Large-scale access scheduling in wireless mesh networks using social centrality

Di Wu, Lichun Bao, Amelia C. Regan, Carolyn L. Talcott

Department of Computer Science, University of California, Irvine, CA, USA
Computer Science Laboratory, SRI International, Menlo Park, CA, USA

HIGHLIGHTS
- Distributed node cliques for cluster communications by social centrality metrics.
- Efficient channel access in each cluster by compact schedule based on Latin squares.
- Valid node and cluster access activation using an on-demand coloring algorithm.
- Fair radio and channel allocation to nodes and clusters in large scale WMNs.
- Coexistence of LaSo with traditional 802.11 DCF channel access mechanisms.

ABSTRACT
Wireless mesh networking is an economic and convenient way to provide last mile Internet access through ad hoc peer-to-peer communication links. However, without systematic network configuration and channel resource management, these networks suffer from scalability, performance degradation and service disruption issues due to overwhelming co-channel interference, unscrupulous channel utilization and inherent network mobility. The IEEE 802.11 DCF and EDCA mechanisms based on CSMA/CA are the most widely used random channel access mechanisms, but unfortunately these cannot effectively eliminate hidden terminal and exposed terminal problems in multi-hop scenarios. Social network analysis techniques proposed for economic and social studies have recently been shown to be a successful approach for characterizing information propagation in multi-hop wireless networks. We propose a set of efficient resource allocation algorithms and channel access scheduling protocols based on Latin squares and social centrality metrics for wireless mesh networks (WMNs) with multi-radio multi-channel (MRMC) communication capabilities, called LaSo, which can coexist with IEEE 802.11 DCF and be effectively applied in large scale WMNs. Based on interference information provided by the interference graph, LaSo uses nodal degree centrality to form cliques for intra-cluster communication, and betweenness centrality to choose bridge nodes to form cliques for inter-cluster communication in WMNs, and then applies Latin squares to map the clique-based clustering structure to radios and channels for wireless communication purposes. Afterwards, LaSo again applies Latin squares to schedule the channel access amongst nodes within each cluster in a collision-free manner. We evaluate LaSo using simulations, and results show that LaSo achieves much better performance than existing IEEE 802.11 standards and other multi-channel access control protocols.

1. Introduction
Large scale wireless mesh networks (WMNs) have become feasible for a wide range of solutions, standardized by IEEE [18,1]. Of these wireless networks, WMNs based on IEEE 802.11 standards [19] have attracted special interest due to the simplicity of channel access control functions, low unit cost, flexibility of reconfigurations and wide acceptance by end-users and the research community.

However, without systematic network configuration and channel resource management, WMNs suffer from scalability, performance degradation and service disruption issues due to overwhelming co-channel interference, unscrupulous channel utilization and inherent network mobility. In this paper, we address the following five critical challenges in order for WMNs to achieve their full potential:

© 2013 Elsevier Inc. All rights reserved.
(1) **Channel resource allocation** — The micro-wave radio spectrum in which WMNs operate is crowded by wireless users. Fortunately, multiple non-overlapping channels are available in each of the IEEE 802.11 communication standards a/b/g — around three at 2.4 GHz band, and thirty at 5 GHz band in North America [19]. As WMN nodes are increasingly equipped with multiple radio interface cards to harvest the full capacity of the channels, practical and efficient channel allocation and switching mechanisms are highly desirable to continue the success of IEEE 802.11 networks.

(2) **Medium access control (MAC)** — The MAC protocols in IEEE 802.11 include three coordination functions, DCF (distributed coordination function), PCF (point coordination function) and EDCA (enhanced distributed channel access) [18], which are all based on CSMA/CA mechanisms [19]. Unfortunately, these mechanisms are unable to guarantee basic data services due to overwhelming network congestion, severe hidden-terminal interference [33] in large-scale WMNs. More informed, scheduled and traffic-adaptive channel access mechanisms for using multiple interfaces over multi-channels are required than the simplistic CSMA/CA mechanisms in current IEEE 802.11 MAC protocols.

(3) **Multi-radio multi-channel communication (MRMC) capability** — Nodes equipped with multiple radio can communicate with multiple neighbors simultaneously over non-overlapping channels. Therefore, MRMC could reduce interference and increase network capacity. Multi-channel allocation is a challenging problem in MRMC networks. Since practically the number of radios at each node is always much smaller compared to that of orthogonal channels due to reasons such as cost and node size factors, it is not feasible to assign one fixed channel to each radio. A radio interface may need to switch to different channels in its permitted spectral domain for better performance. This radio constraint makes the channel assignment in MRMC mesh networks much harder. Furthermore, the MRMC allocation also needs to be considered with the access scheduling problem together to achieve network connectivity in the multi-hop multi-flow WMNs. Meanwhile, transmission collision and the network performance on throughputs and delay are tightly relevant to the MRMC allocation and scheduling as well.

