

# Distributed Transmission Scheduling Using Code-Division Channelization\*

Lichun Bao and J.J. Garcia-Luna-Aceves<sup>1</sup>

School of Engineering, University of California Santa Cruz, CA 95064, USA  
baolc, jj@soe.ucsc.edu

**Abstract.** We present the Hybrid Activation Multiple Access scheduling protocol (HAMA) for wireless ad hoc networks. Unlike previous channel access scheduling protocols that activate either nodes or links only, HAMA is a node-activation oriented channel access scheduling protocol that also maximizes the chance of link activations. According to HAMA, the only required information for scheduling channel access at each node is the identifiers of neighbors within two hops. Using this neighborhood information, multiple winners for channel access are elected in each contention context, such as a time slot in a frequency band or a spreading code. Except for time slot synchronization and neighbor updates on the two-hop neighborhood changes, HAMA dedicates the bandwidth completely to data communication. The delay and throughput characteristics of HAMA are analyzed, and its performance is compared with pure node activation based scheduling protocols by simulations.

## 1 Introduction

In ad hoc networks, a channel access scheme usually takes one of two approaches: on-demand or by-schedule. The on-demand approach started with ALOHA and CSMA [9] and continued with several collision avoidance schemes (e.g., MACA [8], FAMA [5] and others). However, as traffic load increases, network throughput drastically degrades because the probability of collisions rises, preventing any station from acquiring the channel. Furthermore, random access MAC protocols cannot provide end-to-end quality assurance due to adverse channel conditions, such as near-far phenomena, and capture effects on the channel.

On the other hand, 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 in the code, time, frequency or space divisions of the channel. Collision-free channel access scheduling is typically treated as a node or link coloring problem on graphs representing the network topologies. Searching for an optimal channel access scheduling results in NP-hard problems in graph theory (such as  $k$ -colorability on nodes or edges) [3,

---

\* This work was supported in part by Advanced Technology Office of the Defense Advanced Research Projects Agency (DARPA) under grant No. DAAD19-01-C-0026, and by the U.S. Air Force/OSR under grant No. F49620-00-1-0330.

4,13]. Polynomial algorithms are known to achieve suboptimal solutions using randomized approaches or heuristics based on such graph attributes as the degree of the nodes.

A unified framework for TDMA/FDMA/CDMA channel assignments, called UxDMA, was described by Ramanathan [12], which combines the general approaches of many other channel access scheduling algorithms in that these algorithms are now represented by UxDMA with different parameters. The parameters to UxDMA indicate the constraints put on the graph entities (nodes or links) such that entities related by the constraints are colored differently. The results of color assignments correspond to channel assignments to these nodes or links in the time, frequency or code domain.

However, since the global topology is required to derive the channel access schedule, topology information needs to be collected and frequent schedule broadcasts have to be carried out in mobile networks, which would consume a significant portion of the scarce wireless bandwidth.

We propose a new channel access scheduling protocol, called HAMA (Hybrid Activation Multiple Access scheduling), that supports both broadcast, multicast and unicast communications in wireless networks using a time and code division multiple access scheme. HAMA establishes channel access schedules directly on local topologies of a node within two hops, which avoids the schedule exchanges and resolutions present in other scheduling algorithms.

Section 2 describes the assumptions we make about multihop wireless networks, and section 3 specifies HAMA. Section 4 presents the neighbor protocol for propagating and acquiring two-hop neighbor information. Section 5 analyzes the delay and throughput attributes of HAMA using queuing theory, and HAMA's performance is studied by simulations and comparisons with NAMA, the pure node-activation based channel access scheduling protocol, and UxDMA, the best-performing scheduling algorithm known to date. Section 6 concludes the paper.

## 2 Assumptions

The creation and analysis of a packet radio network usually depend on the physical technologies provided and the scenarios where the multihop wireless network is deployed. We assume that:

- Each node in the network is mounted with an omni-directional radio transceiver, and assigned a unique ID number;
- The radio of each node may only work in half-duplex mode, i.e., either transmit or receive data packet at a time, but not both;
- Time is synchronized at each node, and nodes access the channel based on slotted time boundaries. Each time slot lasts long enough to transmit a complete data packet, and is identifiable relative to a consensus starting point.

