Macrothink Institute™

Network Protocols and Algorithms
ISSN 1943-3581
2009, Vol. 1, Issue. 1

New Channel Access Coordination Functions in Large Scale Wireless LAN Systems

Lichun Bao

Dept. of Computer Science, University of California, Irvine

3019 Donald Bren Hall, Irvine, California 92697-3435, United States

E-mail: lbao@ics.uci.edu

Shih-Hsien Yang

Dept. of Computer Science, University of California, Irvine

3019 Donald Bren Hall, Irvine, California 92697-3435, United States

E-mail: shihhsy@ics.uci.edu

Abstract

Large-scale wireless LAN systems based on the IEEE 802.11 standards have become the most successful wireless networks deployed in large organizations, such as educational campuses and corporate warehouses. However, the suite of IEEE 802.11 MAC protocols, including DCF, PCF and EDCA mechanisms, was unable to meet the challenges to provide collision freedom and differentiated quality of services. We propose a channel access scheduling protocol based on Latin squares, called DCLASS (Distributed Coordination based on LAtin SquareS), that provisions a set of highly desirable features to large scale wireless networks with stringent performance demands. DCLASS is scalable, fair, and co-exists with IEEE 802.11 nodes with traditional DCF mechanisms. The performance of DCLASS is evaluated using an IEEE standards compliant simulation toolkit, QualNet 4.5, in distributed WLAN systems. The experiment results show the near-optimum performance of DCLASS in contrast to IEEE 802.11 DCF under various scenarios.

Keywords: Wireless network, Channel access scheduling, WLAN, Coordination function, Clustering, Latin square

1. Introduction

Large-scale wireless network deployments using commodity wireless devices have become feasible with wide range of solutions, standardized by IEEE [1][2][3][4] and ETSI [5]. These technologies provide a cost-effective and performance adaptive networking paradigm to deliver ad hoc and last-mile Internet wireless services in the forms of MANETs (mobile ad-hoc networks), WMNs (wireless mesh networks), VANETs (vehicular ad-hoc networks). In this paper, we focus on large-scale WLAN systems based on IEEE 802.11 standards [1] because of their low per-unit cost, flexibility for reconfigurations and wide acceptance in both end-users and the research community, and address the performance issues with regard to the channel access control functions.

Channel access control functions are mainly categorized into the following channel access and packet multiplexing schemes:

1) Random channel access and statistical multiplexing, such as CSMA, CSMA/CA, pure and slotted ALOHA, which were most extensively used and studied. For instances, MACA, MACAW [6], IEEE 802.11 DCF [1], PAMAS [7]. However, the commonly used random backoff mechanism along with CSMA/CA has constantly failed the performance goals in highly demanding environments, such as real-time and multimedia applications.

2) The scheduled channel access and fixed multiplexing, using FDMA, TDMA, CDMA mechanisms in wireless cellular networks, GSM, UMTS and CDMA2000 systems [8]. Scheduled channel access problems are commonly modeled as graph coloring problems, which is a well-known NP-complete problem [9]. Polynomial algorithms exist for graph coloring problems using centralized computations [10] or in-band signaling mechanisms [11]. Several topology-independent channel access mechanisms provided constant-time algorithms that combat collisions by repetitions [12]. NCR [13], SEEDEX [14] and RRMS [15] have been proposed to apply pseudo random number generators to create node orders for channel access. Although the scheduled channel access guarantees the transmission reliability, the waste of unused resource allocations is a perplexing problem.

3) The hybrid reservation based packet transmission scheme, in which the channel is separated in two virtual components, one is for control purpose to allocation channels using random access mechanisms, and the other is for data packet transmissions [16][17][18][19]. However, in a distributed wireless network, negotiation based channel resource allocation and channel access often incur overhead and cannot completely eliminate conflicts.

The current WLAN systems are mostly based on the IEEE 802.11 standards, which have grown to encompass the most advanced communication technologies, such as FHSS (frequency hopping spread spectrum), DSSS (direct sequence spread spectrum), OFDM (orthogonal frequency division multiplexing), MIMO (multiple input multiple output). However, the MAC (medium access control) protocols in IEEE 802.11 have seldom changed since their inception [1]. In relatively small wireless networks, such as WLANs, where nodes can carrier sense each other's transmission, CSMA (carrier sense multiple access) works most efficiently. Beside the optional and uncommon implementation of PCF (point coordination function), only two coordination functions, DCF (distributed coordination function) and EDCA (enhanced distributed channel access), are widely discussed and deployed, both based on CSMA/CA mechanisms, using RTS, CTS, ACK control messaging, various inter-frame spacings (IFSs) and binary exponential backoff (BEB) mechanisms. Unfortunately, the BEB mechanism does not eliminate packet collisions due to its randomness in deriving the backoff intervals, and the WLAN network performance is susceptible to throughput and fairness degradation under various channel conditions, unless fair scheduling mechanisms are implemented.

New coordination functions that comprehend the various requirements of the wireless applications are long sought after and overdue [20]. To achieve better throughput and fairness to the DCF function, Bao [21] proposed a medium access scheme based on Latin squares [22], called MALS (Medium Access based on Latin Squares), from which deterministic channel access schedules are generated for nodes in ad-hoc networks. In MALS, Latin square is mapped to channel resources in terms of time per node. However, when we apply MALS in WLAN systems for channel access control purposes, the current MALS lacks sufficient descriptions to address the following problems:

- how it handles the hidden terminal interference issues in large scale WLAN systems;
- how channel resources are shared among different BSSs (basic service sets);
- how it fits into existing IEEE 802.11 systems seamlessly for co-existence purposes;
- how it handles the channel resource allocations regarding asymmetric up-link and down-link traffic loads.

We propose the DCLASS (Distributed Coordination based on LAtin SquareS) protocol in this paper that addresses the aforementioned problems in MALS for applications in large scale WLAN systems, and provide scalable, fair and seamlessly co-existing channel access scheduling solutions for the WLAN systems. Namely, we apply these advanced techniques to improve the capabilities of MALS for such environments — traffic adaptive channel resource allocation, Latin square scaling, Latin square interleaving and extended DCF mechanisms for the co-existence of DCLASS and DCF. The prominent mechanisms and contributions of DCLASS include the following:

- Latin square applications for distributed and deterministic channel access scheduling.
- Seamless integration of DCLASS with IEEE 802.11 DCF function.
- Channel hopping algorithms for multi-channel operations in large scale WLAN systems.
- Inter-BSS coordination functions that are within the same collision domain.
- BSS clustering algorithm for isolating collision domains in large scale WLAN

systems.

The rest of this paper is organized as follows. Section 2 introduces the Latin square concept and its application in MALS. Section 3 presents the protocol details of DCLASS, including fair channel allocation, Latin square scaling, interleaving algorithms, and DCF co-existence mechanisms. Section 4 shows the throughput and end-to-end delay of our schemes in contrast to other channel access protocols using simulations. Section 5 concludes our paper.

2. Latin Squares

A *Latin square* of order n is an $n \times n$ square matrix that consists of n symbols $\{1, 2, \ldots, n\}$, in which the symbols of each row and column are also distinct. 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}.

Extending the Latin square concept, a *k-dimensional Latin hypercube of order n* is a k-dimensional array $H_k = [h_{i1,i2,\cdots,ik}]$ in which each row is a permutation of symbols 1, 2, …, n (Laywine & Mullen, 1998). *Latin cube* is a 3-dimensional Latin hypercube. Although we will apply Latin hypercube when possible, only Latin squares are discussed in this paper.

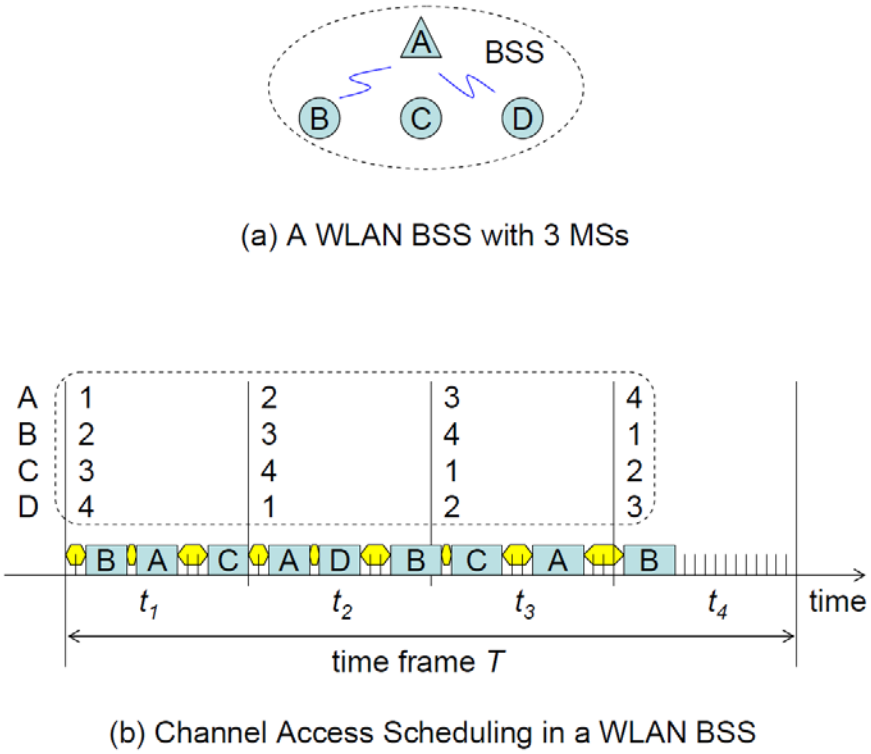
There are many algorithms to generating Latin squares and Latin hypercubes [22][23]. The one that we adopt is based on the modulo multiplication group theory because of its adaptiveness and simplicity of computations. Suppose $n + 1$ is a prime number, and vectors $A = [a_i]$ and $B = [b_i]$ ($i = 1, 2, \ldots, n$) are two permutation arrays of symbol set $\{1, 2, \ldots, n\}$, then a Latin square $L_{n\times n} = [l_{i,j}]$ of order n can be generated by multiplying A^T and B using the modulo multiplication operations:

$$L_{n\times n} = A^T \bullet B$$

$$l_{i,j} = a_i \bullet b_j \bmod (n+1), i, j \in \{1,2,\ldots,n\} \tag{1}$$

in which A^T is the transpose of A. According to Eq. (1), each permutation of vector A or B creates a different Latin square. Therefore, while the mappings of rows and columns of a Latin square to different entities in applications remain invariant, the ordering of these entities changes corresponding to different permutations of the generator vectors A and B.

