A New Approach to Channel Access Scheduling for Ad Hoc Networks [*]

Lichun Bao
Computer Science Department
University of California
Santa Cruz, CA 95064
baolc@soe.ucsc.edu

J.J. Garcia-Luna-Aceves
Computer Engineering Department
University of California
Santa Cruz, CA 95064
jj@soe.ucsc.edu

ABSTRACT

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 each node to elect deterministically one or multiple winners for channel access in a given contention context (e.g., a time slot), given the identifiers of its neighbors one and two hops away. The new protocols are shown to be fair and capable of achieving maximum 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.

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 802.11(b) standard for wireless local area networks [7]. 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 data packets resulting from near-far phenomena, fading, and capture effects on the channel [12, 14]. 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 in the time and frequency axes. TDMA, FDMA, CDMA, SDMA, and their combinations are widely deployed in cellular systems [1] [11]. 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) [8] [9] [22]. In some cases, however, the problems can be solved by reducing them to simpler cases 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 [4] [5] [24]. Specifically, a few time slot assignment algorithms were presented by Cidon and Sidi [6], and Pond and Li [20] 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 [25] [27] 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 [3] [17] 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

---

[*] This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Grant No. F30602-97-2-0338.

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 channel assignment in time, frequency, and code division multiple access called UxDMA was described by Ramanathan [21] to compute a k-coloring of an arbitrary graph within polynomial steps. The heuristic was to begin coloring nodes or edges randomly or sequentially according to vertex degrees, and 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 [26]. 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) by: (a) each node maintaining the identifiers of its one- and two-hop neighbors, and (b) 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, contending entities are nodes or links (edges) between nodes. We assume that every entity knows the set of its contenders by some appropriate means, such as each node periodically broadcasting the identifiers of its one-hop neighbors if contending entities are nodes, or the identities of links in its two-hop neighborhood if contending entities are links. We also assume that each contention context is identifiable, which is reasonable in networks based on a time-division multiple access or frequency hopping.

Given the knowledge of contenders for an entity i, the contention resolution algorithm must decide whether i is the winner in the contention context. The problem of contention resolution with neighborhood information can thus be stated 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 other contender yields to i whenever i derives itself as the winner for the common channel?*

To describe our solution to the problem, we assume that primary operants in mathematical formulas are of fixed length, and the sign '$\oplus$' lends to carrying out concatenation operation on its operants. During the contention context t, the solution to the problem is the following Neighborhood-aware Contention Resolution (NCR) algorithm:

NCR(contention context t):

1. Compute a priority p_k^t for each member k in set $M_i \cup \{i\}$:
$$p_k^t = \texttt{Rand}(k \oplus t) \oplus k, k \in M_i \cup \{i\} \qquad (1)$$
where function `Rand`(x) is a pseudo-random number generator that produces a uniformly distributed random number using the random-seed x.
2. Exit if Eq. (2) is not true.
$$\forall j \in M_i, p_i^t > p_j^t \qquad (2)$$
3. i may access the common channel during t. □

Note that, while the `Rand` function can generate the same number on different inputs, each priority number is unique since $p_k^t, k \in M_i \cup \{i\}$ is appended with k to the corresponding `Rand`$(k \oplus t)$.

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 demands at different nodes can vary, which requires different nodes to receive different amounts of bandwidth. Variable bandwidth requirements are easily accommodated in NCR by assigning multiple *pseudo identities* to each entity, with each entity being assigned up to a maximum of L_{pi} pseudo identities.

A pseudo identity of an entity is identified by the concatenation of the identifier assigned to the entity and a number identifying one of the one ore more pseudo identities assigned to the entity. If an entity i claims $pi_i \in [0, L_{pi}]$ pseudo identities, the l-th pseudo identity is denoted as $i \oplus l$, where $1 \leq l \leq pi_i$.

Consequently, NCR modified for multiple identities for each node (NCR-MI) is the following:

NCR-MI (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 $p^t_{k\oplus l}$:
$$\begin{aligned} p^t_{k\oplus l} = \ & \texttt{Rand}(k \oplus l \oplus t) \oplus k \oplus l, \\ & k \in M_i \cup \{i\}, 1 \le l \le pi_k \end{aligned} \tag{3}$$
2. Exit if Eq. (4) is not true.
$$\begin{aligned} & \forall j \in M_i, \exists\ m,\ p^t_{i\oplus m} > p^t_{j\oplus n}, \\ & 1 < m < pi_i, 1 < n < pi_j. \end{aligned} \tag{4}$$
3. i may access the common channel during t. □

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

$$q_i = \frac{pi_i}{\sum_{k\in M_i\cup\{i\}} pi_k}. \tag{5}$$

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 of 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. Allow 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, commensurate to its contention level,

$$q_i = \frac{1}{|M_i \cup \{i\}|} \tag{6}$$

to win in each contention context.