We do not address the time synchronization issue, but suggest achieving it by attaching timing information in physical layer packet framing and listening to data traffic in the network to align time slots to the latest starting point of a complete packet transmission by one-hop neighbors [10].

The topology of a packet radio network is represented by an undirected graph  $G = (V, E)$ , where  $V$  is the set of network nodes, and  $E$  is the set of links between nodes. If link  $(u, v) \in E$ , then  $(v, u) \in E$ , and node  $u$  and  $v$  are within the transmission range of each other, so that they can exchange packets via the common channel, in which case  $u$  and  $v$  are called *one-hop neighbors* of each other. The set of one-hop neighbors of a node  $i$  is denoted as  $N_i^1$ .

Since the same codes on a set of the appropriately selected frequency bands can be equivalently considered to be different codes, we only consider channel scheduling based on a time-slotted code division multiple access scheme. Before describing a neighbor protocol in section 4, we assume that each node already knows its two-hop neighbor information, which is the one-hop neighbors of the node itself and the one-hop neighbors of its one-hop neighbors. That is, the collection of neighbor information gathered at node  $i$  is:

$$N_i^1 \cup \left( \bigcup_{j \in N_i^1} N_j^1 \right).$$

## 3 HAMA

### 3.1 Code Assignment

HAMA is a time-slotted code division multiple access scheme using direct sequence spread spectrum (DSSS) transmission techniques. In DSSS, code assignments are categorized into transmitter-oriented code assignment (TOCA), receiver-oriented code assignment (ROCA), or a per-link-oriented code assignment (POCA) schemes [6,7,11]. HAMA adopts transmitter-oriented code assignment (TOCA), because of its broadcast capability.

We assume that a pool of well-chosen quasi-orthogonal pseudo-noise codes,  $C_{pn} = \{c^k\}$ , are available for each node to choose from. The codes are identified by the superscript  $k = 0, 1, 2, \dots, |C_{pn}| - 1$ .

The code for each node is denoted by  $i.\text{TxCode} (\in C_{pn})$ , and is computed in every time slot. Consequently, a node has varying contention situations for transmission from time slot to time slot. Eq. (1) computes the pseudo-noise code for node  $i$  at time slot  $t$ .

$$i.\text{TxCode} = c^k, \quad k = \text{Hash}(i \oplus t) \bmod |C_{pn}|. \quad (1)$$

where  $\text{Hash}(x)$  is an integer pseudo-random number generator that generates the message digest of byte-stream input  $x$ . The sign ‘ $\oplus$ ’ is designated as the mathematical operation to carry out concatenation on its two operands.

Because we have a limited number of pseudo-noise codes for assignment in HAMA, it is possible that multiple nodes share the same code. We resolve possible collisions using a contention resolution algorithm in HAMA.

### 3.2 Nodal Modes and Operations

The state of a node is related with a dynamic priority assigned to the node in each time slot. The priority of a node  $i \in V$  is computed according to Eq. (2), given the current time slot  $t$ :

$$i.\text{prio} = \text{Hash}(i \oplus t), \quad (2)$$

where the same function  $\text{Hash}(\mathbf{x})$  was used in Eq. (1).

The mode of a node is determined by the priorities of itself and the surrounding neighbors. We differentiate nodes in terms of their capabilities for transmissions in either broadcast or unicast mode. In general, we allow data packets transmitted from nodes with higher priorities to nodes with lower priorities. Accordingly, the mode of a node is defined into six states under two categories:

- Receiver:
  1. *Sniffer* (Sf): The node does not have the highest priority among its one-hop neighbors.
  2. *Sink* (S): The node has the lowest priority among its one-hop neighbors.
- Transmitter:
  1. *B-Transmitter* (BT): The node has the highest priority among its two-hop neighbors, and can broadcast to its one-hop neighbors without contentions from its two-hop neighbors.
  2. *U-Transmitter* (UT): The node has the highest priority among its one-hop neighbors only. Therefore, the node may transmit to some one-hop neighbors, but cannot do so to other one-hop neighbors.
  3. *S-Transmitter* (ST): The node has the highest priority among the one-hop neighbors of a *Sink* neighbor.
  4. *Yield* (Y): The node could have been either a *B-Transmitter* or a *U-Transmitter*, but chooses to abandon channel access if its transmission may result in hidden-terminal interference at its one-hop neighbors.

