MALS: Multiple Access Scheduling Based on Latin Squares

Lichun Bao
Donald Bren School of Information and Computer Sciences
University of California, Irvine, CA 92697

ABSTRACT

A Latin square of order n is an $n \times n$ square matrix that consists of n distinct symbols, where the symbols of each row and column are distinct. The number of different Latin squares beyond a very small order, such as 10, grows almost to infinity for practical applications. The distinctiveness and orderliness of each symbol in the Latin square present very attractive features in applying Latin squares to the scheduling problems. We propose a medium access scheme based on Latin squares, called MALS (Medium Access based on Latin Squares), from which deterministic time-division channel access schedules are generated for nodes in ad hoc networks. In MALS, each node is assigned with an index of a row in a dynamic Latin square, and the columns of the Latin square corresponds to the time-slotted channel. The value of the symbol in the Latin square for each time slot determines the priority of the node to access the channel. Because of the distinctiveness of the symbol value in each column, a unique priority is assigned to each node in each time slot, and the channel access at any moment is guaranteed collision-free. We apply MALS in both macro-time division channel access scheduling, where each time slot is long enough to contain a packet, and micro-time division channel access scheduling, where the time slot is comparable to the duration of the round trip time in point-to-point wireless communication. Simulations for both scenarios show the near-optimum performance of MALS for channel access scheduling in comparison with UxDMA and IEEE 802.11 DCF, respectively.

I. Introduction

The scheduled channel access problem in packet-switching wireless networks usually focuses on creating a conflict-free channel access schedule such that the transmitted packets from each node successfully arrive the intended receivers without detrimental interferences. However, scheduled channel access schemes have inherent performance drawbacks due to the wastes of unused time slot assignments and the long delays between successive channel accesses by each node. In addition, the time synchronization constraints, schedule establishment overheads, non-adaptiveness to traffic load changes have frequently put the applications of scheduled channel access schemes for ad hoc networks in doubts. On the other hand, although scheduled channel access was not a panacea to all the application scenarios, time-division channel access schemes are dominant in cellular networks [18] for voice communications because of its capability to provide real-time guarantees. Therefore, channel access scheduling still stands as an attractive approach in wireless communications.

Wireless medium is broadcast in nature. Transmission scheduling to a common channel in multi-hop packet radio networks can be based on node or link activations according to the communication models. The schedule generation consists of assigning stations to different times and data channels (e.g., frequencies, spreading codes, or their combination) in a way that no collisions (conflicts) occur, and efficient spatial reuse of the available bandwidth is achieved.

The problem of find the optimum node/link activation in a given network has long been proved an NP-hard problem [16]. A generic unified framework for (T/F/C)DMA channel assignment in multi-hop networks, called UxDMA, was proposed to compute a suboptimal k-coloring of a directed graph within polynomial steps [15]. A number of topology-transparent scheduling methods have been proposed [3] [7] [8]. The basic idea of the topology-transparent scheduling is for a node to transmit each data frame in a number of time slots during a time frame.

Many other solutions provide distributed channel access scheduling, combining on-demand resource request and conflict-free allocation mechanisms [4] [5], thus eliminating the requirement of the complete topology information. T-MAH used a token to schedule channel access in a round-robin fashion [12].

In contrast to other protocols that depend on explicit reservation and allocation cycles, NAMA [2] and SEEDEX [17] proposed to utilize a pseudo random number generator to decide the priority or probability of channel access in an implicit fashion. Because the pseudo random number generator is known by all the participating nodes in the networks, the channel access schedule of each node is also known by its neighbors, therefore saving the schedule exchange and coordination overheads over the wireless channel.

Besides the scheduling approaches, random access scheme provides a more flexible approach in wireless communications because it does not require explicit topology information of channel access schedules from other nodes to determine the channel access decision.

Random access schemes are not totally different from the scheduled channel access mechanisms mentioned above — they mainly differ in the time slot sizes. The scheduled access schemes use time slots long enough to contain complete data frames, while the random access schemes use short time slots just enough for carrier sensing. In this paper, we refer to the former schemes as **macro-time division multiple access**, and the latter as **micro-time division multiple access**.