An ad hoc network has a finite number of entities; therefore, NCR always produces one or multiple winners for each contention context since 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 Performance

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/link).

We suppose 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 for V of 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 to wait before transmission is a geometric distribution with parameter $1 - q_i$, where q_i is the probability of the entity i winning a contention context (Eq. (6) and (5)). Therefore, the service time X_i for a data packet is $Y_i + 1$, where $P\{Y_i = k\} = q_i(1 - q_i)^{k-1}$.

The mean and second moments of random variable X_i are:

$$\overline{X_i} = \overline{Y_i} + 1 = \frac{1}{q_i}$$

$$\overline{X_i^2} = \overline{Y_i^2} + 2\overline{Y_i} + 1 = \frac{q_i^2 - 2q_i + 2}{q_i^2}$$

And the mean and second moments of random variable V are: $\overline{V} = \overline{V^2} = 1$.

So that the extended Pollaczek-Kinchin formula

$$W = \frac{\lambda\overline{X^2}}{2(1 - \lambda\overline{X})} + \frac{\overline{V^2}}{2\overline{V}},$$

for M/G/1 system with vacations readily yields the average waiting time in the queue at entity i:

$$W_i = \frac{\lambda_i(q_i^2 - 2q_i + 2)}{2q_i(q_i - \lambda_i)} + \frac{1}{2}$$

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

$$T_i = W_i + \overline{X_i} = \frac{\lambda_i q_i + 2(1 + q_i)}{2(q_i - \lambda_i)} + \frac{3}{2} \tag{7}$$

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

$$T_i = 1/q_i + 2.5 \tag{8}$$

Depending on whether the entity is a node or link, the probabilities of the entity winning a contention context are different, so are the delays of data packets going through that entity. Figure 1 shows the average delay of a packet in the

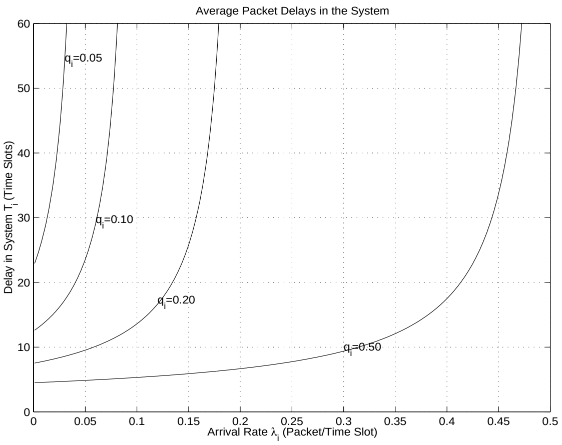


Figure 1: Average System Delay of Packets

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. We have the following system throughput S from each and every entity k that competes for the common channel:

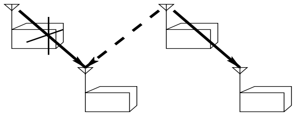
$$S = \sum_k \min(\lambda_k, q_k) \tag{9}$$

where q_k is the probability that k may access the common channel, and λ_k is the data packet arrival rate at k.

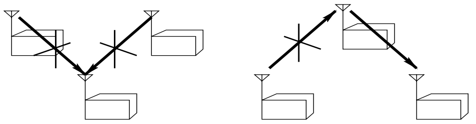
4. CHANNEL ACCESS PROTOCOLS

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

In multihop wireless networks, a single radio channel is spatially reused at different parts of the network. Collisions happen in three cases as illustrated in Figure 2 [23]. It is sufficient for collision-freedom if nodes within two hops do not transmit at the same time. Hence, contentions at a node i should be resolved on the subgraph derived from the two-hop neighbors of i, i.e., $N_i^1 \cup N_i^2$, depending on node/link activation schemes and signal coding methods as shown in the following protocols.



(a) Hidden Terminal Problem



(b) Direct Interference (c) Self Interference

Figure 2: Examples of Collision Types

The following channel access protocols are described assuming that nodes already know their neighborhood, i.e., they have exchanged the necessary information about their two-hop neighborhood.

4.1 Node Activation Protocol

We first present the NAMA (Node-Activation Multiple Access) protocol, which is based on NCR, node activation, and a distributed time division multiplexing scheme.