Latin squares provide a convenient tool to address various problems in such fields as algebra, finite geometries, coding theory, combinatorial design theory and statistics. Using Latin Squares for conflict-free access to parallel memories was proposed [23]. In channel access scheduling problems, Ju and Li proposed topology-transparent channel access scheduling approach based on Latin Squares for WMNs with multiple transmission channels [24]. The TDMA scheduling algorithm which they proposed 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 determine the time slot number of the node. Bao proposed a simpler multiple access protocol, called MALS, which applied Latin square to replace the backoff interval calculation such that the backoff intervals directly map to Latin square symbols [21]. In MALS, the Latin square is assigned to nodes and time slots according to Latin square rows and columns, respectively.



(a) A WLAN BSS with 3 MSs

(b) Channel Access Scheduling in a WLAN BSS

Figure 1: Channel Access Deferral according to Latin Squares.

Fig. 1(a) illustrates the Latin square assignments to nodes and time slots in a small WLAN BSS with one AP *A* and three MSs *B*, *C* and *D*. In Fig. 1(b), MALS structures the time by periodic time frames, which are further divided into time slots. Then MALS assigns the stations to different rows of the Latin square, and each Latin square row corresponds to a time slots in the figure. In each time slot, the deferral of each node for channel access (represented by the yellow hexagons) is equal to the Latin square symbol in the corresponding time slot. Because of the distinctiveness of Latin square symbols assigned to the nodes in each time slot, MALS eliminates potential packet collisions from different nodes. Using this mechanism, MALS can be implemented by simply replacing the backoff computation module in IEEE 802.11 DCF. In practice, a time slot can last a few milliseconds, and the duration of a time frame can be around tens of milliseconds.

One fact worth noticing is that the smaller the Latin square symbol of a node is, the more likely the node can access the channel because it defers for a shorter period of time. Therefore, if a WLAN system has fewer nodes than the order of the Latin square, some nodes may get better chances to access the channel than the others. The solution to this unfairness issue is to use Eq. (1) to regenerate Latin squares in subsequent time frames by creating different vectors *A* and *B* in Eq. (1), thus distributing the channel access opportunities over all the nodes [21].

As we see, the channel backoff mechanism is synchronized based on the end of the previous busy period. In regular channel access, if the channel has been idle for a long time, the node should start accessing the channel by sending a busy signal burst first to synchronize the channel clock.

3. DCLASS: Distributed Coordination based on Latin Squares

3.1 Network Assumptions

For large scale WLAN systems, we assume the common ESS (extended service set) concept used by IEEE 802.11 infrastructure WLANs, in which the wireless access points (APs) are connected to the infrastructure networks for backend interconnections. Mobile stations (MSs) are associated with the individual APs that provide the best channel quality.

The multi-channel communication capabilities are desirable for large scale WLAN systems [25]. However, because of the limited number of IEEE 802.11 channels, we are eventually still faced with the co-channel interference problems as we study in this paper. Therefore, we further assume that the WLAN systems work on a single wireless channel throughout the discussions.

When WLAN systems are scaled up to wide areas, the impact of the hidden terminal problems on the performance grows dramatically, and has to be handled systematically, such as using clustering approach to isolate WLAN BSSs into collision domains, and schedule the utilization of the channel among different collision domains over the time axis. Once the BSSs are organized into smaller collision domains, the wireless transmissions within each domain can be carrier sensed by all the members of the domain, therefore avoiding the potentially hidden terminal problem effectively. However, the algorithms to isolating collision domains and scheduling the domain channel access are worth another paper of discussions, and we plan this challenge in our future work.

A *collision domain* is the set of WLAN BSSs in which every station can carrier sense the transmissions of other members in the domain. The number of BSSs in a collision domain is variable, and the channel access problems in such scenario are the focus of this work.

3.2 Asymmetric Bandwidth Assignment

In infrastructure mode WLAN systems, the bandwidth requirements are different for APs and MSs, respectively. Previous MALS solution addressed the channel access control problems in ad hoc networks, and allocated all stations the same amount of channel access resources. However, in WLAN systems, because data packets all go through the APs, and the Latin square assignment determines the channel access priorities, we need to distribute Latin square resources asymmetrically to the APs and MSs, so that the APs have more chances to forward the traffic of the MSs.

The asymmetric bandwidth allocation is achieved by assigning more Latin square rows to the APs. Because that the AP is the traffic bottleneck in each BSS, and that the traffic flows of mobile host applications are dominated by downlink transmissions in most WLAN system deployments, we allocate half of the Latin square entries for each WLAN BSS. Such an assignment guarantees that the AP gets at least half of the WLAN bandwidth when the BSS is heavily loaded.

To implement this allocation, we increased the number of Latin Square indices assigned to an AP by the number of stations associated with the AP. According to IEEE 802.11 BSS management functions, the APs with DCLASS communication capabilities attaches the Latin square generation function and channel hopping information in its periodic beacon messages in order to coordinate channel access using DCLASS. When a mobile station joins the network, it sends authentication and association requests to an AP. Even after a mobile station already joins the BSS, it can still send reassociation request to the AP without disruption its current association states. Upon receiving the association or reassociation requests, the AP assigns the MS an unused Latin square entry, and attaches the Latin square entry index to the response packet before sending it to the station. Thereafter, the MS can compute its backoff value according to the assigned Latin square index.

In addition to assigning a Latin square index to the station, the AP also assigns an additional Latin square index to itself at the same time. With multiple Latin square row assignments, the backoff interval of the AP in each time slot is derived as the minimum Latin square value of the assignments. This allows the AP to have more chance to forward the packets than any other mobile stations within the BSS.

3.3 Coexistence with 802.11 DCF

There are two reasons why we need to enable the coexistence capability in DCLASS:

- Because Latin square row assignments are centrally controlled by the APs, mobile stations have to depend on a more flexible DCF channel access mechanism to contact the AP.

- In DCLASS, the size of a Latin square is a fixed per-BSS parameter, and determines the number of stations that can be supported under DCLASS channel access paradigm. Once the Latin square rows are distributed and assigned to the AP and MSs, the DCLASS does not allow new MSs to access the channel but to hold till the BSS disassociates with existing MSs so as to reassign the released Latin square rows to the new MSs.

Therefore, a fallback mechanism is necessary in DCLASS so that nodes without any Latin square assignment may still associate with the BSS and access the channel using normal IEEE 802.11 DCF mechanisms. In order to clarify the discussions of stations operating under two channel access control functions, we define nodes that access the shared channel using DCLASS as DCLASS nodes, and those using DCF as DCF nodes.

We modify the NAV (network allocation vector) mechanism in IEEE 802.11 DCF to achieve the coexistence of DCLASS and DCF channel access mechanisms. In IEEE 802.11, NAV is a way of communicating stations reserving the channel for certain amount of time in order to exchange data and control frames, using the Duration field in the MAC header. Specifically, stations shall update their NAV with the information received in the Duration field of a received valid frame, except the NAV shall not be updated where the RA field of the frame is equal to the receiving station's MAC address.

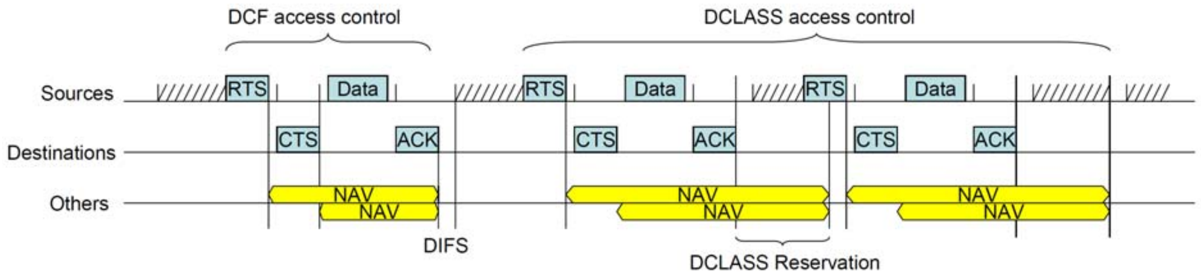
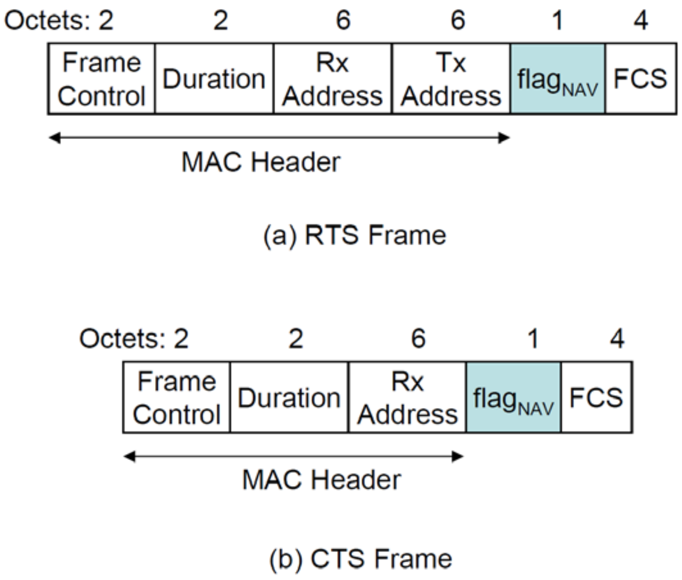


Figure 2: The NAV Mechanism for DCF/DCLASS Coexistence.