Random access scheme has been extensively studied throughout the years [20] [22] [23]. IEEE 802.11 distributed coordination function (DCF) is a dominant wireless MAC protocol, and has been constantly adapted by the research community to satisfy many quality of service (QoS) requirements [1]. However, random access scheme has always received criticisms on its susceptibility to throughput and fairness degradation under various channel conditions unless fair scheduling mechanisms are implemented or sufficient topological information is known beforehand.

Given the pros and cons of both scheduled channel access and random channel access schemes, we try to find a unified solution, which is based the scheduling approach, and provides fair, prompt and efficiency channel access under various network traffic scenarios. We propose a medium access scheme based on *Latin squares* [6], called *MALS* (Medium Access based on Latin Squares), from which deterministic time-division channel access schedules are generated for nodes in ad hoc networks.

MALS is applied in both macro-time division channel access scheme and and micro-time channel access scheme. Section II describes the problems in scheduled channel access mechanism, NAMA, and random access mechanism, IEEE 802.11 DCF. Section III introduces Latin squares. Section IV describes MALS in macro-time division and micro-time division scenarios. Section V shows mechanisms to support network mobility and neighbor maintenance. Simulations for both scenarios show the near-optimum performance of MALS for channel access schedules in section VI, where the performance of MALS is compared with UxDMA [15] and IEEE 802.11 DCF [1], respectively. Section VII concludes the paper.

II. Problems in NAMA and IEEE 802.11 DCF

A. Node Activation Multiple Access (NAMA)

NAMA is a distributed channel access scheduling mechanism where the required information for conflict-free channel access is minimally reduced to the two-hop neighborhood of each node [2]. In NAMA, a hash function, similar to a pseudo random number generator (PRNG), is implemented at each node. The hash function takes a distinctive string of a node as input, and derives a random priority for each neighbor within two hops. The distinctive input string is the concatenation of the corresponding node identifier and the current time slot number such that the priority changes in different time slot.

The channel access eligibility of each node is then determined by the node comparing its own priority with those of its two-hop neighbors. If a node has the highest priority, the node can access the channel within the corresponding time slot, while its two-hop neighbors are forbidden from channel access because they have lower priorities than the node.

The advantage of NAMA is that NAMA completely eliminated the communication overhead with regard to building the dynamic channel access schedule, except for collecting the two-hop neighbor information, which is minimal comparing with the complete network topology information.

However, the application of NAMA in regular ad hoc networks was unsuccessful because

1. The random priority generation mechanism provides no bound to the channel access delay because a node may probabilistically derive low priority for a long period of time and never get access to the channel.

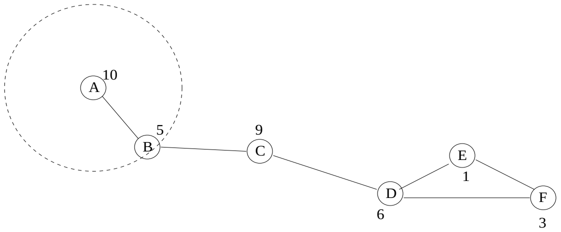


Figure 1. Chain Reaction in Channel Access Contention Resolution

2. There may be chain effects to the channel access opportunities, in which the priorities of nodes cascades from high priority to low priority across the network, and only a limited number of nodes are able to access the channel. Figure 1 illustrate one of such phenomena where the node priority numbers happen to cascade from high values to low ones, thus disallowing all but one node to access the channel. The phenomenon dramatically decreases the channel throughput.
3. The channel bandwidth is wasted when eligible nodes have no traffic to send.
4. The channel bandwidth may also be wasted when a node does not have data to send in the allocated time slot, or the data frame transmission duration is less than that of the time slot.
5. Because of the wasted bandwidth causing starvation to the other nodes with traffic, NAMA incurs long delay to certain applications that are sensitive to the delay, such as TCP congestion and flow control mechanisms [19], AODV route update mechanisms [13].

B. IEEE 802.11 DCF

In comparisons, CSMA/CA (carrier sensing multiple access with collision avoidance), such as MACA [9] and IEEE 802.11 DCF (Distributed Coordination Function) [1], utilizes smaller time slots, and apply a random access scheme with *Binary Exponential Backoff* (**BEB**) mechanism to resolve collisions. According to BEB mechanism, when channel collision happens, the colliding nodes exponentially increase the backoff window size so that the colliding nodes delay their channel access for longer period of time and the collision probability dramatically decreases for each nodes. Nonetheless, there is a *non-negligible* probability that collisions happen again. Therefore, neighbor-aware channel access deferral and backoff mechanisms are required in order to improve the channel good-throughput. However, except for adjusting the backoff window size, there has not been any deterministic mechanism so far to avoid possible collisions.