We do not address how nodes are time synchronized in this paper. This can be achieved by either: (a) listening to data traffic in the network, and aligning time slots to the latest starting point of a complete packet transmission by one-hop neighbors; or (b) other means, such as GPS (global positioning systems) timing signals.

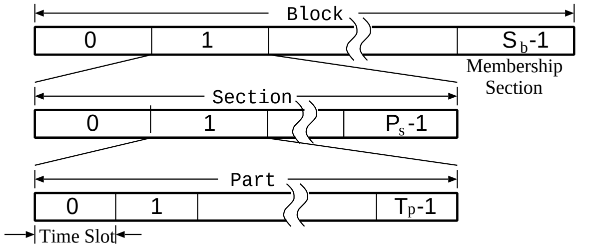


Figure 3: Time Division in NAMA

A time slot is the smallest time unit for transmitting one or more complete data packets. In NAMA, we impose more structures on time slots such that the combination of T_p consecutive time slots forms a *part*, P_s consecutive parts form a *section*, and S_b consecutive sections give the largest unit of time, *block*, as illustrated in Figure 3. Given the current time slot number t, we derive the current time slot number of a part, the current part and section numbers as follows:

$$\begin{array}{l} t' = t \;\texttt{mod}\; T_p \\ p' = (t/T_p) \;\texttt{mod}\; P_s \\ s' = [t/(T_p \times P_s)] \;\texttt{mod}\; S_b \end{array} \tag{10}$$

where `mod` is a modular operator, and all operants are integers.

A node i chooses only one part p_i, during which to contend for a time slot to transmit data packets. The choice of a part is dependent on the density of neighbors already using that part, usually decided when the node joins a network.

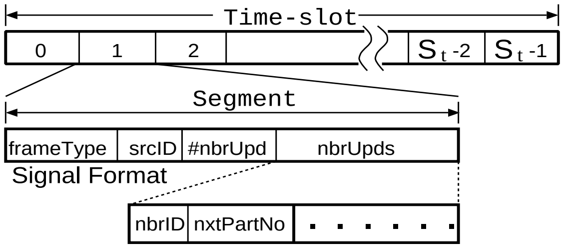


Figure 4: Signal Frame Format in Membership Section

For management purposes, the last section of a block is allocated for membership maintenance and is called *membership section*. New neighbors that did not transmit but listened in previous sections transmit signals in the membership section. For this purpose, time slots in the last section are further divided into S_t *segments* of equal duration for sending signals. Each signal contains the sender's ID and the part number that the node is willing to use in the coming blocks (Figure 4).

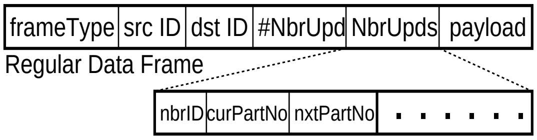


Figure 5: Data Frame Format in Regular Sections

In order to obtain two-hop neighbor information, every node broadcasts its one-hop neighbor IDs and corresponding part numbers whenever necessary. Portion of the header field of each data frame and signal frame is allocated for this purpose (Figure 4 and 5). Depending on the payload of a data frame, neighbors exchange their one-hop neighbor updates in a single or multiple data frames. The signal also contains as many as possible one-hop neighbor IDs and corresponding part numbers.

In NAMA, the contender set M_i of node i is a subset of $N_i^1 \cup N_i^2$, and changes from section to section in time as described in the following algorithm:

NAMA:

1. Compute the current part number p' according to Eq. (10).
2. Exit if $(p' \neq p_i)$ is true.
3. Compute the priority p_i^t using Eq. (1).
4. Assign node i to time slot $t_i = p_i^t \ \texttt{mod} \ T_p$.
5. Compute the current time slot t' in part p' using Eq. (10).
6. If $(t_i \neq t')$ then proceed to Step 10.
7. Compute the set of contending neighbors
   $$M_i = \{k \mid \ k \in N_i^1 \cup N_i^2 \text{ and } p_k = p' \text{ and } (p_k^t \ \texttt{mod} \ T_p) = t'\}$$
   where priority p_k^t is obtained from Eq. (1) for k, and p_k is the part number chosen by node k.
8. Exit if Eq. (2) does not hold for node i.
9. Access the common channel in current time slot t and exit.
10. Exit if
    $$\exists k, k \in N_i^1 \cup N_i^2 \text{ and } p_k = p' \text{ and } (p_k^t \ \texttt{mod} \ T_p) = t'.$$
11. The set of contending neighbors of node i now becomes:
    $$M_i = \{k \mid k \in N_i^1 \cup N_i^2 \text{ and } p_k = p'\}$$
    Compute another priority number $p_k^{t'}$ as follows:
    $$p_k^{t'} = \texttt{Rand}(k \oplus t \oplus t') \oplus k, \ \ k \in M_i \cup \{i\} \tag{11}$$
12. Exit if
    $$\exists j \in M_i, \ p_i^{t'} \not> p_j^{t'} \tag{12}$$
13. Access the common channel in time slot t. □

4.2 Link Activation Protocol

The LAMA (Link Activation Multiple Access) protocol is a time-slotted code division medium access scheme using direct sequence spread spectrum (DSSS) together with NCR.