If a node  $i$  determines that is a transmitter (BT, UT or ST), it prepares data flows for transmissions on its assigned code. The node has to select corresponding one-hop neighbors that can receive its packets. For convenience, we denote the receiver set by  $i.\text{out}$ , and the packets stored for the eligible receivers by  $i.Q(i.\text{out})$ .

If a node  $i$  happens to be in reception mode (Sf or S), it chooses a neighbor, denoted by  $i.\text{in}$ , which has the highest priority among its one-hop neighbors, and listens on the transmission code assigned to  $i.\text{in}$ .

The transmission code is assigned to a node in transmission mode using Eq. (1), and the reception code of a node  $i$ , denoted by  $i.\text{RxCode}$ , is naturally aligned to transmission code of  $i.\text{in}$ .

HAMA applies a novel neighbor-aware contention resolution algorithm (NCR) [1] to compute the channel access schedules in each time slot. Provided that each node obtains the accurate knowledge of its neighbors within two hops through neighbor protocol, HAMA decides whether a node  $i$  transmits or receives a packet in time slot  $t$  on an appropriate code as presented in Fig. 1.

```

HAMA( $i, t$ )
{
  /* Initialized to listen. */
  1   $i.mode = Sf$ ;
  2   $i.in = -1$ ; /* Null. */
  3   $i.out = \emptyset$ ; /* Empty set. */

  4  for ( $k \in N_i^1 \cup (\bigcup_{j \in N_i^1} N_j^1)$ ) {
  5     $k.prio = Hash(t \oplus k)$ ;
  6     $code = k.prio \bmod |C_{pn}|$ ;
  7     $k.TxCODE = c^{code}$ ;
  8  }

  9  for ( $\forall j \in N_i^1 \cup \{i\}$ ) {
 10    if ( $\forall k \in N_j^1, j.prio > k.prio$ )
      /* May unicast. */
 11       $j.mode = UT$ ;
 12    elseif ( $\forall k \in N_j^1, j.prio < k.prio$ )
      /* A sink. */
 13       $j.mode = S$ ;
 14  }

  /* More findings about  $i$ . */
 15  if ( $i.mode \equiv UT$  and
 16     $\forall k \in \bigcup_{j \in N_i^1} N_j^1, k \neq i,$ 
 17     $i.prio > k.prio$ )
      /* Can broadcast. */
 18     $i.mode = BT$ ;
 19  elseif ( $i.mode \equiv Sf$  and
 20     $\exists j \in N_i^1, j.mode \equiv S$  and
 21     $\forall k \in N_j^1, k \neq i, i.prio > k.prio$ ) {
      /* Can unicast to a sink */
 22     $i.mode = ST$ ;

      /*  $i$  has to listen to  $j$ . */
 23    if ( $\exists j \in N_i^1, j.mode \equiv UT$  and
 24       $\forall k \in N_i^1, k \neq j, j.prio > k.prio$ )
 25       $i.mode = Sf$ ;
 26  }

  /* Determine dest or src. */
 27  switch ( $i.mode$ ) {
 28    case  $BT$ :
      29      for ( $j \in N_i^1$ )
      30         $i.out = i.out \cup \{j\}$ ;

      31      case  $UT$ :
      32        for ( $j \in N_i^1$ )
      33          if ( $\forall k \in N_j^1, k \neq i,$ 
      34             $i.prio > k.prio$ )
      35             $i.out = i.out \cup \{j\}$ ;

      36      case  $ST$ :
      37        for ( $j \in N_i^1$ )
      38          if ( $j.mode \equiv S$  and
      39             $\forall k \in N_j^1, k \neq i,$ 
      40             $i.prio > k.prio$ )
      41             $i.out = i.out \cup \{j\}$ ;

      42      case  $S, Sf$ :
      43        if ( $\exists j \in N_i^1$  and
      44           $\forall k \in N_i^1, k \neq j,$ 
      45           $j.prio > k.prio$ ) {
      46           $i.in = j$ ;
      47           $i.RxCODE = j.TxCODE$ ;
      48        }
      49  }

      /* Hidden-Terminal Avoidance. */
 50  if ( $i.mode \in \{ UT, ST \}$  and
 51     $\exists j \in N_i^1, j.mode \neq UT$  and
 52     $\exists k \in N_j^1, k.prio > i.prio$  and
 53     $k.TxCODE \equiv i.TxCODE$ )
 54     $i.mode = Y$ ;

      /* Ready to communicate. */
 55  if ( $i.mode \in \{ BT, UT, ST \}$  and
 56     $i.Q(i.out) \neq \emptyset$ ) {
      /* FIFO */
 57     $pkt = Dequeue(i.Q(i.out))$ ;
 58    Transmit  $pkt$  on code  $i.TxCODE$ ;
 59  }
 60  else
 61    Listen on code  $i.RxCODE$ ;
  } /* End of HAMA. */