Fig. 2 illustrates the channel access activities for exchanging three data frames between three pairs of source and destination nodes, in which both DCF and DCLASS channel access functions operate. The first frame exchange sequence is carried out with the normal DCF handshakes, and the last two are based on DCLASS. In the normal DCF operations, the NAVs of overhearing stations are determined by the duration fields of the RTS and CTS frames, which end at the end of acknowledge frame in an RTS/CTS/DATA/ACK based handshake.



(a) RTS Frame

(b) CTS Frame

Figure 3: The Extended RTS/CTS Frames with NAV Flags.

Once the DCLASS takes control of the channel as shown in the second frame exchange sequence, the Duration fields of the RTS and CTS frames are modified such that the NAVs are extended for a number of time slots, equal to the size of the Latin square used by DCLASS. In addition, the modified RTS and CTS frames include an additional *flag*$_{NAV}$ field, which tells the DCLASS nodes to deduct the extended duration from their NAVs if present.

This way, the DCLASS channel access mechanisms operate before the declared NAV expires, and DCLASS nodes achieve prioritized channel access over the DCF nodes.

The prioritized DCLASS channel access continues as long as there exists other DCLASS nodes have packets to transmit. Once the NAVs determined DCLASS nodes expire, the channel access falls back to DCF mechanisms in the BSS.

In case that the NAVs are not properly set when a pair of DCLASS and DCF nodes communicate or the DCLASS frames are corrupted, collisions do occur in DCLASS. However, because we have assumed WLAN BSS scenarios, such occasions are rare. In our simulation, we discover that the performance of our schemes still converges to a stable state even when the traffic load is heavy and the network has both DCF- and DCLASS-nodes.

3.4 Multiple BSSs within the Same Collision Domain

So far, we have discussed the channel access mechanisms based on DCLASS in a single BSS, which inherently constitutes a single collision domain. In this section, we expand the single BSS WLAN system settings so that multiple BSSs are deployed within a single collision domain, and address the Latin square allocation and channel access fairness issues. Such dense WLAN system deployment scenario is common for improved network coverage and throughput.

3.4.1 Latin Square Scaling

There are two ways to apply Latin square assignments in multi-BSS scenarios. One way is to use a single Latin square for controlling the channel access among all the BSSs using DCLASS. However, such a Latin square row assignment mechanism requires a ways of dividing the Latin square rows between the BSSs, and incurs constant inter-AP coordination when mobile stations associate and disassociate with the BSSs. We discuss a second way of scaling up the Latin square assignment in multi-BSS scenarios, which we call *Latin square scaling*.

The Latin square can be scaled up using *unit matrix*, whose elements are all “1”s, and the Kronecker product. For instance, suppose that $U_{m\times m}$ and $U_{n\times n}$ are two unit matrices, $A_{m\times m} = [a_{i,j}]$ and $B_{n\times n} = [b_{i,j}]$ are two Latin squares, then the following matrix $K_{mn\times mn} = [k_{i,j}]$ is still a Latin square:

$$K_{mn\times mn} = [n(A_{m\times m} - U_{m\times m}) \otimes U_{n\times n}] + U_{m\times m} \otimes B_{n\times n} \tag{2}$$

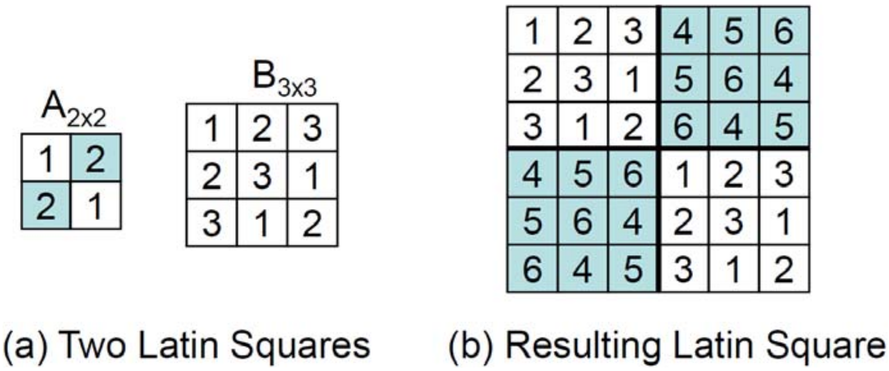
in which $\otimes$ is the Kronecker multiplication operator. Each element $k_{i,j}$ in the resulting matrix $K_{mn\times mn}$ can also be directly calculated using Eq. (3):

$$k_{i,j} = n\cdot(a_{d_i,d_j} - 1) + b_{r_i,r_j} \tag{3}$$

, where

$$\begin{aligned}
&i, j \in \{1,2,\ldots,mn\},\\
&d_i = (i-1)/n+1,\\
&d_j = (j-1)/n+1,\\
&r_i = (i-1)\bmod n+1,\\
&r_j = (j-1)\bmod n+1.
\end{aligned}$$

We call Latin square $A_{m\times m}$ the scaling Latin square, and $B_{n\times n}$ the base Latin square in Eq. (2).



A_{2x2}

1	2
2	1

B_{3x3}

1	2	3
2	3	1
3	1	2

(a) Two Latin Squares

1	2	3	4	5	6
2	3	1	5	6	4
3	1	2	6	4	5
4	5	6	1	2	3
5	6	4	2	3	1
6	4	5	3	1	2

(b) Resulting Latin Square

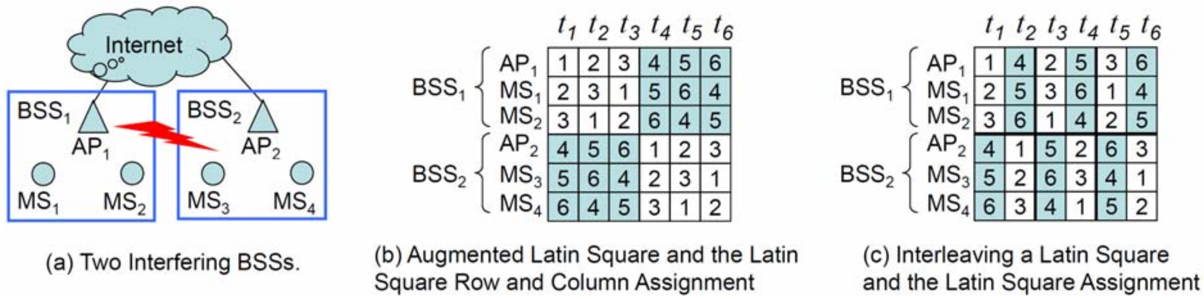
Figure 4: Latin square examples, and the scaling mechanism.

Fig. 4(a) illustrates two Latin squares $A_{2\times 2}$ and $B_{3\times 3}$, by which we generate a Latin square in Fig. 4(b) using Eq. (2).

Using the Latin square scaling algorithm, we can easily manage the Latin square assignment problem by mapping different BSSs to the rows of the scaling Latin square, and delegating the Latin square row assignments to the AP of each BSS. For instance in Fig. 5(a), two BSSs BSS_1 and BSS_2 with AP_1, AP_2, MS_1, MS_2, MS_3 and MS_4 are located within the same collision domain, and we assign the augmented Latin square in Fig. 4(b) to the BSSs and stations as shown in Fig. 5(b). Essentially in Fig. 4(b), the scaling 2×2 Latin square handles the per-BSS assignments, and the base 3×3 Latin square handles the individual station assignments within each BSS. Therefore, such an approach eliminates the need to coordinate Latin square mappings between the APs.

3.4.2 Latin Square Interleaving

One problem exists with regard to the channel access latency in Fig. 4(b). Because the Latin square symbols determines the channel access deferrals, it appears that stations in BSS_1 would occupy the channel in time slots t_1, t_2 and t_3, while unable to access the channel in time slots t_4, t_5 and t_6, which could cause long delays and high jitter under heavy traffic load in multimedia applications.



(a) Two Interfering BSSs.

		t_1	t_2	t_3	t_4	t_5	t_6
BSS$_1$	AP$_1$	1	2	3	4	5	6
	MS$_1$	2	3	1	5	6	4
	MS$_2$	3	1	2	6	4	5
BSS$_2$	AP$_2$	4	5	6	1	2	3
	MS$_3$	5	6	4	2	3	1
	MS$_4$	6	4	5	3	1	2

(b) Augmented Latin Square and the Latin Square Row and Column Assignment

		t_1	t_2	t_3	t_4	t_5	t_6
BSS$_1$	AP$_1$	1	4	2	5	3	6
	MS$_1$	2	5	3	6	1	4
	MS$_2$	3	6	1	4	2	5
BSS$_2$	AP$_2$	4	1	5	2	6	3
	MS$_3$	5	2	6	3	4	1
	MS$_4$	6	3	4	1	5	2

(c) Interleaving a Latin Square and the Latin Square Assignment

Figure 5: Latin square scaling and interleaving mechanisms for fair channel access.

