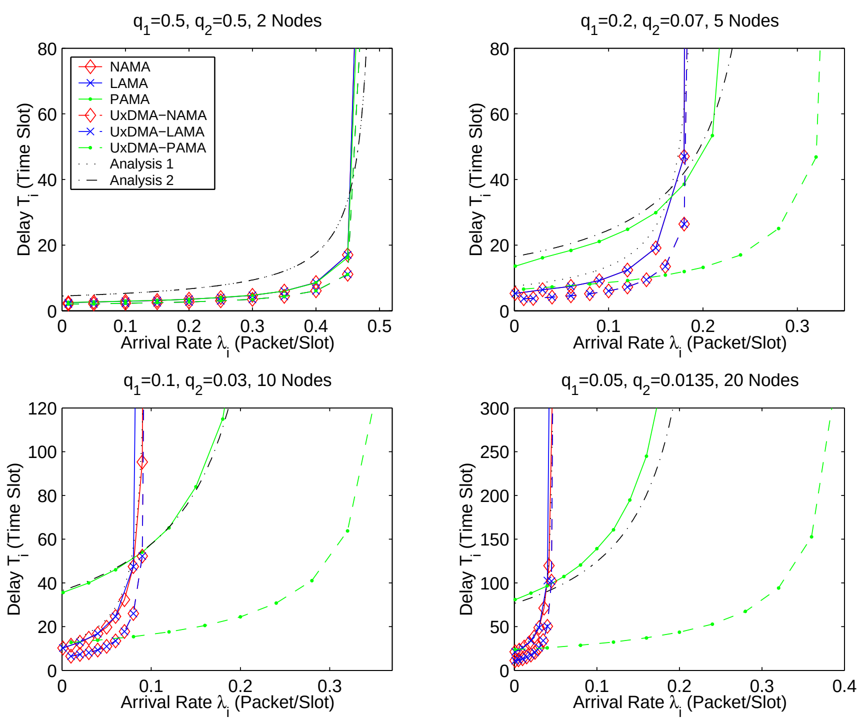

FIG. 9 Average Packet Delays In Fully-Connected Networks

In the fully connected scenario, simulations were carried out in four configurations: 2-, 5-, 10-, 20-node networks, to manifest the effects of different contention levels. FIG. 9 shows the delay values under different loads in the four cases as well as two analytical curves derived from Eq. (8) with q_1 values for NAMA and LAMA, q_2 values for PAMA as shown in the figures, respectively. All protocols appear to fit well with the analytical curves. PAMA shows higher delays in the same situations. This is because the contention sources are different in PAMA than in NAMA and LAMA. In PAMA, contending entities are links, which are much more than the number of nodes. The q_i value for PAMA in the fully connected scenario is:

$$q_i = \frac{1}{4 \cdot |V| - 6} \tag{18}$$

For NAMA and LAMA, q_i in the same scenario is:

$$q_i = \frac{1}{|V|} \tag{19}$$

Taking the 10-node network as an example, the q_i value for each link is $\frac{1}{4\times 10-6} = 0.029$ in PAMA, which would result in a delay of at least 36.5 time slots by Eq. (9). In cases of NAMA and LAMA, nodes are the contending entities, and the q_i values for each node are both around $\frac{1}{|V|} = \frac{1}{10}$=0.1, which leads to delays of at least 12.5 time slots.

In every simulation setting, UxDMA performs better than its counter protocol, NAMA, LAMA, and especially PAMA. This is because UxDMA can always produce a more compact schedule where a dynamic scheduling protocol may give up due to local priority comparisons. However, considering that UxDMA is a centralized algorithm with global topology information, NAMA, LAMA and PAMA trade off a little performance for much better efficiency.

FIG. 10 shows the throughput of the three protocols. As predicted in Eq. (10), all protocols show linear system throughput under the different sustainable loads and flat throughput when network load exceeds the available channel capacity, which is advantageous over any other randomized multiple access protocols that experience great loss in the throughput when the network load goes beyond certain point. The analytical curves in each subplot still fit well with respective protocols.