```

Fig. 1. HAMA Specification

## 4 Neighbor Protocol

In HAMA, topology information within two hops of a node plays a essential role for channel access operations. In mobile networks, network topologies change frequently, which affects the transmission schedules of the mobile nodes. The ability to detect and notify such changes promptly relies on the neighbor protocol so as to reharmonize channel access scheduling.

### 4.1 Signals

Since HAMA adopts dynamic code assignment for channel access using the identifier and time slot number, it is impossible for a node to detect a new one-hop neighbor that transmits data packets with varying codes. We have to use an additional time section, called *neighborhood section*, for sending out “hello” messages and for mobility management purposes. The neighborhood section lasts for  $T_{nbr}$  time slots following every  $T_{hama}$  HAMA time slots. Channel access is still based on code division scheme but the transmission code is fixed over a commonly known one selected from  $C_{pn}$ .



Fig. 2. Signal Frame Format

In addition, a time slot within the neighborhood section is further divided into a number of smaller time segments fit for transmitting short signal packets, where their format is as illustrated in Fig. 2.

In Fig. 2, the signal frame transmitted by node `srcID` is indicated by its `type` field. And field `#add` and `#del` count the numbers of the following `nbrIDs` for addition and deletion, respectively, of the neighbors from the transmitter’s neighbor topology.

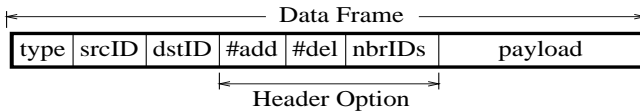


Fig. 3. Data Frame Format

Besides signals, one-hop neighbor updates are also propagated using broadcast packets if a node is activated in BT-mode, so that the update information of a node gets to all its neighbors efficiently. One-hop neighbor updates are piggy-back in the option field of a data frame whenever possible and necessary. Fig. 3

illustrates the data packet format, which includes similar neighbor update fields as in Fig. 2, besides regular fields such as destination `dstID` and payload of the packet.

## 4.2 Mobility Handling

Signals are used by the neighbor protocol for two purposes. One is for a node to say “hello” to its one-hop neighbors periodically in order to maintain connectivity. The other is to send neighbor updates when a neighbor is added, deleted or needs to be refreshed. In case of a new link being established, both ends of the link need to notify their one-hop neighbors of the new link, and exchange their complete one-hop neighbor information. In case of a link breaking down, a neighbor-delete update needs to be sent out. An existing neighbor connection also has to be refreshed periodically to the one-hop neighbors for robustness. If a neighbor-delete update is not delivered to some one-hop neighbors, those neighbors age out the obsolete link after a period of time.

However, because of the randomness of signal packet transmissions, it is possible for a signal sent by a node to collide with signals sent by some of its two-hop neighbors. Due to the lack of acknowledgments in signal transmissions, multiple retransmissions of the update information are needed for a node to ensure the delivery of the message to its one-hop neighbors. Signal intervals also jitter by a small value so that signals transmitted in the neighborhood spread out evenly over the neighborhood section to avoid collisions.