In order to solve the fairness issue in terms of channel access delays among stations within the same collision domain, we propose a Latin square interleaving algorithm that shuffles and interleaves the Latin square columns, as shown in Fig. 5(c), which alternates the channel access opportunities between the two BSSs faster than before, thus further improving the fairness of the channel access mechanisms.

```
Algorithm 1: Latin Square Scaling and Interleaving
  Input: Node ID $i$, time slot number $T$, node $i$'s scaling
         Latin square row assignment $k$, node $i$'s base
         Latin square row assignment $l$.
  Output: Back-off interval of node $i$ in time slot $T$.

1 /* The time slot number in a time frame. */
  $t = (T - 1) \bmod mn + 1$;
2 /* Reverse engineering the column number in the
  scaled-up Latin square. */
  $t' = ((t - 1) \bmod m) * n + (t - 1)/m + 1$;
3 /* The column number in the base Latin square. */
  $t_m = \lfloor (t' - 1)/n \rfloor + 1$;
4 /* The column number in the scaling Latin square. */
  $t_n = ((t' - 1) \bmod n) + 1$;
5 /* The symbol value in the scaled-up Latin square. */
  $\texttt{backoff}_i = n * (L_{m \times m}[k, t_m] - 1) + L_{n \times n}[l, t_n]$;
```

To describe the Latin square scaling and interleaving algorithms, we denote the scaling Latin square as $L_{m\times m}$ of order m, and the base Latin square as $L_{n\times n}$ of order n. Further, suppose that the BSS associated with a station is assigned with the k-th row of the scaling Latin square $L_{m\times m}$, and the station is assigned with the l-th row of the base Latin square $L_{n\times n}$, then the Latin square symbol for the station in time slot t is computed according to Algorithm 1.

Algorithm 1 specifies the backoff interval calculation algorithm for a node i, given the current time slot number and node i's assignments in the scaling and base Latin squares, using Eq. (2). Because the Latin squares are applied to coordinate the channel access according to time frames, we first derive the current time slot number t within time frame boundaries on line 1. Because Algorithm 1 uses the interleaved Latin square scaled up from two Latin squares according to Eq. (2), we recover the Latin square column number of the non-interleaved Latin square on line 2. Afterward on lines 3-4, the Latin square column numbers corresponding to the base and scaling Latin squares are derived, respectively. Note the floor operation in order to calculate t_m. After applying Eq. (2) using the base and scaling Latin square assignments on line 5, we derive the backoff interval for node *i* in time slot *T*.

The orders of the base Latin square and scaling Latin square are system-wide parameters. Because the WLAN systems can be closely administered by a single authority, and are topologically stable, choosing static Latin square parameters is not a great concern in DCLASS deployments and applications. When a mobile station joins one of the BSSs in the network, the AP of the BSS will now also tell the station its row assignments in both the base and scaling Latin squares in the association response packet, and each station will compute its backoff interval values according to Algorithm 1.

3.5 Large-Scale WLAN Systems with Multiple Collision Domains

So far we have describe the channel access mechanisms for multiple BSSs that are located within a single collision domain, and that use the same shared channel for communication purposes.

In large-scale WLAN system deployments, multiple collision domains exist over a large area, and a channel coordination mechanism is required to schedule the BSS activation order among the distributed collision domains. Because the channel access can no longer depend on carrier sensing to avoid collisions between different collision domains, a high level TDMA (time division multiple access) schedule has to be applied, in which the time is segmented into time slots, and the collision domains are assigned to the time slots such that no interfering collision domains are activated at the same time.

We model the WLAN networking systems using carrier sensing graphs, *i.e. CS-graph*. That is, a large scale WLAN system is abstracted as a CS-graph $G_{CS} = \{V, E\}$, in which *V* consists of nodes, each representing a BSS, and E consists of edges (*u*, *v*), which connects two BSSs *u* and *v*, indicating that stations in the two BSSs can mutually carrier sense the transmissions of each other. In large-scale WLAN systems, such carrier-sensing relationship can be accurately modeled using signal measurements, or by the actual geographic location information of the BSSs. Because the channel access activities are moderated by the APs of the BSSs, we model the carrier-sensing relationship between two BSSs with an edge if their APs are located less than the carrier sensing range. In addition, we assume the availability of such a graph for the WLAN systems under study in this paper.

Given a CS-graph, the channel access scheduling problem is to allocate BSSs to the shortest time superframe such that adjacent BSSs are never activated in the same time slot and channel. We derive the TDMA schedule using graph coloring algorithm in three steps.

1) We apply the graph coloring algorithm on the CS-graph to figure out the maximum number of colors required for channel access scheduling using the TDMA scheme. Here the number of colors represents the number of exclusive accesses within a time superframe, which could be different time frames or different channels in the same time frame.

2) We apply a new clustering algorithm to reduce the number of colors derived in the previous step, because less colors in the CS-graph implies more channel reuse and lower channel access latencies. Since our DCLASS can handle the channel access problem within a single collision domain, and a collision domain can contain multiple BSSs, we reduce the number of colors by merging the some colors with others.

3) If the number of colors is equal to the number of channels, each and every BSS can always be activated on one of the channels, and there is no need to timesharing the limited number of channels. Otherwise, we apply the time frame and channel mapping algorithm based on Latin squares to assign the BSSs of different colors to different time frames for activations.

3.5.1 Graph Coloring and Clustering

The graph coloring on the CS-graph follows a simple greedy algorithm which finds the lowest color number for a node that has not been taken by its one-hop neighbors in the graph. Or, starting from the graph model G of the WLAN system, we could also apply one-hop graph coloring using the generic algorithm UxDMA [10], in which adjacent nodes are always assigned different colors, and a minimum number of colors is derived.

After the CS-graph is colored, we group the BSSs into different clusters based on the collision domains. A collision domain was defined as a set of stations that interfere with each other's transmissions. That is, the BSSs that are located within each collision domain are fully connected with one another, and form a clique of the graph in the corresponding interference graph. As we mentioned before, BSSs within a single collision domain can share a common channel without conflicts by applying Latin square scaling and slicing algorithms for Latin square row assignments and channel access scheduling.

Assuming that the BSSs of a large-scale WLAN system are grouped into non-overlapping collision domains, it is easy to see each collision domain forms a clique of the network graph. However, finding the right partitions to form the optimum clique cover that results in the minimum number of colors in the clique graph is an NP-hard problem [26].

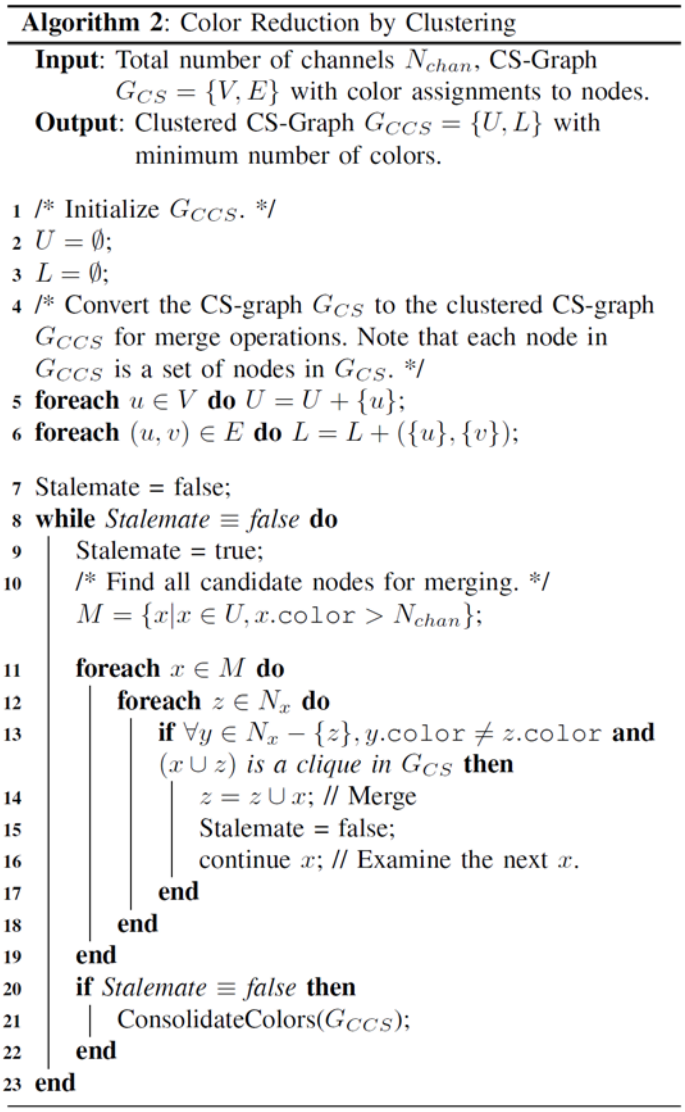
We provide a new clique coloring algorithm that reduces the number of colors based on an existing coloring of a graph, which provides both optimal coloring results and takes into account of the number of available channels. In order to describe the color-reduction algorithm, we introduce the notation in Table 1.

Table 1: Notation and Meaning

$G_{CS} = \{V, E\}$	Carrier sensing graph, in which V is the set of BSSs, and E is the set of edges, each indicating the two end-points are within carrier sensing range of each other.
$G_{CCS} = \{U, L\}$	Clustered carrier sensing graph, in which U is a set of clustered BSSs, each forming a collision domain, and L is a set of edges, each indicating that there exist two BSSs of the two end-points that could carrier sense each other.
u.color	The assigned color of a node u.

N_u	The set of one-hop neighbors of node *u*.
N_{chan}	The total number of available channels.