(4) **Clustering structure in MRMC** — For the clustering purpose, we introduce the interference graph concept, in which two nodes have a link between them if they can carry-sense each other’s transmissions. In addition, we define a collision domain of a node as the set of adjacent nodes that can interfere and damage the packet receptions of the node. Nodes within each collision domain are fully connected with each other, and form a clique in the corresponding interference graph. Assuming that the nodes of a large-scale WMN 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 collision-free MRMC scheduling and guaranteed network connectivity in the clique graph is an NP-hard problem. The connectivity-based k-hop clustering problems has been discussed in [10]. A cliques solution for unit disk graphs in k-Clustering was presented in [14].

(5) **Social network analysis in WMN** — In this paper we use social network analysis techniques to form cliques of nodes for inter-cluster and intra-cluster communications in multi-hop WMNs. The deployment of WMNs is not arbitrary in real life. For example, the deployment of mesh nodes in urban scenarios, including offices, restaurants and universities, is related to population density and access frequency. Therefore, based on the social connections between human and mesh nodes, social analysis techniques may be suitable for WMN clustering and scheduling. At the spots with high density of data access to WMNs, we need to deploy more mesh routers with corresponding density to solve the congestion control and load balancing issues. The social network analysis for routing in disconnected networks have been addressed in [11]. To the best of our knowledge, our paper is the first work to use social centrality to measure and obtain good clustering results for channel access purposes in WMNs.

(6) **Coexistence with IEEE 802.11 standards** — There are many channel scheduling protocols proposed to improve the performance of medium access control, but the coexistence issues with existing IEEE 802.11 standards have been rarely mentioned for the practical application and real deployment of large-scale WMNs. Mobile clients need to depend on a more flexible channel access mechanism to contact the servers with the combined service of heterogeneous wireless networks based on the widely used IEEE 802.11 standards and other new emerging protocols. In this paper, we address the reasons that we need the random channel access scheme based on IEEE 802.11 DCF, and enable a coexistence mechanism for DCF and our MRMC protocol. Multi-radio multi-channel resource allocation and access scheduling in large scale wireless mesh networks is the focus of this paper. Because Latin squares could be used for fair and efficient channel allocation, and social centrality could be used to solve the clustering in large scale WMNs, we propose LaSo (Latin Square Scheduling in Multi-Radio Multi-Channel Wireless Mesh Networking using Social Centrality), that applies Latin squares [27] and social centrality metrics to address the aforementioned five challenges at two scales. In the macro-time scale, we design radio and channel resource allocation algorithms to provide dynamic and adaptive conflict-free radio and channel assignments to individual nodes or clusters in wireless networks; in the micro-time scale, we specify multiple channel access control protocols to provide collision-free packet transmissions within each cluster. The features and contributions of LaSo include:

- avoiding co-channel interference by isolating collision domains using two-hop interference information, and guaranteeing network connectivity using bridges;
- forming cliques of nodes for inter-cluster and intra-cluster communications using different social centrality metrics, namely degree centrality and betweenness centrality;
- achieving efficient channel access within each cluster using compact access schedules based on Latin squares;
- obtaining valid node and cluster access activation using on-demand coloring algorithm;
- guaranteeing fairness of radio and channel allocation to nodes and clusters;
- achieving the coexistence of LaSo with traditional 802.11 DCF channel access mechanisms.

The rest of this paper is organized as follows. Section 2 discusses the related work in MRMC allocation for WMNs. Section 3 introduces the Latin square concept and its application for channel access scheduling in different time-scales, and the social centrality concept with different metrics. Section 4 presents the protocol details of LaSo, including clique-based clustering schemes, on-demand coloring algorithms, fair MRMC allocation and scheduling, coexistence mechanisms with 802.11 DCF. Section 5 evaluates LaSo using simulations in terms of clustering performance, network throughput, end-to-end delay and jitter in contrast to other channel access protocols. Section 6 discusses a number of related issues. Section 7 concludes our paper.

2. Related work

Multi-channel allocation and scheduling have been extensively studied. An excellent survey of these problems and solutions in multi-channel wireless network settings was provided in [41]. WMNs with single-radio multi-channel (SRMC) capabilities were
addressed in [28, 21, 35, 37, 29]. In this section, we survey the most related research in WMNs with multi-radio multi-channel (MRMC) capabilities.