Furthermore, retransmissions of a signal packet can only achieve a certain probability of successful delivery of the message. Even though the probability approaches one as the retransmissions are carried out repetitively, the neighbor protocol has to regulate the rhythm of sending signals, so that the desired probability of the message delivery is achieved with a small minimum number of retransmissions in the shortest time, thus incurring the least amount of interference to other neighbors’ signal transmissions.

The number of signal retransmissions and the interval between retransmissions depends on the number of two-hop neighbors. The more neighbors a node has, the longer the interval value is chosen for signal retransmissions. Since the probability of each signal transmission trial can be determined by the interval value, the number of retransmissions can be derived to achieve the desired probability of successful message delivery.

Consequently, the latency of the message delivery using retransmission approach is decided by the product of the interval and the number of retransmissions. If we do not depend on the neighbor updates transmitted in the option field of data frames, enough time slots should be allocated to the neighborhood section in the time division scheme to achieve the desired latency of message delivery, which determines the ratio between  $T_{hama}$  and  $T_{nbr}$ . Since the time division is fixed during the operations of HAMA, the ratio  $T_{hama} : T_{nbr}$  is computed beforehand in the neighbor protocol to handle networks with moderate density. We do not specify the relations from this aspect in this paper.

## 5 Performance

### 5.1 Delay Analysis

When data packets arrive at a node according to a Poisson process with rate  $\lambda$  and are served according to the first-come-first-serve (FIFO) strategy, we can analyze the delay properties of HAMA by a steady-state M/G/1 queues with server vacations, where the single server is the node. The server takes a vacation for  $V$  of one time slot when there is no data packet in the queue; otherwise, it looks for the next available time slot to transmit the first packet waiting in the queue.

Because a node accesses the channel in a time slot by comparing the random priorities assigned its one-hop neighbors, the attempt of channel access in each time slot is a Bernoulli trial for each node. Depending on the neighborhood topology, each node has a probability  $q$  of winning the channel access contention in each time slot. Therefore, the service time of a data packet in the queuing system is a random variable following geometric distribution with parameter  $q$ . Denote the service time as  $X$ , we have  $P\{X = k, k \geq 1\} = (1 - q)^{k-1}q$ .

The mean and second moments of random variable  $X$  and  $V$  are:

$$\bar{X} = \frac{1}{q}, \quad \overline{X^2} = \frac{2 - q}{q^2};$$

$$\bar{V} = \overline{V^2} = 1.$$

So that the extended Pollaczek-Kinchin formula for the waiting time in the M/G/1 queuing system with server vacations [2]

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

readily yields the average waiting period of a data packet in the queue as:

$$W = \frac{\lambda(2 - q)}{2q(q - \lambda)} + \frac{1}{2}.$$

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

$$T = W + \bar{X} = \frac{2 + q - 2\lambda}{2(q - \lambda)}. \quad (3)$$

To keep the queuing system in a steady state without packet overflow problems, it is necessary that  $\lambda < q$ .

Since HAMA is capable of both node activation and link activation, the delays of broadcast and unicast traffics should be considered separately because the contenders of node activation and link activation are different, so are the respective activation probabilities.

## 5.2 Throughput Analysis

Network throughput is defined as the number of packets going through the network at the same time including both broadcast and unicast traffics. On account of the collision freedom in HAMA, the shared channel can serve certain load up to the channel capacity allowed without degradations. That is, the throughput over the common channel is the summation of arrival rates at all network nodes as long as the queuing system at each node remains in equilibrium on the arrival and departure events. Therefore, the system throughput  $S$  is derived as:

$$S = \sum_{k \in V} \min(\lambda_k, q_k), \quad (4)$$

where  $q_k$  is the probability that node  $k$  may be activated, and  $\lambda_k$  is the data packet arrival rate at link  $k$ .