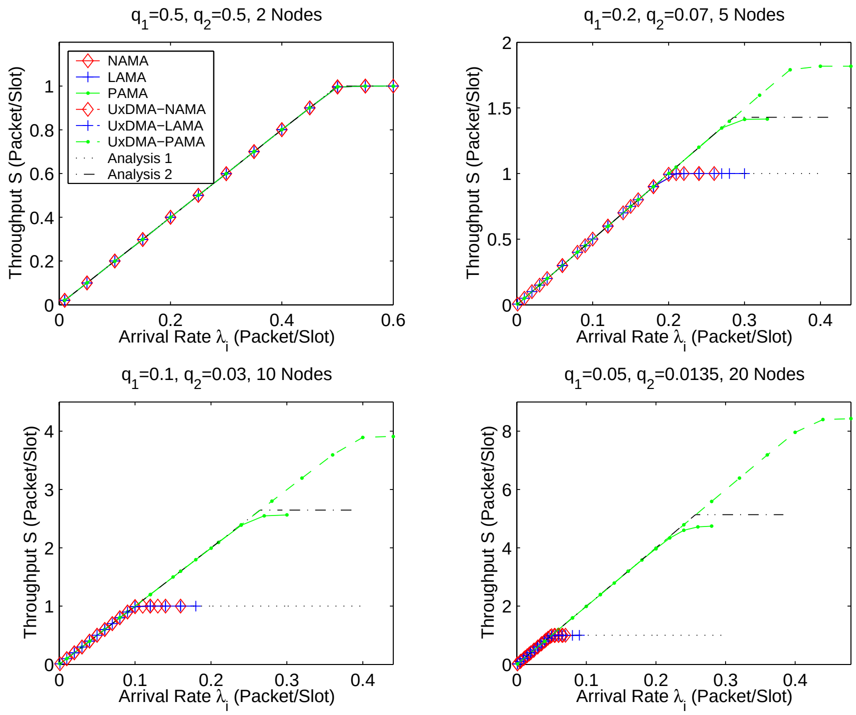

FIG. 10 Packet Throughput Of Fully-Connected Networks

6.2.2. Multihop Network Scenario:

FIG. 11 and 12 show the delay and throughput of the three protocols in multihop networks. The networks are generated by randomly placing 100 nodes within an area of 1000×1000 square meters. To simulate an infinite plane that has constant node placement density, the opposite sides of the square are seamed together, which visually turns the square area into a torus. By setting the transmission ranges of the transceiver on each node to 100, 200, 300 and 400 meters, respectively, we also change the topology and contention levels in each case.

FIG. 11 demonstrates the advantage of LAMA over NAMA, obtained from better channel reuse within two hops of each node by applying code division multiplexing in LAMA. PAMA produces higher delays than the other two protocols when network load is low. This is because of the same reasons discussed in fully-

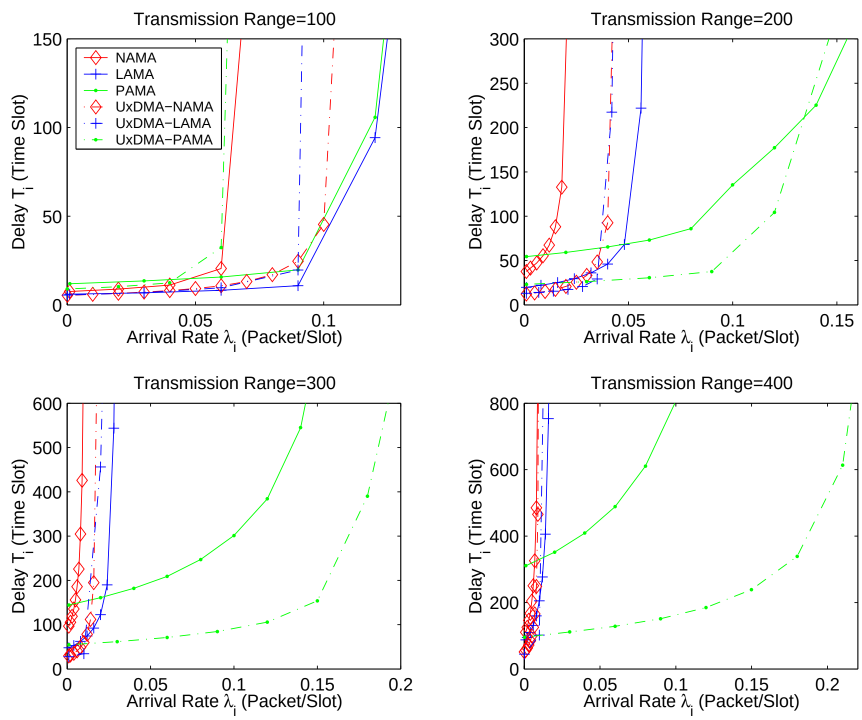

FIG. 11 Average Packet Delays In Multihop Networks

connected scenario. However, PAMA appears to have slower increases in delay when the network load increases, which explains the higher spectrum and spatial reuse of the common channel by pure link-oriented scheduling.