Most MRMC protocols introduce a common default channel to handle network partitioning by dynamic radio and channel assignment, and to facilitate radio and channel negotiation for data communications [23, 36, 42]. To assign channels to the interfaces other than the default radio, [23] presented a localized greedy heuristic based on an interference cost function defined for pairs of channels. [36] considered the mesh networks with main traffic flowing to and from a gateway, which is also in charge of the channel computation. A multi-channel MAC protocol in [42] requires all nodes to meet at the common channel periodically to negotiate the channels for data communication. In [24], each node fixes one radio on some channel but different nodes possibly use different fixed channels. Therefore the remaining radios of the node could dynamically switch to its neighbors fixed channels for data transmission. In [5], radios switch among channels following some pseudo-random sequences such that neighboring nodes meet periodically at a common channel. Beside the multi-channel allocation based on default channel, a new superimposed code based method was introduced by [48], to assign channels to all radios in a static fashion for a mesh network with a more general peer-to-peer traffic pattern.

Game theory also has been applied in MRMC mesh networks to solve the multi-channel allocation problem for different purposes [16, 13, 47]. [16] studied the competitive multi-radio channel allocation problem in multi-hop wireless network, and proposed the min–max coalition-proof Nash Equilibrium channel allocation scheme in the game, which aims to max the achieved data rates of communication links. [13] studied the channel allocation in non-cooperative MRMC wireless networks with multiple collision domain. A charging scheme has been proposed to induce players to converge to a Nash Equilibrium (NE). [47] proposed a globally optimal channel assignment for non-cooperative wireless networks, using strongly dominant strategy equilibrium to guarantee that the system must converge to a state in which the effective system-wide throughput is optimized.

The topology control approaches to multi-channel assignment in multi-radio wireless mesh networks have been studied in [31, 43, 30]. [31] proposed a cluster-based multi-path topology control and channel assignment scheme, which explicitly creates a separation between the channel assignment and topology control functions, thus minimizing flow disruptions. [43] gave a definition of co-channel interference, and used it to define and present an heuristic for the minimum interference survivable topology control problem which seeks a channel assignment for the given network such that the induced network topology is interference-minimum among all K-connected topologies. [30] presented a formulation of the base channel assignment as a topology control problem, and showed that the resulting optimization problem is NP-complete. They then developed a greedy heuristic channel assignment algorithm for finding connected, low interference topologies by utilizing multiple channels.

Other issues in MRMC mesh networks include the joint channel assignment and routing problem in WMNs [38, 17, 46, 3]. [38] proposed the centralized channel assignment and routing algorithms. [17, 46] studied the distributed channel assignment and routing problems. The joint channel assignment and routing for throughput optimization in MRMC mesh networks has been addressed in [3]. The capacity of MRMC in wireless networks has been studied in [25, 22, 7]. [25] addressed the impact of number of channels and radios on the capacity. [7] studied the connectivity and capacity of MRMC networks with channel switching constraints.

Compared to the works mentioned above and our previous work using Latin square to solve grid-based access scheduling in single-radio multi-channel WMNs [45], our LaSo adopts a different network structure based on interference clusters and bridge clusters using social centrality for multi-radio multi-channel scheduling in WMNs. To the best of our knowledge, LaSo is the first work to use social centrality to reflect the social behavior pattern of human’s access in WMNs, and then obtain good clustering results for channel allocation and access purposes. Nodes/clusters that are in two-hop range are assigned with different Latin squares for multiple access scheduling, therefore LaSo can avoid the co-channel interference caused by hidden terminal both during intra-cluster communication and inter-cluster multi-hop communication. Meanwhile, the compact Latin squares can be applied in a distributed way to utilize the on-demand graph coloring according to the updated traffic nodes in two-hop range, so it can guarantee fair and efficient multi-radio multi-channel resource allocation and access scheduling in multi-hop wireless mesh networks. The coexistence issues with IEEE 802.11 DCF have also been addressed in LaSo.

3. Fundamentals

3.1. 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. For examples, the following matrices $A$ and $B$ are two Latin squares of order 4.

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

Beyond even a very small order, such as 10, the number of different Latin squares 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 $H_k = [h_{i_1, i_2, \ldots, i_k}]$ in which each row is a permutation of symbols $1, 2, \ldots, n$ [27]. Latin cube is a three dimensional Latin hypercube. Although we apply Latin hypercube later, our discussions are mostly based on Latin squares.