Algorithm 2 specifies the procedures to reduce the number of colors for channel access purposes by merging nodes in the carrier sensing graph G_{CS}. Lines 1-6 create a clustering CS-graph G_{CCS}, which is a one-to-one mapping of nodes and edges from the CS-graph G_{CS}, except that each node in G_{CCS} is a set of nodes in G_{CS}. Line 7 establishes a flag variable *Stalemate*, which terminates the program if no colors can be merged in the *while* loop. Line 10 finds all the nodes with color number greater than the number of channel, which are candidates for merging. Out of these candidates, we look at each cluster node *x*, and see if node *x* can merge with one of its one-hop neighbors *z*, provided the one-hop neighbor *z* does not share the same color as node *x*'s other one-hop neighbors on lines 11-14. If so, node *x* is merged into node *z* using node *z*'s color. Afterward, we keep looking through all candidate nodes in *M* on lines 16 and 11. If there is any merging, then potential color gaps exist between one-hop neighbors in *U*. We consolidate colors so that the color numbers are always continuous and keep to the minimum.



Algorithm 2: Color Reduction by Clustering

Input: Total number of channels N_{chan}, CS-Graph $G_{CS} = \{V, E\}$ with color assignments to nodes.

Output: Clustered CS-Graph $G_{CCS} = \{U, L\}$ with minimum number of colors.

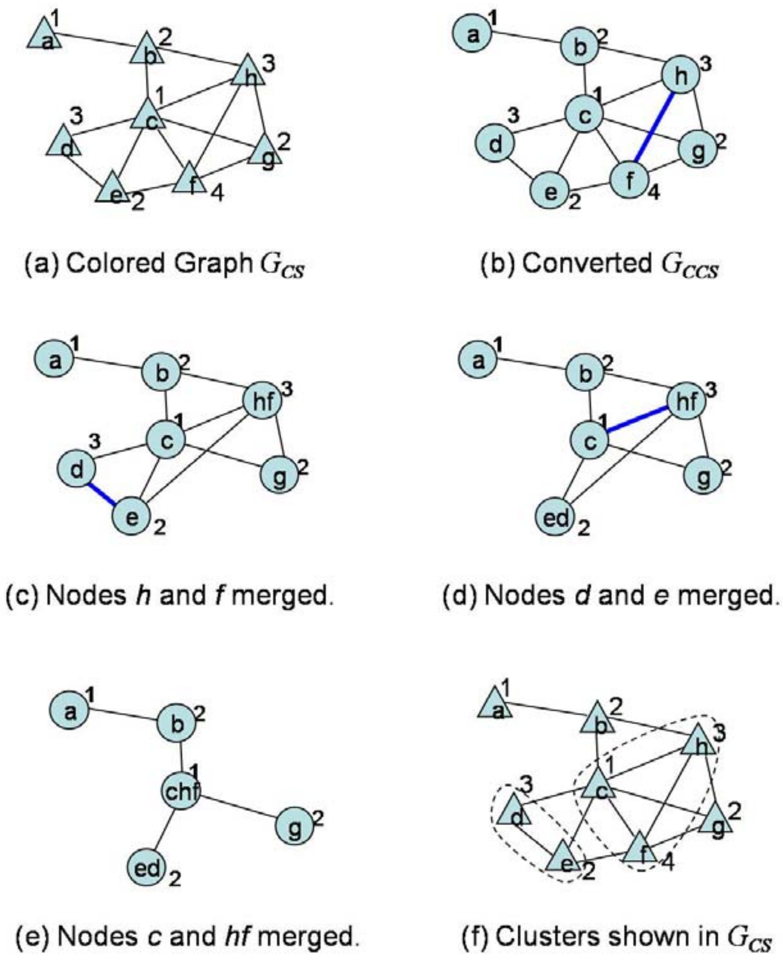
```
1  /* Initialize $G_{CCS}$. */
2  $U = \emptyset$;
3  $L = \emptyset$;
4  /* Convert the CS-graph $G_{CS}$ to the clustered CS-graph
   $G_{CCS}$ for merge operations. Note that each node in
   $G_{CCS}$ is a set of nodes in $G_{CS}$. */
5  foreach $u \in V$ do $U = U + \{u\}$;
6  foreach $(u, v) \in E$ do $L = L + (\{u\}, \{v\})$;

7  Stalemate = false;
8  while Stalemate $\equiv$ false do
9      Stalemate = true;
10     /* Find all candidate nodes for merging. */
       $M = \{x | x \in U, x.\texttt{color} > N_{chan}\}$;

11     foreach $x \in M$ do
12         foreach $z \in N_x$ do
13             if $\forall y \in N_x - \{z\}, y.\texttt{color} \neq z.\texttt{color}$ and
               $(x \cup z)$ is a clique in $G_{CS}$ then
14                 $z = z \cup x$; // Merge
15                 Stalemate = false;
16                 continue $x$; // Examine the next $x$.
17             end
18         end
19     end
20     if Stalemate $\equiv$ false then
21         ConsolidateColors($G_{CCS}$);
22     end
23 end
```

Fig. 6(a)-(f) provides an example to illustrate the steps in reducing the number of colors of a graph, assuming the number of available channels is 2. Fig. 6(a) is the CS-graph of the WLAN systems, along with coloring information. Fig. 6(b) converts the graph into the clustering CS-graph, and begins to merge nodes forming a clique. Whenever two nodes of a G_{CCS} merge, the high color number is abandoned. In Fig. 6(b)-(e), three nodes were able find cliques to merge, and we only need two colors for channel allocation purposes. Fig. 6(f) shows the cliques in the original CS-graph.

3.5.2 Channel Access Scheduling

Once the graph coloring of the clustered CS-graph is derived, DCLASS distributes the color information to the APs, respectively, including the color of the BSSs and the maximum number of colors required.

Most WLAN systems based on IEEE 802.11 have three non-overlapping channels 1, 6 and 11 for concurrent communications. Multiple channels provide more opportunities for improving the network performance from throughput, delay and jitter perspectives.



(a) Colored Graph G_{CS}

(b) Converted G_{CCS}

(c) Nodes *h* and *f* merged.

(d) Nodes *d* and *e* merged.

(e) Nodes *c* and *hf* merged.

(f) Clusters shown in G_{CS}

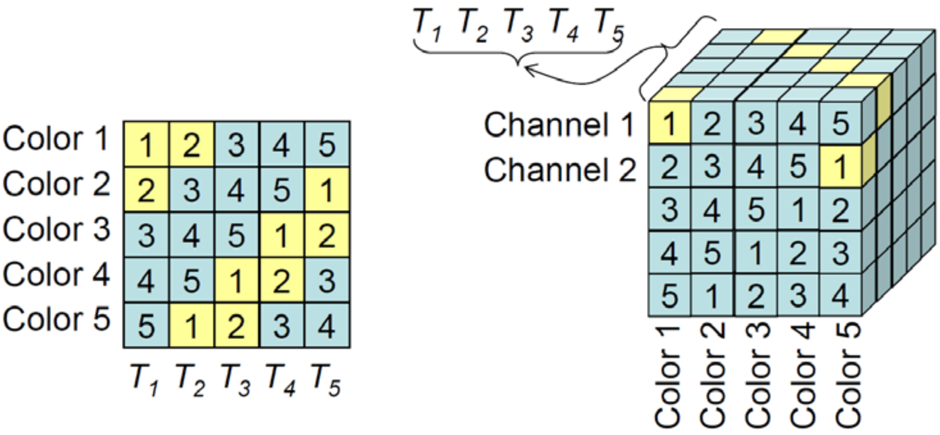
Figure 6: Color Reductions Using Clique-based Clustering.

The channel allocation and access scheduling problems in multi-channel networking environments have been extensively studied in the literature. An excellent survey of these problems and solutions in multi-channel wireless network settings were provided in [25]. The commonly used techniques include the use of a single or multiple dedicated control channels [27][28], common hopping [29] and split phase [30]. Using mechanisms that are compatible to IEEE 802.11 with one transceiver per host, MMAC uses the beacon messages to synchronize the wireless nodes over a common channel, and to negotiate the channel allocations within the Ad hoc Traffic Indication Message (ATIM) window [31]. After the ATIM window, nodes switch back to the common control channel for the next round of multi-channel access procedure.

Using the AP coloring information, the maximum number of colors and the number of available channels in the large-scale WLAN systems, each AP can independently generate their channel access schedules. There are three scenarios:

- The number of colors is equal to the number of available channels. In this case, the colors are one-to-one mapped to the channels, and each node in the clustered CS-graph G_{CCS} occupies one channel throughout the network operations. We do not elaborate on this scenario further in this section.
- The number of colors may be far less than that of available channels, which enables each node to be allocated with multiple channels. Similar to IEEE 802.11 MAC protocols which do not address such scenario, we do not address the channel access control problems in this scenario in this paper.
- The number of colors is more than the number of available channels. In such situations, we have to apply TDMA schedules to share the channels according to the coloring information.

In the second scenario, we have two methods to assign channels, namely the one based on Latin square and the other based on Latin cube.



	T_1	T_2	T_3	T_4	T_5
Color 1	1	2	3	4	5
Color 2	2	3	4	5	1
Color 3	3	4	5	1	2
Color 4	4	5	1	2	3
Color 5	5	1	2	3	4

	Color 1	Color 2	Color 3	Color 4	Color 5
Channel 1	1	2	3	4	5
Channel 2	2	3	4	5	1
	3	4	5	1	2
	4	5	1	2	3
	5	1	2	3	4

(a) Latin square based channel assignments.

(b) Latin cube based channel assignments.

Figure 7: Multi-channel Allocation in DCLASS.