Because of the dramatic difference between the throughput of NAMA, LAMA and PAMA, FIG. 12 only shows the maximum throughput available in these individual protocols, instead of showing the gradual throughput variations in accordance with the load changes. Next to each bar of these dynamic scheduling protocols, the throughput of the corresponding UxDMA algorithm is also contrasted. Except for the first simulation when the transmission range is 100 meters, UxDMA performs better than NCR. This is because network connection densities show more variety at lower transmission range. While UxDMA schedules the network channel access according to the worst dense part of the network, NCR computes the channel access

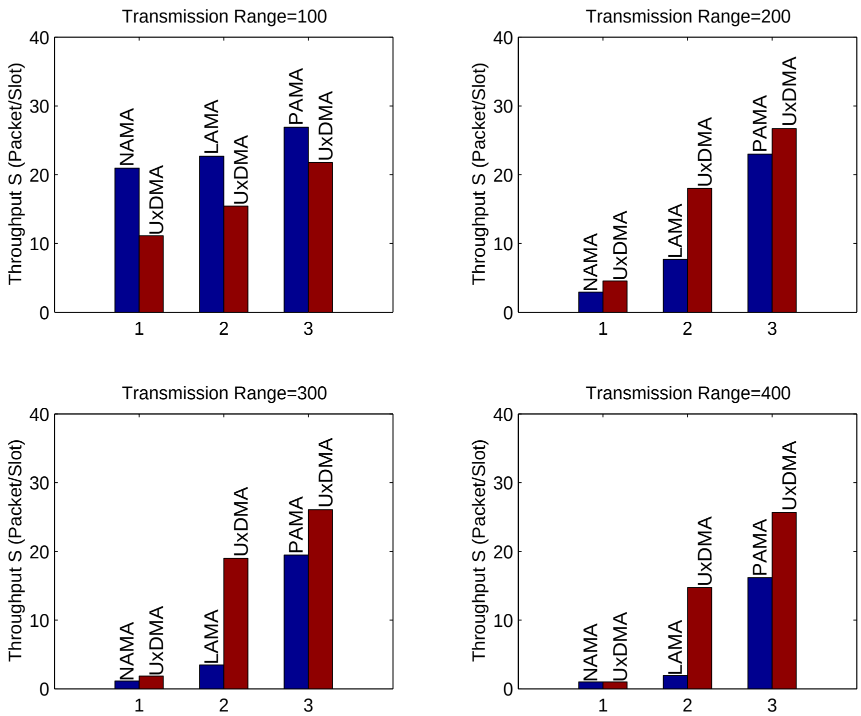

FIG. 12 Packet Throughput Of Multihop Networks

schedule according to local topology only.

Although not shown in the figure, the throughput of these protocols still demonstrates linear increases along with load increases, and levels off when the load values approximate or exceed the probabilities that a node and link may be activated, at which point delays increase drastically, as shown in FIG. 11. System throughput is an indication of the average channel reuse ratio in multihop wireless networks. PAMA achieves higher loads than the other two protocols.

7. CONCLUSION

We have introduced a new approach to contention resolution that eliminates much of the complexity of prior collision-free scheduling approaches by using only two-hop neighborhood information to dynamically determine which node is allowed to transmit in each collision-resolution context. Based on this approach, protocols were introduced for both node-activation and link-activation channel access scheduling in packet radio networks using time-division scheme. The advantages of the protocols are that (a) they do not need the contention phases or schedule broadcasts adopted by many other channel access scheduling algorithms; and (b) they only need the local topology information within two hops, which can be obtained by the propagation of one-hop neighbor information from each node to its neighbors. This contrasts with other schedule broadcasting algorithms that require complete network topology for collision-free channel access scheduling. NAMA is suitable for broadcasting and multicasting, while LAMA and PAMA are suitable for unicasting using spread spectrum techniques.

REFERENCES

[1] IEEE Std 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Technical report, IEEE, 1997.

[2] P.W. Baier. CDMA or TDMA? CDMA for GSM? In *5th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1280–4, The Hague, Netherlands, Sep. 18-23 1994. IOS Press.

[3] A.A. Bertossi and M.A. Bonuccelli. Code assignment for hidden terminal interference avoidance in multihop packet radio networks. In *IEEE INFOCOM '92*, volume 2, pages 701–9, Florence, Italy, 1992.

[4] D. Bertsekas and R. Gallager. *Data Networks, 2nd edition.* Prentice Hall, Englewood Cliffs, NJ, 1992.

[5] I. Chlamtac and A. Farago. Making transmission schedules immune to topology changes in multi-hop packet radio networks. *IEEE/ACM Transactions on Networking*, 2(1):23–9, Feb. 1994.

[6] I. Chlamtac and S. Kutten. A spatial-reuse TDMA/FDMA for mobile multi-hop radio networks. In *Proceedings IEEE INFOCOM*, pages 389–94, Washington, DC, Mar. 1985.

[7] I. Chlamtac and A. Lerner. Fair algorithms for maximal link activation in multihop radio networks. *IEEE Transactions on Communications*, 35(7):739–46, Jul. 1987.

[8] I. Cidon and M. Sidi. Distributed assignment algorithms for multihop packet radio networks. *IEEE Transactions on Computers*, 38(10):1353–61, Oct 1989.