III. Latin Squares

A *Latin square* of order n is an $n \times n$ square matrix that consists of n distinct symbols $\{1, 2, \cdots, n\}$, where the symbols of each row and column are also distinct. For instances, the follow matrices A and B are two Latin squares of order 4.

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{bmatrix} \quad B = \begin{bmatrix} 4 & 1 & 2 & 3 \\ 3 & 2 & 1 & 4 \\ 1 & 4 & 3 & 2 \\ 2 & 3 & 4 & 1 \end{bmatrix}.$$

The number of different Latin squares beyond even a very small order, such as 10, grows almost to infinity for practical applications. For instance, the number of Latin squares of order 10 is about 10^{40}. There are some simple ways to generate Latin squares. In fact, if one take any finite group G of order n, then the multiplication table of G is a Latin square of order n [11].

The study of Latin squares provides an environment that is rich in various problems, applications and results of such fields as algebra, finite geometrics, coding theory, combinatorial design theory and statistics. Using Latin Squares for conflict-free access to parallel memories was proposed [11].

The distinctiveness and orderliness of each symbol in the Latin square present very attractive features in applying Latin squares to the scheduling problems. As an example, Ju and Li proposed topology-transparent channel access scheduling approach based on Latin Squares for ad hoc networks with multiple transmission channels [8]. The TDMA scheduling algorithm in [8] maps a $p \times p$ Latin square onto an $M \times p$ time division multiple channels, where the number of channels is M, and the number of time slots in each frame is p. Then it assigns a unique symbol from the Latin square to each node, and the positions of the symbol in the Latin square corresponds the time slot assignments to the node.

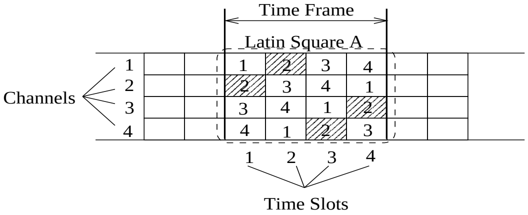


Figure 2. Multi-Channel TDMA Time Frame Structure.

For example, in Figure 2, a network with four nodes has four channels for transmissions, and each time frame consists of four time slots. Each node is assigned a symbol from $\{1, 2, 3, 4\}$, and the time slots corresponding to the symbol positions are assigned to the node. For instance, if a node is assigned symbol 2, it can access the 1st channel in time slot 2, the 2nd channel in time slot 1, the 3rd channel in time slot 4, and the 4th channel in time slot 3, shown as shaded blocks in the figure.

In addition, it is assumed that each node is equipped with p receivers for all the channels, and one transmitter to send data frames in the assigned time slots. Each data frame is transmitted for p times in each time frame. This way, the number of nodes in the network can be up to p^2.

In the following sections, we demonstrate how Latin squares are utilized in macro-time and micro-time division schemes for multiple medium access control. Essentially, we

- apply Latin square in computing the node priorities used in NAMA [2], instead of randomly generating the priorities according to the hash function. Such an adaptation improves the delay bound issue and the chain reaction mentioned in section II.
- apply Latin square symbols to determine the backoff intervals in random access schemes. The time slot size is comparable to the round-trip-time of point-to-point wireless communications. In IEEE 802.11 DSSS, the slot time is set to $20\mu s$. Such an adaptation of time slot size in scheduled channel access improves channel utilization efficiency due to the incompatible time slot size and packet size, and reduces delays in scheduled channel access.

IV. Multiple Access Based on Latin Squares (MALS)

A. Topology Assumptions

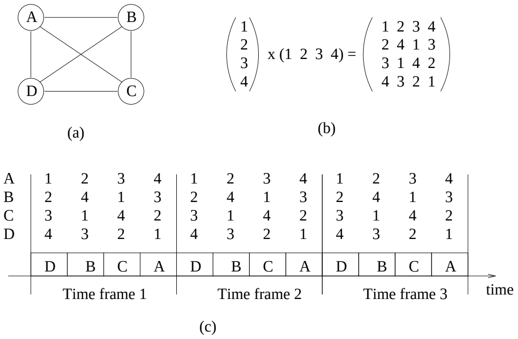
We assume that the network nodes share a single wireless channel for communication purposes, similar to the operations in networks based on IEEE 802.11. For simplicity, the topology of the network is represented as an undirected graph $G = (V, E)$. V is the set of nodes, each mounted with an omnidirectional 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. In contrast to NAMA [2], MALS does not require a node have the topological information within two hops. Instead, only a conflict-free assignment of a Latin square index is required for each node, which is guaranteed by the neighbor protocol.