In DSSS, code assignment can adopt transmitter-oriented, receiver-oriented or a per-link oriented coding schemes [13] [16] [19]. A channel access scheduling based on transmitter-oriented code assignment handles very much the same case as NAMA, because both approaches advocate the broadcast nature of transmission.

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 catered to code assignment algorithms to eliminate packet collisions [2] [15], the code assignment for LAMA is relatively static and random, and the contentions for transmission on the code of the intended receiver are resolved by other computations in LAMA.

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. The pseudo-noise codes inside C_{pn} are sorted according to their values: $c^0 < c^1 < \ldots < c^{|C_{pn}|-1}$. A receiver i is assigned a pseudo-noise code c_i from C_{pn} by the following hashing operation, which utilizes the pseudo-random number generator used in Eq. (1):

$$c_i = c^k, \ \ k = \texttt{Rand}(i) \ \texttt{mod} \ |C_{pn}| \tag{13}$$

LAMA establishes a channel access schedule for each individual time slot. Having the knowledge of one-hop neighbors is sufficient for a node to avoid collision of type (b) in Figure 2, and knowledge of its two-hop neighbors is enough to eliminate collision of type (a) and (c).

Because we have a limited number of pseudo-noise codes for assignment, it is possible that 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 packet to one of the receivers in $n^1_{i,c}$ during time slot t. Therefore, the set of contenders to node i includes one-hop neighbors of i and one-hop neighbors of nodes in the set $n^1_{i,c}$ excluding node i itself, as shown in the following formula:

$$M_i = N^1_i \cup \left(\bigcup_{k \in n^1_{i,c}} N^1_k \right) - \{i\}. \tag{14}$$

LAMA:

1. Compute the priority p^t_k of every node $k \in M_i \cup \{i\}$ using Eq. (1).
2. If Eq. (2) holds, then activate link $(i,j), j \in n^1_{i,c}$ in time slot t. $\square$

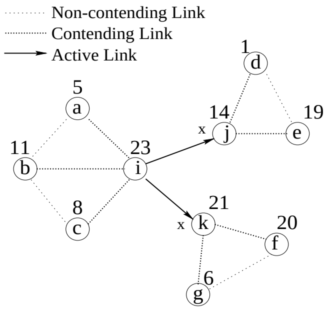


Figure 6: An Example of Contending Resolution

Figure 6 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 i can activate links on code x, we compare priorities of 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.

4.3 Pairwise Link Activation Protocol

The PAMA (Pairwise-link Activation Multiple Access) protocol is also a time-slotted link activation protocol based on a code division multiplexing scheme using DSSS. The difference with LAMA is that a code is assigned for a given transmitter-receiver pair, and computed every time-slot, so that the contention situation is different from time slot to 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 link is represented by two directed links in opposite directions.

As in LAMA, we assume 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:

$$c_u = c^k, \ k = \texttt{Rand}(u \oplus t) \oplus u \ \texttt{mod} \ |C_{pn}|. \tag{15}$$

Note that it is unnecessary to involve v in the code assignment, because of the simple fact that a node can activate only one link at a time.

Like LAMA, the two-hop neighbor information is presumed to be available in PAMA by the appropriate integration of NAMA and PAMA.

PAMA decides whether a directed link (u,v) can be activated by node u in time slot t. The set of contenders to link (u,v) are the incident links of u and v excluding (u,v) itself, i.e.,

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

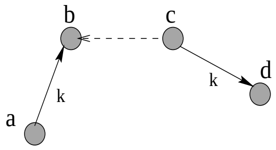


Figure 7: An Example of Hidden Terminal Problem In PAMA

When a link (u,v) is activated, there are possible hidden terminal conflicts from one-hop neighbors of v if any outgoing link on the one-hop neighbors of node v is assigned the same code as (u,v). Figure 7 illustrates that a collision happens at node b when link (a,b) and (c,d) are activated using the same code k. PAMA is able to deactivate link (a,b) for the current time slot as described in the following algorithm:

PAMA:

1. Compute the priority $p^t_{(x,y)}$ of every link $(x,y) \in M_{(u,v)} \cup \{(u,v)\}$ using Eq. (16):
$$p^t_{(x,y)} = \texttt{Rand}(x \oplus y \oplus t) \oplus x \oplus y \tag{16}$$
2. Exit if Eq. (17) does not hold.
$$\forall (x,y) \in M_{(u,v)}, \ p^t_{(u,v)} > p^t_{(x,y)} \tag{17}$$
3. Compute the priority of each two-hop neighbor of node u, that is:
$$p^t_k = \texttt{Rand}(k \oplus t) \oplus k, \ k \in N^2_u \tag{18}$$
4. Compute the code assignment c_k on node $k, k \in N^2_u$ using Eq. (15).
5. Activate link (u,v) in time slot t if:
$$\forall k \in N^2_u \text{ and } c_u = c_k, p^t_u > p^t_k \tag{19}$$

$\square$

5. NEIGHBOR PROTOCOL

Because collision-free transmission scheduling based on NCR depends on accurate two-hop neighborhood information, it is critical for a node to realize and incorporate neighborhood changes promptly. A neighbor protocol handles these changes in a reliable fashion for NAMA, LAMA and PAMA. We describe briefly the mechanisms provided in NAMA to deal with topology changes.

We rely on the membership section of each block to accept new members to channel access scheduling. Nodes are differentiated in terms of those that are already participating in the channel access scheduling and those that are not. The former nodes are silent during the membership section of a block, when latter nodes announce their existence using signal frames.

A new member first listens to the network traffic for at least a complete block before it tries to participate in the scheduling. The duration of a block, which is S_b sections, is derived such that it is highly probable that every two-hop neighbor of the new member transmits at least once in the block. We consider two-hop instead of one-hop neighbors because the probability of each node being activated is the reciprocal of the number of its two-hop neighbors. This situation has been formulated as an occupancy problem in combinatorial mathematics [10] [18], which pursues the probability of having m empty cells after randomly placing r balls into n cells, where r corresponds to the block size, and n corresponds to the number of two-hop neighbors of a new member. We use the result on the probability of leaving exactly m cells empty, which is:

$$p_m(r,n) = n^{-r} \binom{\text{n}}{\text{m}} \sum_{v=0}^{n-m} (-1)^v \binom{\text{n-m}}{\text{v}} (n-m-v)^r \quad (20)$$

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 in a block. Figure 8 shows the minimum numbers of balls (block size) to allow $p_0(r,n) > 0.99$, given different numbers of cells (two-hop neighbors). In reality, we put an upper and a lower bound on the block size such that less time is spent on neighbor coordination while new members still can quickly notify the network.

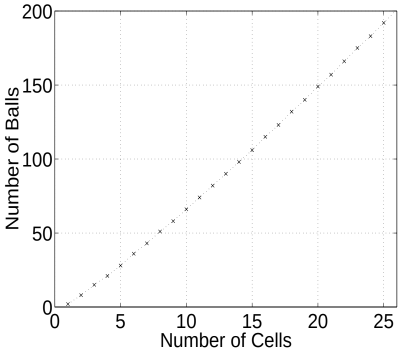


Figure 8: Number of Balls vs. Number of Cells Such That $p_0(r,n) > 0.99$

When the time for the membership section comes, the new member randomly selects a segment within the section to transmit its signal, which contains its ID number and part number and those of its one-hop neighbors (Figure 4). It is expected that all one-hop neighbors of the new member hear the signal and incorporate the new member in their one-hop neighbor set.

There could be collisions when multiple new members within two hops try to notify the network in exactly the same segment of the membership section. We resolve these hidden-neighbor relations by the next case.

When two nodes become one-hop neighbors, they need to firstly recognize each other and then to synchronize their one-hop neighbor information. We name such two nodes a and b, respectively, and consider the recognition of b by a in two cases.

- a receives a complete data frame from b. In this case, a sends out a neighbor update in its next data frame regarding the status of b. Unless b sends back a neighbor update about a, a sends out a signal in the membership section like a new member does and waits for the acknowledgment from b. The process is repeated until b recognizes a, when a and b exchange complete one-hop neighbor information.
- a does not receive a complete data frame from b, but detects collisions in some time slots. In this case, a sends out a signal in the following membership section like a new member does. As long as collisions exist, a keeps sending signals in the membership section. Once a is recognized by b, b follows the process as in the previous case.