[9] B.P. Crow, I. Widjaja, L.G. Kim, and P.T. Sakai. IEEE 802.11 Wireless Local Area Networks. *IEEE Communications Magazine*, 35(9):116–26, Sept 1997.

[10] A. Ephremides and T.V. Truong. Scheduling broadcasts in multihop radio networks. *IEEE Transactions on Communications*, 38(4):456–60, Apr. 1990.

[11] S. Even, O. Goldreich, S. Moran, and P. Tong. On the NP-completeness of certain network testing problems. *Networks*, 14(1):1–24, Mar. 1984.

[12] W. Feller. *An introduction to probability theory and its applications*, volume 1. New York: John Wiley, 2nd edition, 1957.

[13] W. Fuhrmann and A. Eizenhofer. Radio access protocol of the new GSM land mobile communication standard. In *38th IEEE Vehicular Technology Conference: Telecommunications Freedom - Technology on the Move*, pages 30–7, Philadelphia, PA, USA, Jun. 15-17 1988.

[14] C.L. Fullmer and J J. Garcia-Luna-Aceves. Solutions to hidden terminal problems in wireless networks. In *Proceedings ACM SIGCOMM*, Cannes, France, Sep. 1997.

[15] J.J. Garcia-Luna-Aceves and J. Raju. Distributed assignment of codes for multihop packet-radio networks. In *MILCOM 97 Proceedings*, pages 450–4, Monterey, CA, USA, Nov. 2-5 1997.

[16] J.J. Garcia-Luna-Aceves and A. Tzamaloukas. Reversing The Collision-Avoidance Handshake in Wireless Networks. In *Proc. ACM/IEEE Mobicom*, Seattle, Washington, Aug. 15–20 1999.

[17] L. Hu. Distributed code assignments for CDMA packet radio networks. *IEEE/ACM Transactions on Networking*, 1(6):668–77, Dec. 1993.

[18] M. Joa-Ng and I.T. Lu. Spread spectrum medium access protocol with collision avoidance in mobile ad-hoc wireless network. In *IEEE INFOCOM '99*, pages 776–83, New York, NY, USA, Mar. 21-25 1999.

[19] J.H. Ju and V.O.K. Li. An optimal topology-transparent scheduling method in multihop packet radio networks. *IEEE/ACM Transactions on Networking*, 6(3):298–306, Jun. 1998.

[20] R.J. Larsen and M.L. Marx. *An introduction to probability and its applications.* Englewood Cliffs, N.J. : Prentice-Hall, 1985.

[21] T. Makansi. Trasmitter-Oriented Code Assignment for Multihop Radio Networks. *IEEE Transactions on Communications*, 35(12):1379–82, Dec. 1987.

[22] L.C. Pond and V.O.K. Li. A distributed time-slot assignment protocol for mobile multi-hop broadcast packet radio networks. In *MILCOM 89*, volume 1, pages 70–4, Boston, MA, USA, 1989.

[23] S. Ramanathan. A unified framework and algorithm for channel assignment in wireless networks. *Wireless Networks*, 5(2):81–94, 1999.

[24] R. Ramaswami and K.K. Parhi. Distributed scheduling of broadcasts in a radio network. In *IEEE INFOCOM'89*, volume 2, pages 497–504, Ottawa, Ont., Canada, Apr. 23-27 1989. IEEE Comput. Soc. Press.

[25] R.L. Rivest. The MD4 message digest algorithm. In A.J. Menezes and S.A. Vanstone, editors, *Advances in Cryptology - CRYPTO '90 Proceedings*, pages 303–11, Santa Barbara, CA, USA, 11-15 Aug 1991. Berlin, Germany: Springer-Verlag.

[26] T.J. Shepard. A channel access scheme for large dense packet radio networks. In *ACM SIGCOMM '96 Conference*, pages 219–30, Stanford, CA, USA, Aug. 26-30 1996.

[27] J.A. Silvester. Perfect scheduling in multi-hop broadcast networks. In *Proceedings of the Sixth International Conference on Computer Communication*, pages 449–54, London, UK, Sep. 7-10 1982.

[28] Z. Tang and J.J. Garcia-Luna-Aceves. A Protocol for Topology-Dependent Transmission Scheduling. In *Proc. IEEE Wireless Communications and Networking Conference 1999 (WCNC 99)*, New Orleans, Louisiana, Sep. 21–24 1999.

[29] F. A. Tobagi and L. Kleinrock. Packet switching in radio channels: Part II - the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution. *IEEE Transactions on communications*, 23(12):1417–33, Dec. 1975.

[30] C. Zhu and M.S. Corson. A five-phase reservation protocol (FPRP) for mobile ad hoc networks. In *IEEE INFOCOM '98*, volume 1, pages 322–31, San Francisco, CA, USA, Mar. 29-Apr. 2 1998.