In ad hoc networks, the concepts about three radio ranges are essential in understanding CSMA and CSMA/CA, namely the transmission range R_{tx}, the carrier sensing range R_{cs} and interference range R_i [21]. Transmission range R_{tx} represents the range within which a packet is successfully received if there is no interference from other radios. The transmission range is mainly determined by transmission power and radio propagation properties (*i.e.*, attenuation). Carrier sensing range R_{cs} is the range within which a receiver detects the carrier as busy if there is a transmission. Interference range R_i is the range outside the carrier sensing range, and within which the radio signal appears as noise at the receivers.

Carrier sensing range R_{cs} is an adjustable parameter in the antennas, and is usually set to 2 to 2.78 times R_{tx} [21]. Therefore, the two-hop distance in ad hoc networks is a good approximation of the carrier sensing range, and node activation scheduling usually requires all neighbors of a node within two hops be silent when the node transmits.

B. Macro-Time Division Channel Access

MALS for macro-time division multiple access achieves conflict-free node activation channel access for each node within its two-hop neighborhood.



(a) (b) (c)

Figure 3. A Sample Network and Corresponding Latin Square

To describe the basic mechanism in MALS, we start with a Latin square for a simple network as shown in Figure 3(a) for macro-time division multiplexing. As mentioned before, each time slot lasts long enough to contain the transmission of a complete data packet in the macro-time division schemes.

In Figure 3, the Latin square is generated using *modulo multiplication group*, representing the residue class $\{1, 2, 3, 4\}$ prime to 5 under multiplication mod-5. An element a_{ij} in the Latin square corresponds to the mod-5 product of ith and jth elements of the two identical sequences $(1, 2, 3, 4)$, as shown in Figure 3 (b).

Different from NAMA, each node is assigned a row index of the Latin square in Figure 3(b) as shown in the first time frame of Figure 3(c). In Figure 3(c), each column in the Latin square corresponds to a time slot in the time frame. The Latin square value corresponding to the current time slot in each row is assigned to the corresponding node as its priority. Since all numbers are distinct in each row and each column, there is a unique winner in each time slot to access the channel.

Figure 3 exemplifies the MALS algorithm for macro-time division scheduling. To increase fairness assigning the priorities according the Latin squares, the two multiplying sequences could also dynamically change from time frame to time frame such that the Latin square also changes. For example, two sequences $(1, 2, 3, 4)$ and $(2, 3, 1, 4)$ will generate a totally different Latin square according to Figure 3 (b).

Generally in MALS, each node is assigned an index of a row in a dynamic Latin square, and the columns of the Latin square corresponds to the time slots of a time frame in the channel. The symbol value in the Latin square for each time slot determines the priority of the node in channel access resolutions. Because of the distinctiveness of the symbol value in each column, a unique priority is assigned to each node, and the channel access at any moment is guaranteed collision-free.

Because each node is assigned a unique Latin square row index, MALS requires that two nodes within two-hop neighborhood cannot be assigned the same Latin square index to avoid channel access conflicts. Corresponding to graph coloring problem, the order of the Latin square utilized in MALS needs to be at least the minimum number of colors in the graph coloring solution.

C. Micro-Time Division Channel Access

In the previous section, MALS resembles NAMA in time division scheme and priority-based channel access resolution mechanism, where a node with the highest priority number is granted access to the channel for one time slot duration. However, as pointed out earlier in section II, a macro-time division channel access wastes unused time slots, and incurs long delays for successive channel accesses. In addition, a node has to know its two-hop neighbors and their assigned Latin square indexes to determine its eligibility to access the channel.

We solve the limitations in MALS for macro-time division multiple access scheme by reducing the time slots to a comparable duration of the round-trip-time of point-to-point communication in ad hoc networks. For instance, IEEE 802.11 DSSS defines a slot time of $20\mu s$. Furthermore, the channel access resolution is no longer derived by directly comparing the priority values, but by deferring the channel access by the priority number of time slots.