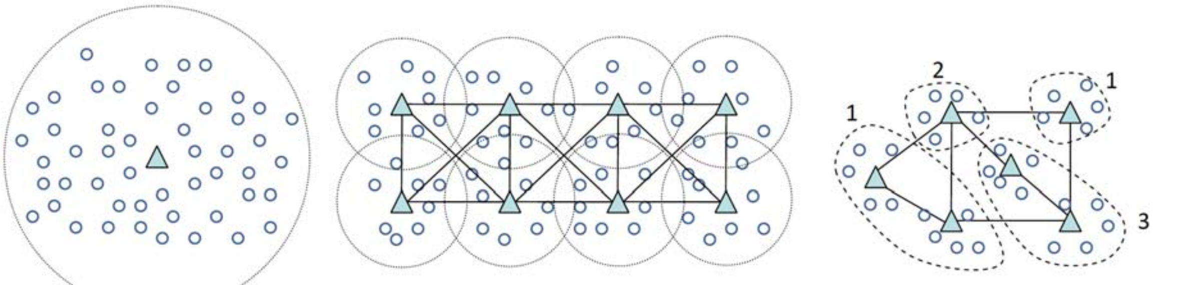
Suppose that a clustered CS-graph G_{CCS} is colored in 5 colors, and that 2 channels are available for channel allocations. Fig. 7(a) shows the Latin square based approach, which uses a Latin square of order 5. In order to assign the 2 channels to the nodes of different colors, we map the colors to the rows of the Latin square, and the time frame numbers to the columns. In each time frame, we require that the nodes colored in 1 and 2 are able to utilize channels 1 and 2, respectively, for communication purposes. For instance, nodes in colors 1 and 2 will access the channels 1 and 2, respectively, in time frame T_1.

Fig. 7(b) shows a second Latin cube based approach, which uses a Latin cube of order 5, and maps channels, nodal colors and time frame numbers to the rows, columns and layers of the Latin cube. In each time frame, channels 1 and 2 are assigned to the nodes with colors that map to Latin square symbol 1. For instance, nodes in colors 1 and 5 will access the channels 1 and 2, respectively, in time frame T_1.

Once a cluster-node in the clustered CS-graph G_{CCS} is assigned a channel, the BSSs inside the cluster can communicate using the channel access protocol, specified in Section 3.4. BSSs without any channel allocation have to remain silent during the corresponding time slots. In our implementations, the silence period of BSSs in large-scale WLAN systems is achieved by each BSS stations backing off for the duration of the time frames.

4. Evaluations

4.1 Experiment Setup



(a) Scenario 1: Asymmetric bandwidth assignment.

(b) Scenario 2&3: Latin square scaling and interleaving; coexisting with 802.11 DCF.

(c) Scenario 4: Multiple collision domains with clustering and 2 channels.

Figure 8: Simulation scenarios.

We implemented DCLASS using the QualNet 4.5 network simulator [32], as well as MALS [21], and evaluated their performance in terms of network throughput and packet delays, in contrast to IEEE 802.11b.

We used four sets of experiments and simulation scenarios, which are illustrated in Fig. 8(a)-(c). The circular dashed lines in Fig. 8(a)-(b) indicate the transmission ranges of the corresponding APs, and the irregular dashed lines in Fig. 8(c) indicate the clustered cliques.

In our first experiment shown in Fig. 8(a), we randomly deployed an AP with 50 mobile stations (MSs), and created 25 CBR (constant bit rate) traffic flows between 25 distinct pairs of MSs. The performance metrics were collected while we gradually increased the network loads from 8 Kbps (kilobits per second) to 36 Kbps per mobile station. The purpose of this experiment is to see how our asymmetric bandwidth assignment scheme improves the performance comparing to MALS and IEEE 802.11b. According to the asymmetric bandwidth assignment scheme, the order of the base Latin square is 50 × 2 + 1 = 101. The AP is assigned half of the Latin square rows, and each of the mobile stations is assigned with one Latin square row, respectively. This way, the AP has about 1/2 chance to transmit in any time slots while any MS only has 1/101 chance to transmit. The number of available channels is 1.

In the second and third sets of simulations, we created a grid topology with 8 BSSs, and each BSS contains an AP with 8 MSs. Fig. 8(b) depicts the CS-graph with APs and the MSs in this scenario. Among the 64 mobile stations, 32 distinct pairs of CBR traffic transmitters and receivers are formed. Similar to the first experiment, we gradually increase the per-node CBR traffic loads from 16 Kbps to 80 Kbps to test the performance of each protocol. The number of available channels is 1.

The scenario of the second and third set of simulation is shown in Fig. 8(b). We actually carried out two sets of tests. One examines the Latin square scaling and interleaving functionalities for inter-BSS channel access coordination, and the other examines the coexistence feature of DCLASS, in which we randomly choose 2 of the 8 BSSs to work under the IEEE 802.11b DCF mode.

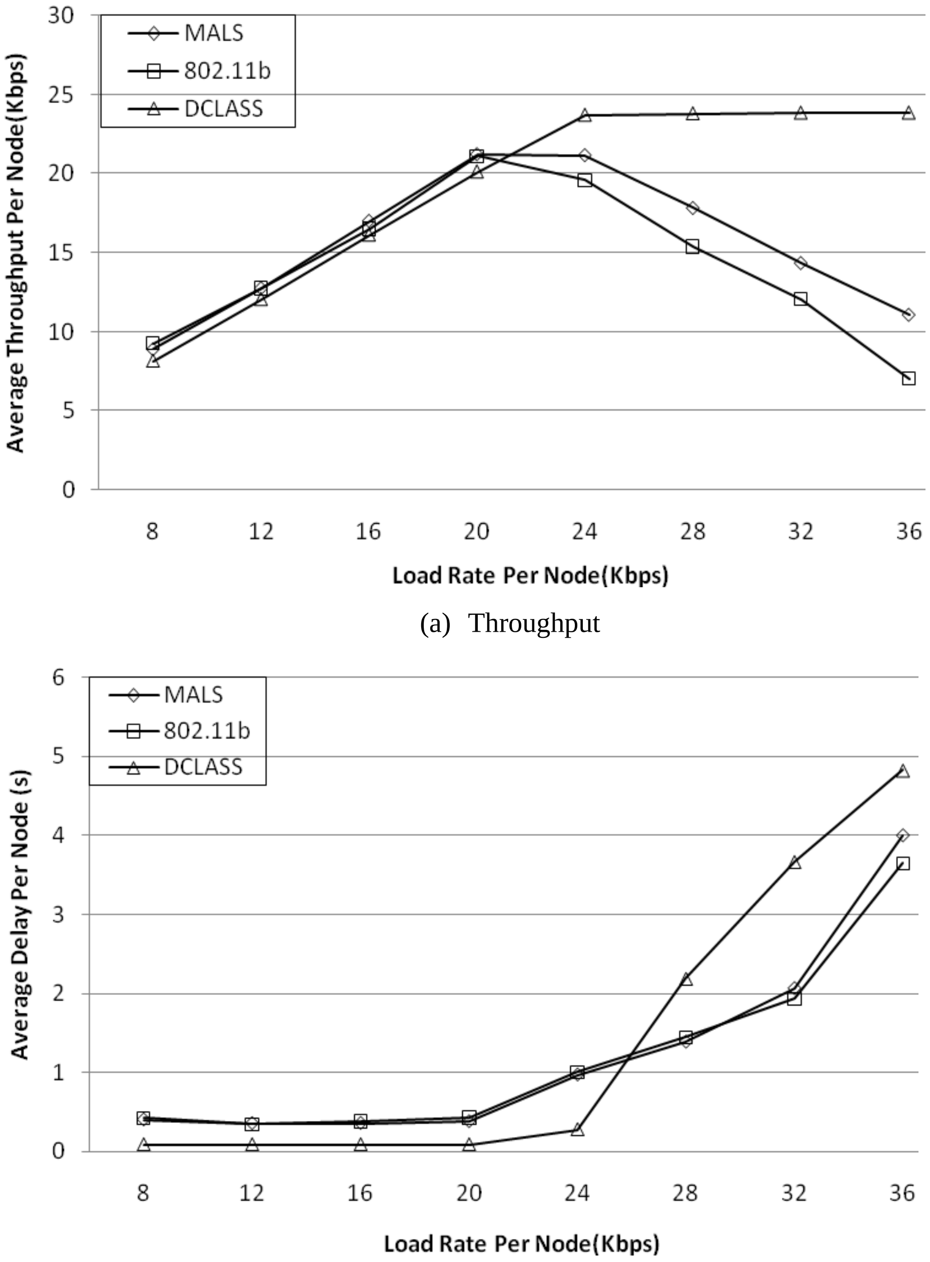
In this topology, the CS-graph of the WLAN system can be colored with exactly four colors. Therefore, in Fig. 8(b), the size of the scaling Latin square is 4. Each BSS will also have a base Latin square of size 17 according to our asymmetric bandwidth assignment algorithm because each BSS associates with 8 MSs. After we augmented the Latin square using the 17 × 17 base Latin square and the 4 × 4 scaling Latin square using our scaling algorithm, we apply the Latin square interleaving algorithm to decrease the channel access delays.

In our last set of simulations shown in Fig. 8(c), we deployed 6 APs in the field, and each AP is associated with 5 MSs. We ran the clustering algorithm (Algorithm 2) on the CS-graph, and we got four clusters which are indicated by the irregular dashed lines. The size of the base Latin square for each BSS is 5 × 2 + 1 = 11, and the size of the scaling Latin square is the maximum number of BSSs in each cluster, which is a variable. In this set of simulations, the number of available channels is 2.