## 5.3 Simulation Results

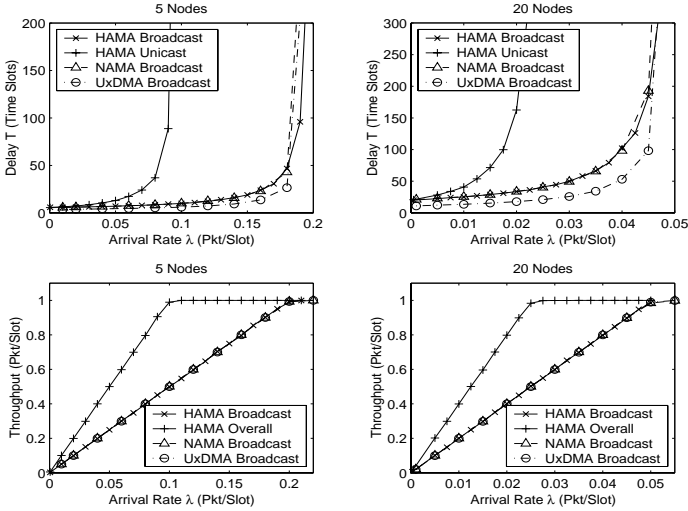
The behaviors of HAMA is simulated in two scenarios: fully connected networks with different numbers of nodes, and multihop networks with different radio transmission ranges. The delay and throughput attributes of HAMA are gathered in each simulation, and compared with those of NAMA [1] and UxDMA [12] in the same simulation scenarios.

In the simulations, we do not model the bandwidth of the radio channel with specific numbers, but use more abstract terms, such as *packets per time slot* for both arrival rate and throughput, which can be later translated into common bandwidth metrics, such as *Mbps* (megabits per second), given certain packet size distribution and transmission media. The following parameters and behaviors are assumed in the simulations:

- The network topologies are static to evaluate the scheduling performance of the algorithms, only.
- 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. Radiation energy outside the effective transmission range of the radio is considered negligible interference to other communications. All radios have the same transmission range.
- 30 pseudo-noise codes are available for code assignments, i.e.,  $|C_{pn}| = 30$ .
- Packets are served in First-In First-Out (FIFO) order. Only one packet can be transmitted in a time slot.
- All nodes have the same broadcast packet arrival rate for all protocols (HAMA, NAMA and UxDMA). In addition, HAMA is loaded with the same amount of unicast traffic as broadcast traffic to manifest the unicast capability of HAMA. The overall load for HAMA is thus twice as much as that of NAMA and UxDMA. The destinations of the unicast packets in HAMA are evenly distributed on all outgoing links.

- The duration of the simulation is 100,000 time slots, long enough to collect the metrics of interests.

In UxDMA, a constraint set is derived for broadcast activations as NAMA does, which is give by  $UxDMA-NAMA = \{V_{tr}^0, V_{tt}^1\}$ . The notation of each symbol is referred to the original paper in [12]. Constraint  $V_{tr}^0$  forbids a node from transmitting and receiving at the same time, while  $V_{tt}^1$  eliminates hidden terminal problem.



**Fig. 4.** Average Packet Delays In Fully-Connected Networks

In the fully connected scenarios, simulations were carried out in two configurations: 5- and 20-node networks, to manifest the effects of different contention levels. Fig. 4 shows the average delay values on the first row and average throughput on the second for HAMA, NAMA and UxDMA-NAMA, respectively, under different loads on each node in the two configurations. The horizontal parts in the throughput plots indicate the total network capacities provided by different protocols. Since all nodes are within one hop to each other, there can be only one unicast or broadcast in each time slot. The network throughput tops when the loads sum up to one.

In the multihop scenario, the simulations were conducted in networks that are generated by randomly placing 100 nodes within an area of  $1000 \times 1000$  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. The power of the transceiver on each node was set to 100, 200, and 300 meters, respectively, so that the network topology and contention levels in these simulations varied accordingly.