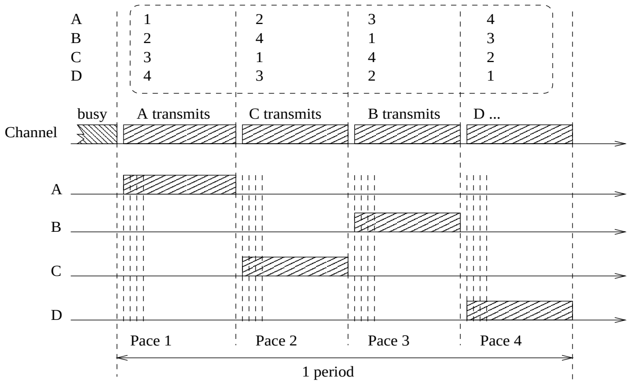


Figure 4. Channel Access Deferral According to Latin Squares

Figure 4 illustrates the channel access deferral mechanism in micro-time division channel access scheduling. Instead of dividing the channel according to the macro-time times, the channel is accessed by the micro-time slots. In addition, the time is divided into longer paces, according to which the Latin square advances along its columns so that the priorities assigned to each node changes. A number of paces are organized into a period after which the Latin square values are reused. In practice, the progressing pace along the Latin square row can be at the level of milliseconds, and requires coarse synchronization between nodes to keep collision-free channel access.

At each pace, the priority value derived from the Latin square determines the deferral interval. For instance during pace 1 in Figure 4, node A can access the channel whenever the channel is idle for one time slot. If node A does not have traffic to send, node B could defer for two time slots, and start accessing the channel. Because the order of the Latin square is equal to the number of two-hop neighbors, each node defers minimum number of time slots for accessing the channel.

By adopting such time slots, MALS can be integrated with IEEE 802.11 by replacing the DCF function. Essentially, the binary exponential backoff (BEB) intervals are substituted with the Latin square values. Whereas BEB mechanism exponentially increase the contention window when collision or packet loss occur, MALS mechanism keeps the backoff or deferral intervals according the node's current Latin square value.

Similar to section IV-B, the Latin square can be dynamically generated by changing the two sequences. The channel access pattern according to IEEE 802.11 with MALS is illustrated in Figure 5.

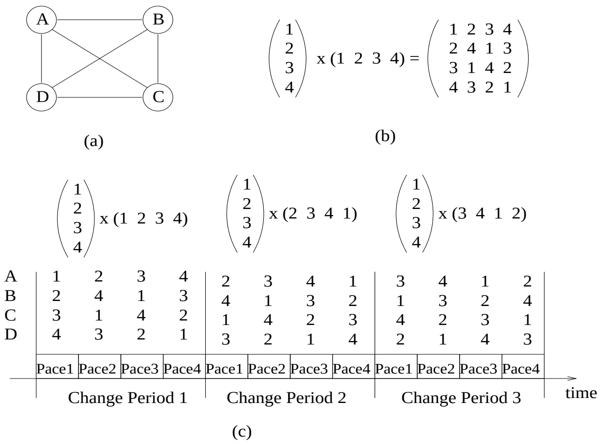


Figure 5. A Sample Network and Dynamic Latin Square

In Figure 5, each node is still assigned fixed Latin square row index. However, the Latin square changes in different "Change Periods", as shown in the figure. In each "Change Period", each node's priority walks through the Latin square row in fixed "progression paces". During each "progression pace", the node backs off or defers to the channel access by the corresponding Latin square values. Because different nodes backoff or defer to channel access by different number of time slots, there is no collision in channel access.

One fact worth noticing is that the higher the priority of a node is, the less probable the node can access the channel because it defers for longer period of time.

D. Latin Square Generation

Latin squares can be generated using many algorithm. We use the modulo multiplication group to generate Latin squares of different prime number sizes, depending on the size of the network. Because prime numbers are not continuous in the integer space, there is no guarantee that the generated Latin square rows can be completely assigned to the nodes. When the Latin square rows are not used up, channel access fairness becomes an issue because some rows may have better chances to access the channel. For example, if the Latin square in Figure 3 is assigned to three nodes A, B, C, one of the nodes always gets twice as much channel bandwidth as the other two. If MALS is applied in macro-time division scheme, node C wins twice as many time slots as nodes A and B. Similarly, if MALS is applied in micro-time division scheme, node C still wins twice as many channel access chances as nodes A and B.