4.2 Result Analysis

Fig. 9(a)-(b) show the performance of three channel access coordination functions, MALS [21], IEEE 802.11b DCF and DCLASS, in the first set of experiments. The graphs show that DCLASS outperforms both MALS and IEEE 802.11b DCF by maintaining constant network throughput while the WLAN system is stressed with overwhelming network traffic. The reason that MALS cannot sustain the network throughput is because the channel access opportunities are equally divided between APs and MSs, which neglects the asymmetric bandwidth utilization fact between APs and MSs. IEEE 802.11b DCF has the same issue, and because of higher collision probabilities, IEEE 802.11b DCF performs the worst in the three protocols.

In terms of the network delay characteristics as shown in Fig. 9(b), DCLASS maintains lower network delay than the other two protocols, but the end-to-end delay of DCLASS eventually becomes higher than IEEE 802.11b. It is because the exponential backoff scheme of IEEE 802.11b potentially grants a node more chances to transmit after a successful transmission, thus other stations will more likely have to wait for the node to finish all its transmissions. As a result, the outgoing queues for most other stations become full and start dropping packets. Therefore, IEEE 802.11b achieves better end-to-end delay by favoring a single node over others.

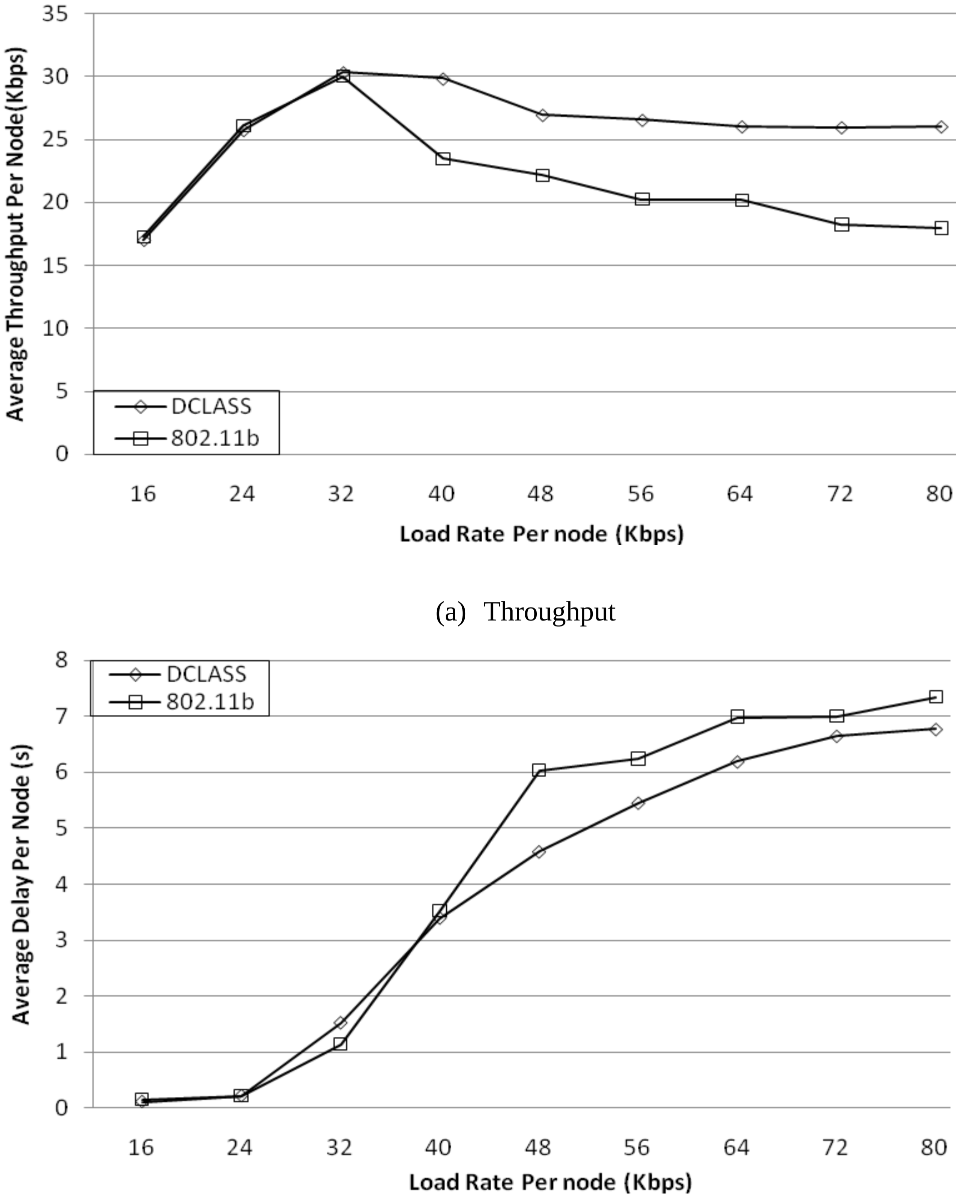


(a) Throughput

(b) End-to-end delay

Figure 9: Asymmetric bandwidth assignment.

Fig. 10(a)-(b) show the average performance of DCLASS and IEEE 802.11b in the second set of experiments with the multi-hop network shown in Fig. 8(b). When we increased the CBR load rate per node to 80Kbps, DCLASS outperforms IEEE 802.11b by 45% in terms of the average throughput, and the average end-to-end delay of DCLASS is also 8% lower than IEEE 802.11b. With respect to overall throughput and end-to-end delay, DCLASS excels or equates with the best performance of IEEE 802.11b. Due to the prioritized channel access planning using the Latin square scaling mechanisms, we have successfully avoided hidden terminal problems by always having the BSSs in the same color to access the channel, therefore enforcing spatial reuse of the channel.

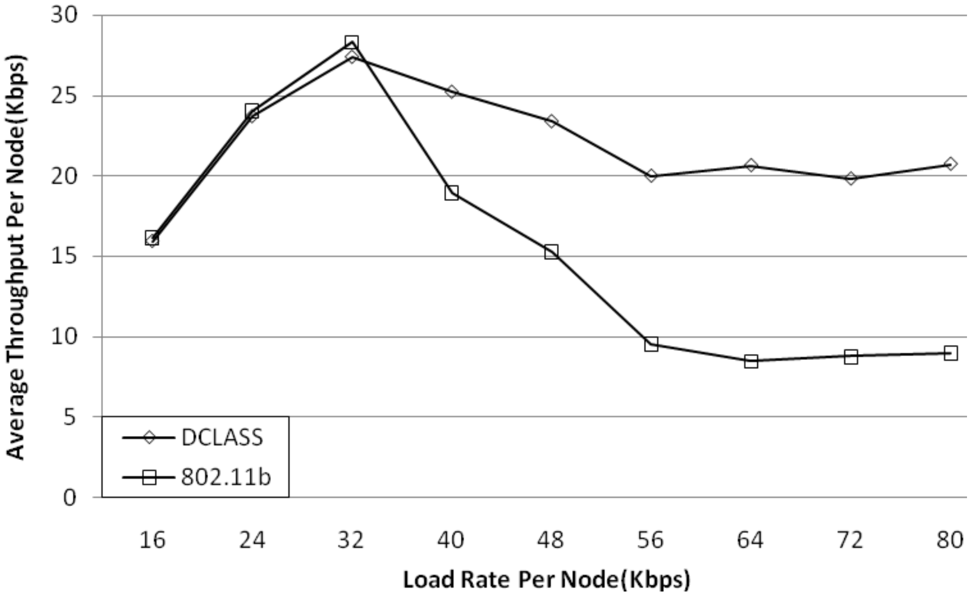


(a) Throughput

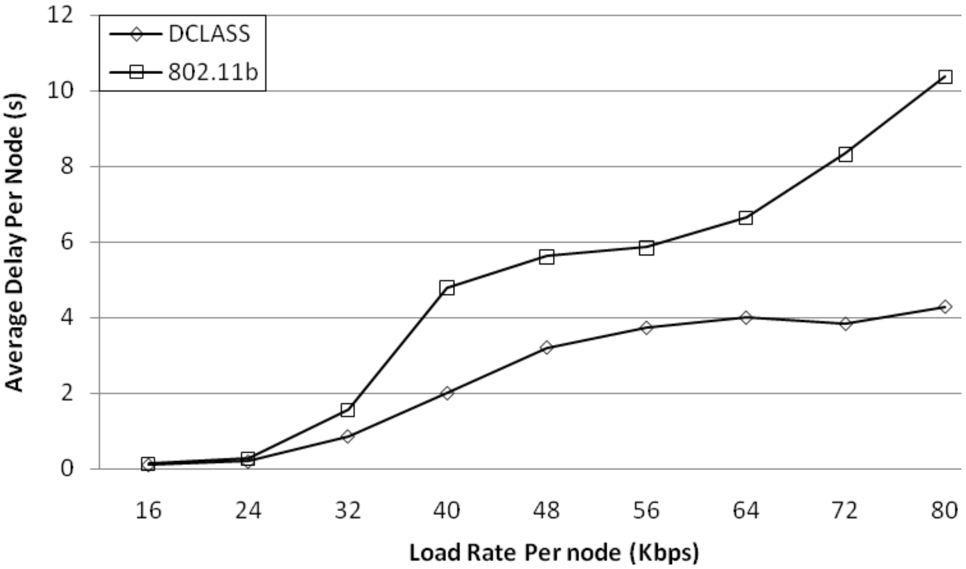
(b) End-to-end delay

Figure 10: Latin square scaling and interleaving.