In the second case, collisions may also be caused by two one-hop neighbors of a not knowing each other, which requires a to send out updates about one-hop neighbors in the colliding part to resolve conflicts.

On the other hand, a node a detects the disappearance of its existing one-hop neighbor b if neither data frame nor signal is received from b for a couple of blocks, in which case a deletes b from its one-hop neighbor set as well as one-hop neighbors reported by b. Node a generates a neighbor-delete update to notify its one-hop neighbors about the change.

Neighbor update information requires reliable propagation, thus acknowledgment and retransmission mechanisms are integral parts of neighbor protocol, which we do not specify in this paper.

6. PERFORMANCE

6.1 Expected Performance Differences

In NAMA, contending members are nodes within two hops, and they are assigned into the different parts and time slots of a section. Therefore, the average number of contending nodes for each time slot becomes

$$\frac{|N_i^1 \cup N_i^2|}{P_s \cdot T_p}$$

Although contention is less than the number of two-hop neighbors, the chances of transmission are also diminished by the same factor.

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

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

according to Eq. (14). Though the contention level is higher than NAMA, nodes compete for every time slot.

PAMA is more topology-dependent than the other two protocols. Not only two-hop neighbors, but also 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, if we assume one-hop neighbors of the endpoints are 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 highest throughput to the network because of a better spatial reuse of the channel, as shown in the simulations.

6.2 Simulation Results

We simulate the performance of NAMA, LAMA and PAMA in static topologies. The performance of the three 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:

- 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.
- Bandwidth of a radio transmission is up to 2 Mbps.
- A time unit in the simulation equals one time slot. A time slot last 8 milliseconds including guard time, long enough to transmit a 2KB packet.
- In NAMA, the number of time slots within a part is $T_p = 5$, and the number of parts within a section is $P_s = 3$. Thus, a section lasts 120 milliseconds.
- In NAMA, the lower and upper limits on the block size are 31 and 97, respectively.
- 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 800 seconds (equal to 100000 time slots) in the fully connected scenario and 400 seconds (equal to 50000 time slots) in the multihop network scenario, long en ough to compute the metrics of interests.

6.2.1 *Fully Connected Scenario:*

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. Figure 9 shows the delay values under different loads in the four cases as well as a theoretical curve derived from Eq. (7) with q_i values as shown in the figures. NAMA and LAMA seem to fit well with the theoretic analysis, but PAMA shows higher delays in the same situations. This is because the contention sources are different in PAMA from NAMA and LAMA. In PAMA, contending entities are links, and contention comes from adjacent links of every link. The q_i value for PAMA in the fully connected scenario is:

$$q_i = \frac{1}{4 \cdot |V| - 2} \cdot \left(1 - \frac{1}{2|C_{pn}|}\right)^{|V|-2} \tag{21}$$

where the second factor is due to elimination of hidden terminal interference. Hidden terminal problem can be improved if more spreading codes are available.

Taking the 10-node network as an example, the q_i value for each link is $\frac{1}{4\times10-2} \cdot \left(\frac{59}{60}\right)^{10-2}$ =0.023 in PAMA, which would result in a delay of at least 46 time slots by Eq. (8). 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.

Figure 10 shows the throughput of the three protocols. As predicted in Eq. (9), 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

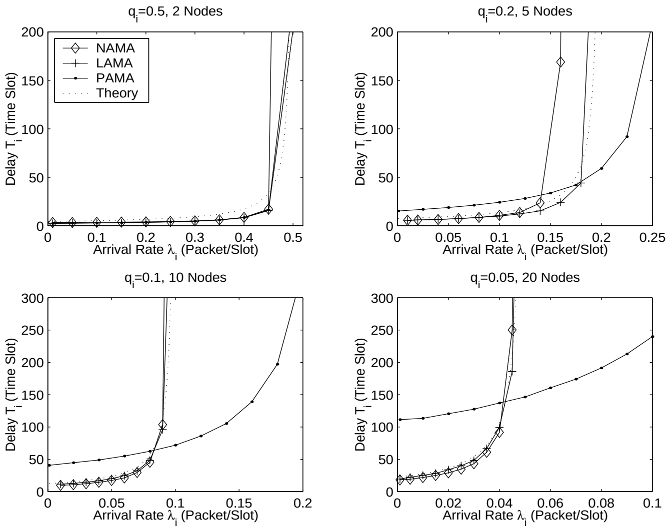


Figure 9: Average Packet Delays In Fully-Connected Networks

access protocols that experience great loss in the throughput when the network load goes beyond certain point. Notice that PAMA allows higher sustainable load in the system than NAMA and LAMA because PAMA allows channel reuse even when the topology is fully connected.