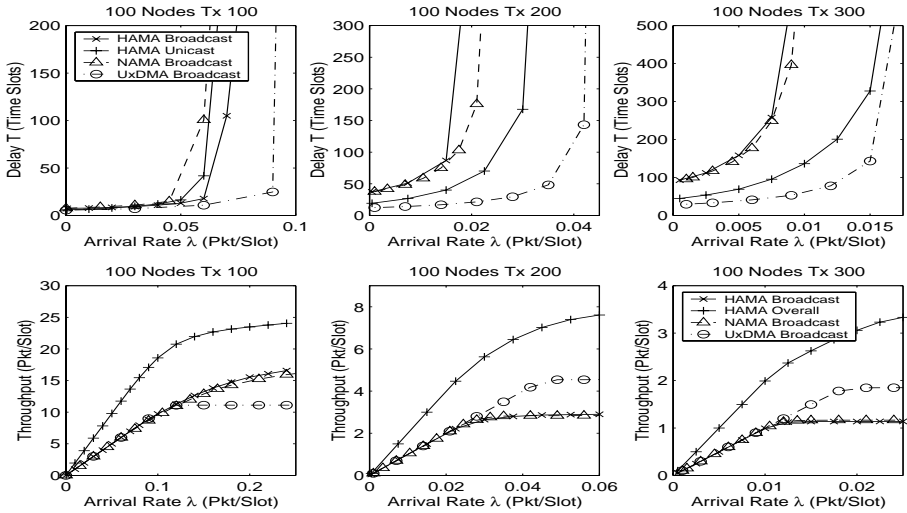


Fig. 5. Average Packet Delays In Multihop Networks

Fig. 5 shows the delay and throughput performance of HAMA, NAMA and UxDMA-NAMA in multihop networks. UxDMA-NAMA is better than HAMA and NAMA at broadcasting in some of the multihop networks, owing to its global knowledge about topologies. However, HAMA outperforms UxDMA-NAMA in overall network throughput.

Overall, HAMA has achieved much better performance than a previously proposed protocol, NAMA [1], by requiring only a little more processing on the neighbor information. Comparing HAMA with UxDMA, which uses global topology information, HAMA sustains similar broadcasting throughput, in addition to the extra opportunities for sending unicast traffic. The dependence on only two-hop neighbor information is also a big advantage over UxDMA.

## 6 Conclusion

We have introduced HAMA, a new distributed channel access scheduling protocol that dynamically determines the node activation schedule for both broadcast and unicast traffics. HAMA only requires two-hop neighborhood information, and avoids the complexities of prior collision-free scheduling approaches that demand global topology information. We have also analyzed the per-node delay and per-system throughput attributes of HAMA and NAMA [1], a node activation protocol, and compared system performance of HAMA with that of NAMA and UxDMA [12] by simulation.

## References

1. L. Bao and J.J. Garcia-Luna-Aceves. A New Approach to Channel Access Scheduling for Ad Hoc Networks. In *Proc. ACM Seventh Annual International Conference on Mobile Computing and networking*, Rome, Italy, Jul. 16-21 2001.
2. D. Bertsekas and R. Gallager. *Data Networks, 2nd edition*. Prentice Hall, Englewood Cliffs, NJ, 1992.
3. A. Ephremides and T.V. Truong. Scheduling broadcasts in multihop radio networks. *IEEE Transactions on Communications*, 38(4):456–60, Apr. 1990.
4. 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.
5. C.L. Fullmer and J.J. Garcia-Luna-Aceves. Floor acquisition multiple access (FAMA) for packet-radio networks. In *ACM SIGCOMM '95*, pages 262–73, Cambridge, MA, USA, Aug. 28 -Sep. 1 1995.
6. 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.
7. 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.
8. P. Karn. MACA - a new channel access method for packet radio. In *Proceedings ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, New York, Apr. 1990.
9. L. Kleinrock and F.A. Tobagi. Packet switching in radio channels. I. Carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, COM-23(12):1400–16, Dec 1975.
10. L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–65, Jul. 1978.
11. T. Makansi. Transmitter-Oriented Code Assignment for Multihop Radio Networks. *IEEE Transactions on Communications*, 35(12):1379–82, Dec. 1987.
12. S. Ramanathan. A unified framework and algorithm for channel assignment in wireless networks. *Wireless Networks*, 5(2):81–94, 1999.
13. 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.