Fig. 11(a)-(b) show the performance comparisons between DCLASS and IEEE 802.11b in terms of average network throughput and end-to-end delay in the coexistence scenarios. In this scenario, we simulated two kinds of traffic flows, one of which has both the source node and the destination node as DCLASS nodes, and the other mixed the source and destinations nodes between DCLASS nodes and IEEE 802.11b DCF nodes. From the simulation result that each node sends 80Kbps CBR traffic to saturate the network, we can recognize that the pure DCLASS flows achieve 131% throughput increase while the end-to-end delay is only 41% of the mixed flows.



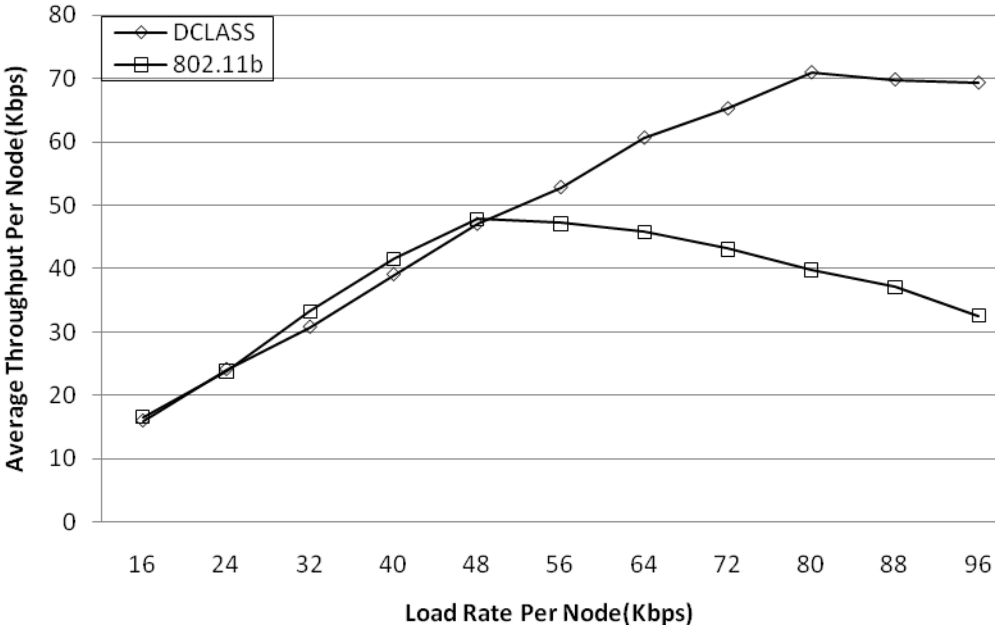
(a) Throughput



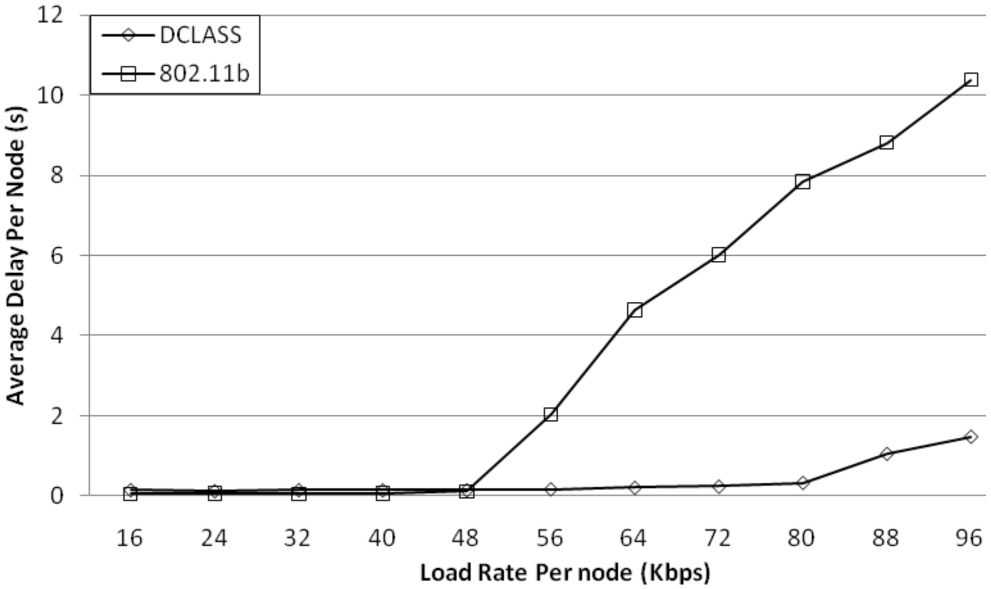
(b) End-to-end delay

Figure 11: Coexistence with 802.11b DCF simulation results.

However, one may notice that there is a period of throughput degradation in DCLASS when the network CBR loads are between 32-56 Kbps in Fig. 11(a). Such degradation is because our DCLASS scheme guarantees collision-free environment through Latin square values. When we introduce nodes which are not following the pre-defined schedule, there will be some collisions happen and decrease the performance of our schemes. The average end-to-end delay also shows the same trends as the average throughput does.



(a) Throughput



(b) End-to-end delay

Figure 12: Clustering and TDMA scheduling simulation results

Fig. 12(a)-(b) show the performance of DCLASS and IEEE 802.11b in multi-channel scenarios with 2 non-overlapping channels. Since the clustered CS-graph is colored in 3 colors, and only 2 channels are available for channel allocations, we can only activate 2 colors alternatively. The simulation results show that DCLASS achieves high throughput with low delay constraints using our TDMA-based channel assignment scheme. When the network traffic is fully saturated, the average throughput of DCLASS in this scenario is 213% of IEEE 802.11b, and the average end-to-end delay of DCLASS is only 14% of IEEE 802.11b. Note that IEEE 802.11b DCF is not capable of switching channels, therefore operates over a single channel.

5. Conclusion and Future Work

We have shown DCLASS (Distributed Coordination based on LAtin SquareS), a novel medium access control and coordination protocol based on Latin squares for large scale WLAN systems. DCLASS was specified step by step from procedures that handle single BSS communication, to complicated multi-BSS scenarios within the same collisions domains using Latin scaling and interleaving approaches. We have also introduced the CS-graph clustering algorithm for multi-channel communication capabilities in DCLASS. Simulation results show the superior performance of DCLASS in contrasts to IEEE 802.11 DCF in large scale WLAN system environments.

References

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

[2] IEEE Std. 802.15.3. Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs). Technical report, IEEE, Sep. 2003.

[3] IEEE Std. 802.16. Part 16: Air Interface for Fixed Broadband Wireless Access Systems. Technical report, IEEE, Oct. 2004.

[4] IEEE Std. 802.16e. Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands. Technical report, IEEE, Feb. 2006.

[5] 3GPP specifications. http://www.3gpp.org/specifications.

[6] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A media access protocol for wireless LANs. *Proceedings of the conference on Communications architectures, protocols and applications*, pages 212-225.

[7] S. Singh and C. Raghavendra. PAMAS: Power Aware Multi-Access protocol with Signalling for Ad Hoc Networks. *ACM Computer Communications Review*, 1999.

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

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

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

[11] 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.

[12] K. Xing, X. Cheng, L. Ma, and Q. Liang. Superimposed Code based Channel Assignment in Multi-Radio Multi-Channel Wireless Mesh Networks. In *Proceedings of MobiCom*, 2007

[13] L. Bao and J.J. Garcia-Luna-Aceves. A New Approach to Channel Access Scheduling for Ad Hoc Networks. In *Proceedings of ACM Seventh Annual International Conference on Mobile Computing and networking*, Rome, Italy, Jul. 16-21 2001.

[14] 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.

[15] J. Eshet and B. Liang. Randomly Ranked Mini Slots for Fair and Efficient Medium Access Control in Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 6(5):481–493, 2007.

[16] M. Andrews and L. Zhang. Scheduling algorithms for multi-carrier wireless data systems. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 3–14, 2007.

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

[18] 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.

[19] C. Zhu and M.S. Corson. A five-phase reservation protocol (FPRP) for mobile ad hoc networks. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, volume 1(2), pages 322–331, San Francisco, CA, USA, Mar. 29-Apr. 2 1998.

[20] I.F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks and ISDN Systems*, pages 445–487, Mar. 15 2005.

[21] L. Bao. MALS: Multiple Access Scheduling Based on Latin Squares. In *Proceedings of IEEE MILCOM 2004*, Monterey, CA, Oct. 31 - Nov. 3 2004.

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

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

[24] 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.

[25] J. Mo, H.S. Wilson So, and J. Walrand. Comparison of multi-channel MAC protocols . In *Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, 2005.

[26] R. Karp. Reducibility among combinatorial problems. In *Proceedgins Symposium on the Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[27] H.S. So, W. Walrand, and J. Jeonghoon Mo. McMAC: A Parallel Rendezvous Multi-Channel MAC Protocol. In *Proceedings of IEEE Wireless Communications and Networking Conference*, 2007.

[28] S.L. Wu, C.Y. Lin, Y.C. Tseng, and J.P. Sheu. A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks. In *International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN),* 2000.

[29] A. Tzamaloukas and J. J. Garcia-Luna-Aceves. Channel-Hopping Multiple Access. In *Proceedings of IEEE International Conference on Communications*, 2000.

[30] J. Chen, S. Sheu, and C. Yang. A new multichannel access protocol for IEEE 802.11 ad hoc wireless LANs. In *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications*, 2003.

[31] J. So and N. H. Vaidya. Multi-channel MAC for ad hoc networks: Handling multichannel hidden terminals using a single transceiver. In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, page 222-233, 2004.

[32] Scalable Network Technologies. http://www.scalable-networks.com. QualNet Network Simulator.