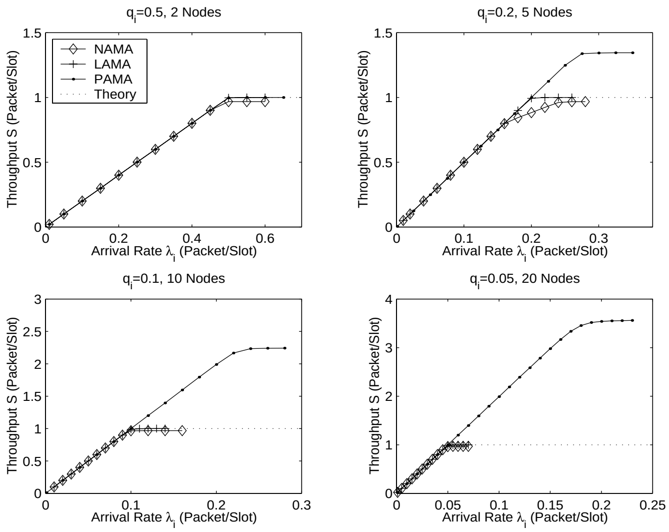


Figure 10: Packet Throughput Of Fully-Connected Networks

6.2.2 *Multihop Network Scenario:*

Figure 11 and 12 show the delay and throughput features 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 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, 400 meters, respectively, we also virtually change the topology and contention levels in each case.

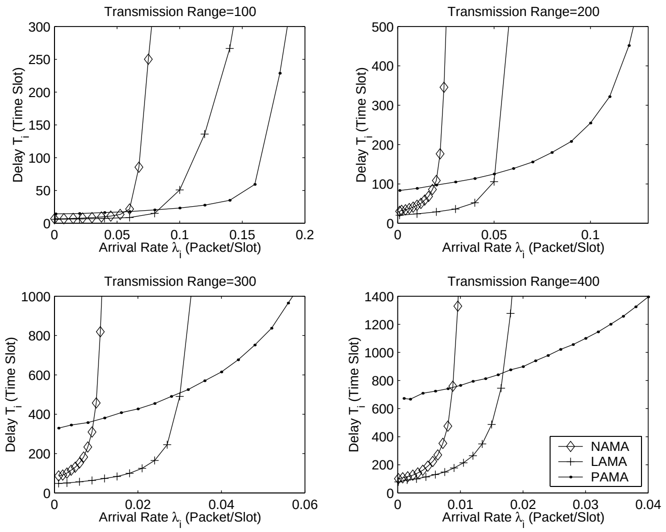


Figure 11: Average Packet Delays In Multihop Networks

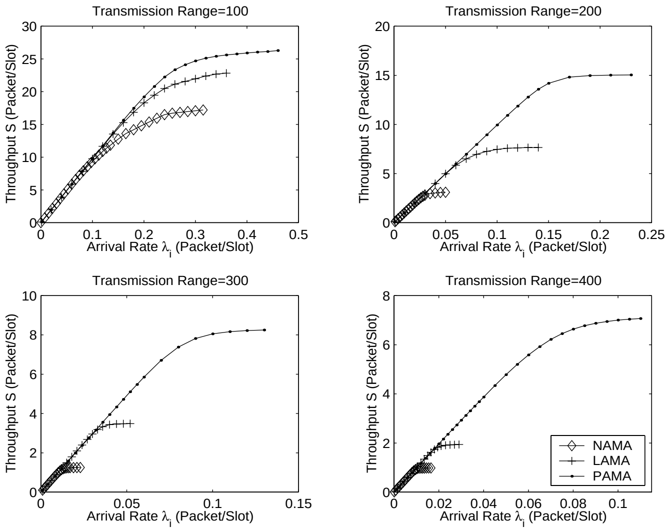


Figure 12: Packet Throughput Of Multihop Networks

Figure 11 demonstrates the advantage of LAMA over NAMA because of improvements on channel reuse within two hops of each node by applying code division multiplexing in LAMA. PAMA still gives higher starting point to delays than the other two even when network load is low due to similar reasons as in fully connected scenario. However, PAMA appears to have slower increases when the network load goes larger, which explains the higher spectrum and spatial reuse of the common channel by pure link-oriented scheduling.