We need a rotating mechanism to give fair channel access chances to all the nodes in the network by dynamically changing the Latin squares. Suppose the Latin square used in channel access schedule is $n \times n$, and the generating two sequences are $a[1 \ldots n]$, and $b[1 \ldots n]$. Further suppose that a node is assigned the ith row of the Latin square, and the current "time frame" (Figure 3) or "change period" (Figure 5) is k, and the "time slot" (Figure 3) or "progress pace" is l, then the Latin square value of the node for the current moment is:

$$a[(i+k) \ \texttt{mod} \ (n-1)+1] \times b[l] \ \texttt{mod} \ n$$

E. MALS Limitations and Neighbor Protocol

From the descriptions of MALS in both macro-time division and micro-time division channel access schemes, we can notice that there are several requirements for MALS to function well. These are also the limitations of MALS comparing with IEEE 802.11 DCF.

- Synchronization. The choice of which Latin square column value of a specific row to use at any moment needs to be synchronized between nodes so that the priority comparisons or channel access deferrals are consistent.
- Latin square order determination. To maximum channel access fairness and efficiency, the minimum Latin square order should be chosen for certain networks. Such decision is a network-wide attribute.
- Latin square index assignment. The selection of certain row index of the Latin index needs to be coordinated between nodes so as to avoid conflict in channel access scheduling.

These limitations can be either solved by a central controller, such as an access point, or by distributed mechanisms such as in ad hoc networks. Ad hoc network synchronization and Latin square information coordination is handled by the neighbor protocol, as specified in section V.

V. Mobility Handling

The IEEE 802.11 standards support the peer-to-peer mode Independent Basic Service Set (IBSS). Lai *el al.* proposed very efficient algorithm that synchronizes clock within 100 μs when the number of stations is more than 300 [10]. Such accuracy is sufficient for MALS to be applied in both macro-time division and micro-time division schemes.

The critical issue in applying MALS in ad hoc networks is to determine Latin square order and distribute Latin square indexes to nodes. Specially, we only consider the micro-time division scheme where the Latin square values are used for deferral and backoff intervals.

Because ad hoc network topology is in constant change, the overheads maintaining a precise Latin square order is difficult. Therefore, a fixed Latin square order is maintained in ad hoc networks. Considering the regular neighborhood density of the networks, a reference value can be 31.

If a node does not have a Latin square index assigned, it joins the network by choosing a random channel access delay, greater than the Latin square order. This way, the transmission from the node least interferes other in-network nodes. In addition, the node also listens to the channel and determine which Latin square indexes have been taken by other nodes. After a period of time, the node starts sending "index request" packet to ask for a Latin Square index assignment. Any node hearing the request responds with a list of available Latin square indexes if the requested Latin square index is already taken. After such information collection at the newly joining node, the node would have an idea of which index to choose for channel access.

After making any choice about the Latin square index, the node reaffirm the choice with others by announcing its choice again, still using the longer random backoff interval than the Latin square order. Whenever there is no objection to such a request, the node obtains the Latin square index. Otherwise, the node has to go through another round of requesting a Latin square index, or continue using the longer backoff interval than the Latin square order.

Four types of messages are necessary in this process:

- Latin square index REQUEST message.
- Latin square index RESPONSE message.
- Latin square index CONFIRM message.
- Latin square index REVOKE message.

In mobility scenarios, if a node hears other nodes have the same transmission Latin index code, it needs to go through the same process to acquire a new Latin index code.

VI. Simulations

The simulations of MALS are carried out in two batches. The first is in a completely controlled scenario where the networks include fully-connected and multi-hop networks, and the channel is organized in macro-time division fashion, where Latin square order and index assignments are determined beforehand. The second is in a fully connected network scenarios, where the channel is organized in micro-time division fashion, the number of nodes is known, and the Latin square order and index is assigned beforehand. More complex simulation scenarios are in our current research when the neighbor protocol is in place.

A. Macro-Time Division

We simulate the performance of NAMA, MALS and UxDMA in mobile ad hoc networks using our custom simulation tools. 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. The performance of the three protocols are studied in different neighborhood density scenarios by changing the transmission ranges in the multi-hop networks. 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.

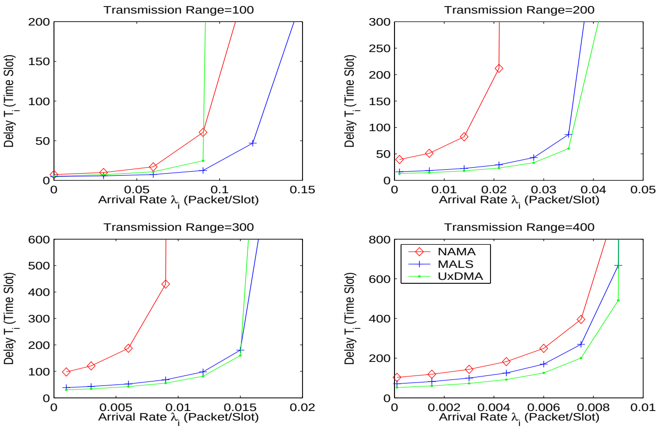


Figure 6. Average packet delay in multi-hop connected networks

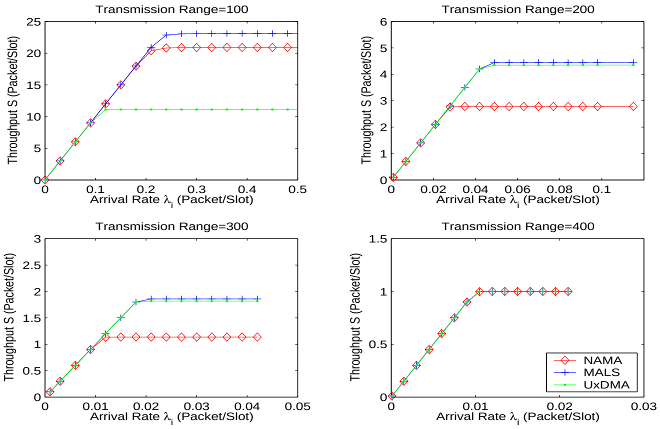


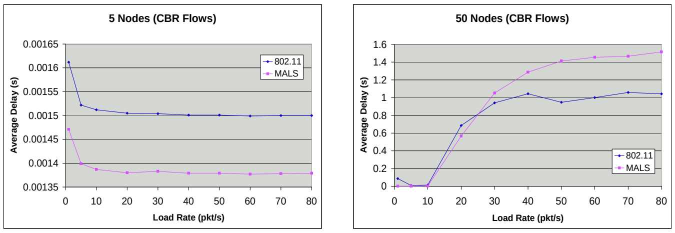
Figure 7. Average packet throughput in multi-hop connected networks

Figure 6 and 7 show the delay and throughput features of the three protocols in multi-hop networks. With respects to both delay and throughput attributes, MALS excels or equates the best performance of UxDMA and NAMA.

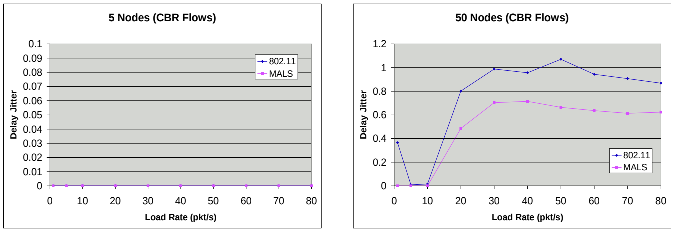
B. Micro-Time Division

MALS is integrated with IEEE 802.11, which is already implemented in NS2 [14]. MALS replaces the DCF function of the original IEEE 802.11, and provides a new backoff and deferral mechanism based on Latin squares. The networks are created such that the nodes are fully connected with each other all the time, and CBR (constant bit rate) traffic is generated randomly between nodes. 5-node networks and 50-node networks are simulated using both IEEE 802.11 DCF function and IEEE 802.11

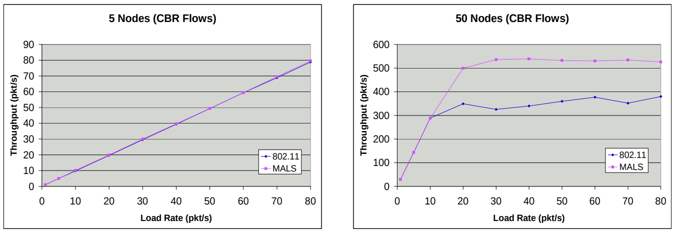
MALS, and their delay and throughput performance metrics are collected.