One interesting point about PAMA is that the contention levels are so different on each outgoing link of a link that it is fair to distribute the load to one-hop neighbors inversely to the contention levels of these links. In routing control protocols, higher contention levels are translated into higher cost to get data packets through those links. Not shown in the figures, this uneven distribution of load onto outgoing links has improved the delay aspect of PAMA. We expect the similar improvements on delays in LAMA, though less obvious than that of PAMA.

Figure 12 shows the simple linear behavior of system throughput to the load values. System throughput is an indication of the average channel reuse ratio in multihop wireless networks. The throughput levels off in NAMA and LAMA when the load values approximate and exceed the probabilities that a node may access the channel, where delays increase drastically as shown in Figure 11. PAMA still achieves higher loads than the other two protocols, and shows linear increases in throughput beyond the highest sustainable loads of NAMA and LAMA.

6.2.3 *Comparison with Static Scheduling:*

The unified framework UxDMA [21] defines a parameterized algorithm to derive various channel access schedules according to the network topology and the type of entities to be colored. 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. Given a group of constraints and the graph, UxDMA computes the coloring on the corresponding entities that satisfies the constraints. 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.

Accordingly, we select appropriate subset of the constraints for each of our scheduling protocols, and the derived numbers of colors from UxDMA are compared against the average activation intervals of each entity in our protocols. Table 1 lists the set of input constraints to UxDMA for NAMA, LAMA and PAMA, respectively. The meaning of each symbol is referred to the original paper in [21].

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

Table 1: Constraint Sets For Our Protocols

Note that LAMA is a node-oriented activation scheme even though links are the actual entities to be colored in UxDMA-LAMA. Similar mixture of node and link identities happens in PAMA, which reflects the great flexibility of the contention resolution algorithms in utilizing available information.

The average interval is obtained from the following formula:

$$k = \frac{\text{Duration of simulation } \times \text{Number of entities}}{\text{Total number of activations}} \qquad (22)$$

which means that the average activation interval for all entities equals to the simulation time, counted in terms of time slots, divided by the number of activations to every entity.

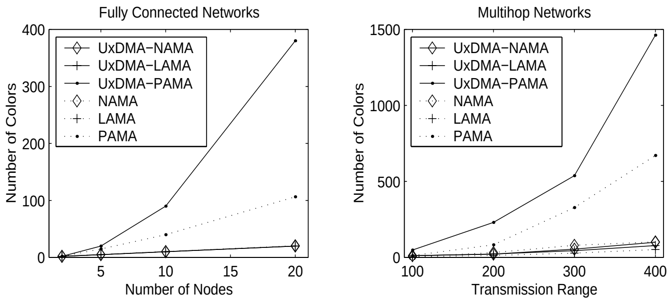


Figure 13: Coloring Efficiency Comparisons

Figure 13 shows the average activation intervals of nodes and links in the graphs used in previous simulations, as are contrasted with the number of colors obtained by running the UxDMA algorithms on the same topology graphs for respective protocols. When computing the colorings on the graphs in UxDMA, an optimal ordering, PMNF (Progressive Minimum Neighbors First) heuristic, has been applied in each computation so that the colorings "perform quite close to optimum" [21].

In Figure 13, NAMA, LAMA and PAMA perform very close to or use fewer colors than their counterparts of the static assignment algorithms. The big discrepancy between PAMA and UxDMA-PAMA is due to the fact that PAMA employs spread spectrum codes that largely invalidate the constraint E_{tr}^1 in color assignments.

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 two-hop neighborhood information to dynamically determine at each node which node should be allowed to transmit in each collision-resolution context, which can be a time slot. Based on this approach, protocols were introduced for both node-activation and link-activation channel access scheduling in packet radio networks. The advantages of the protocols are that (a) they do not need the contention phases or schedule broadcasts, as adopted by many other channel access scheduling algorithms; (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, as opposed to 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 unicast using spread spectrum techniques.

8. REFERENCES

[1] 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, 18-23 Sept. 1994. IOS Press.

[2] 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.

[3] 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.

[4] 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, March 1985.

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

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

[7] 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.

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

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

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

[11] 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, 15-17 June 1988.

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

[13] 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, 2-5 Nov. 1997.

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

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

[16] 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, 21-25 March 1999.

[17] 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, June 1998.

[18] R. J. Larsen and M. L. Marx. *An introduction to probability and its applications*. Number p.165-167. Englewood Cliffs, N.J. : Prentice-Hall, 1985.

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

[20] 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.

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

[22] 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, 23-27 April 1989. IEEE Comput. Soc. Press.

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

[24] 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, 7-10 Sept. 1982.

[25] 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, September 21–24 1999.

[26] 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, December 1975.

[27] 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, 29 March-2 April 1998.