(a) End-to-End Delay



(b) End-to-End Delay Jitter



(c) End-to-End Throughput

Figure 8. Average End-to-End Attributes in 5-Node and 50-Node Scenarios with CBR Traffic

Figure 8 shows the average end-to-end CBR traffic attributes regarding the delay, delay jitter and overall throughput. As we can see, IEEE 802.11 MALS provides higher throughput and low delay jitters in 50-node network scenarios, although the average delay is higher than IEEE 802.11 DCF. However, when the node density is low in 5-node networks, the delay attribute in MALS is better than DCF.

VII. Conclusion

After categorizing medium access approaches into macro-time and micro-time division channel access schemes, we have presented MALS, a novel medium access protocol based on Latin squares, suitable for scheduled channel access under both schemes. MALS is applicable in both traditionally studied time division multiple access (TDMA) scheme, and random access scheme by generating deterministic schedules according to the Latin squares. Because of the inherent fairness attribute of Latin squares, MALS demonstrates obvious advantage over the random access scheme, namely IEEE 802.11 DCF.

REFERENCES

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

[2] 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.

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

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

[6] J. Denes and A.D. Keedwell. *Latin squares and their applications*. Academic Press, 1974.

[7] 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.

[8] J.H. Ju and V.O.K. Li. TDMA scheduling design of multihop packet radio networks based on latin squares. *IEEE Journal on Selected Areas in Communications*, 17(8):1345–52, Aug 1999.

[9] P. Karn. MACA - a new channel access method for packet radio. In *Proc. ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, New York, Apr. 1990.

[10] T.H. Lai and D. Zhou. Efficient and Scalable IEEE 802.11 Ad-Hoc-Mode Timing Synchronization Function. In *17th International Conference on Advanced Information Networking and Applications (AINA)*, Mar. 27-29 2003.

[11] C.E. Laywine and G.L. Mullen. *Discrete Mathematics Using Latin Squares*. Wiley-Interscience, 1998.

[12] D. Maniezzo, G. Pau, M. Gerla, G. Mazzini, and K. Yao. T-MAH: A Token Passing MAC protocol for Ad Hoc Networks. In *MedHocNet2002*, Chia, Sardegna, Italy, Sept. 2002.

[13] C. Perkins, E. Belding-Royer, and S. Das. RFC 3561 - Ad hoc On-Demand Distance Vector (AODV) Routing. Technical report, Internet Engineering Task Force (IETF), Jul. 2003.

[14] VINT Project. The Network Simulator - ns-2. http://www.isi.edu/nsnam/ns/.

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

[16] R. Ramaswami and K.K. Parhi. Distributed scheduling of broadcasts in a radio network. In *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, volume 2(3), pages 497–504, Ottawa, Ont., Canada, Apr. 23-27 1989. IEEE Comput. Soc. Press.

[17] R. Rozovsky and P. R. Kumar. SEEDEX: a MAC protocol for ad hoc networks. In *Proc. of the 2nd ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) 2001*, pages 67–75, Long Beach, CA, USA, Oct. 4-5 2001.

[18] Jochen Schiller. *Mobile Communication*. Addison Wesley, 2nd edition, 2003.

[19] T. Socolofsky and C. Kale. RFC 1180 - A TCP/IP Tutorial. Technical report, Internet Engineering Task Force (IETF), Spider Systems Limited, Jan. 1991.

[20] Z. Tang and J. J. Garcia-Luna-Aceves. Hop-reservation multiple access (HRMA) for ad-hoc networks. In *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, New York, New York, Mar. 1999.

[21] K. Xu, M. Gerla, and S. Bae. How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks? In *IEEE Global Telecommunications Conference (GLOBECOM)*, volume 1, pages 72–77, Nov. 2002.

[22] T. You, C.H. Yeh, and H. Hassanein. BROADEN: an Efficient Collision-Free MAC Protocol for Ad Hoc Wireless Networks. In *28th Annual IEEE International Conference on Local Computer Networks*, Bonn/Knigswinter, Germany, Oct. 20-24 2003.

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