There are many algorithms for generating Latin squares [27]. Compared with other complex and centralized channel scheduling schemes using a family of mutually orthogonal Latin squares (MOLS) in [9], the one that we adopt here is based on the modulo multiplication group theory because of its simplicity of computations. The Latin square generation algorithm is specified as follows:

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 the transpose of vector $A$, denoted as $A^T$, and vector $B$ using the modulo multiplication operation:

$$L_{n \times n} = A^T \cdot B \mod (n + 1), \quad i, j \in \{1, 2, \ldots, n\}. \quad (1)$$

**Lemma 1.** The square matrix $L_{n \times n}$ generated in Eq. (1) is a Latin square.

**Proof.** In order to prove that the square matrix $L_{n \times n}$ is a Latin square, we just have to show that the symbols in each column are distinct. The distinctness of the symbols in each row follows the same proof.
Proposition 1. Two nodes in a common Latin square will be collision free with each other all the time in LaSo.

Proof. The theorem follows directly from the definition of a Latin square and its usage in LaSo. Because every symbol only appears once in each row and once in each column, by mapping the channel-access nodes and time slots to the rows and columns of a common Latin square, the nodes' symbols will be distinct in each time slot, i.e., they do not have any chance to collide with each other.

That is, we need to show that the following two elements are distinct,
\[ l_{i_1,j_1} \neq l_{i_2,j_2}, \quad \forall i_1, i_2, j \in \{1, 2, \ldots, n\}, \quad \text{and} \quad i_1 \neq i_2. \]
These two elements are actually calculated as follows,
\[ l_{i_1,j_1} = a_{i_1} \cdot b_{j_1} \mod (n + 1), \]
\[ l_{i_2,j_2} = a_{i_2} \cdot b_{j_2} \mod (n + 1). \]
Because \( n + 1 \) is a prime number and \( |a_{i_1} - a_{i_2}| < n + 1, b_{j} < n + 1, \) therefore
\[ l_{i_1,j_1} - l_{i_2,j_2} = (a_{i_1} - a_{i_2}) \cdot b_{j_1} \mod (n + 1) \neq 0. \]
That is, \( l_{i_1,j_1} \neq l_{i_2,j_2}. \]

According to Eq. (1), each permutation of vector \( A \) or \( B \) creates a different Latin square. Therefore, by simply permuting vector \( A \) or \( B \) or both, we can generate a sequence of different Latin squares of order \( n \). This is why we choose to use the modulo multiplication group theory for Latin square generation purposes. The method can be easily applied in the distributed environment by the on-demand coloring scheme we take to get the number of active neighbors \( m \) in two-hop range first in Section 4.2, and then choose a prime number \( n + 1 \), which is greater than \( m \) to construct the Latin square.

In LaSo, time is structured into periodic time frames, each consisting of \( n \) time slots, from which we can construct a Latin square \( L_{n \times n} \). A medium access control protocol in LaSo, which respectively maps the channel-access contenders and time slots to the rows and columns of the Latin square, allows conflict-free channel access scheduling using the CSMA/CA based mechanisms in wireless networks. Accordingly, instead of the random exponential backoff mechanism in IEEE 802.11 DCF, we use Latin square symbols of the current time slot for backoff purposes, therefore providing conflict-free backoff intervals to the contending wireless nodes and avoids channel access collisions.

Fig. 1 illustrates the way of using a \( 4 \times 4 \) Latin square \( L_{4 \times 4} \) to schedule channel access among four nodes A, B, C and D. Fig. 1 (a) presents a WLAN system consisting of a single IEEE 802.11 BSS (Basic Service Sets) with one AP A and three MSs B, C and D. Fig. 1 (b) illustrates the mapping of nodes A, B, C and D and time slots \( t_1, t_2, t_3 \) and \( t_4 \) to the rows and columns of a \( 4 \times 4 \) Latin square. Provided with the packet arrival events on each node indicated with solid square boxes, Fig. 1 (c) presents the channel access orders of the nodes, determined by the Latin square symbol assignments of the nodes in the corresponding time slots. For instance, node A backs off for 1, 2 and 3 intervals in time slots \( t_1, t_2 \) and \( t_3 \). Note that a backoff interval lasts 20 \( \mu \)s in IEEE 802.11 DCF using DSSS, and a LaSo time slot may last a few milliseconds in this example.

3.2. Social centrality features

The deployment of WMNs is not arbitrary in real life. It is usually designed to satisfy the distribution of access requests. At the spots with high density of human’s access to WMNs, more mesh nodes will be deployed with corresponding density to solve the congestion control and load balancing issues. Some of these nodes located as the hottest access point may have a very high number of neighbors. These deployment patterns reflect the social behavior patterns of human’s data access. Therefore, social metrics, like social centrality, can measure and obtain good clustering results for channel access purposes.

The centrality of a node in a network is a measure of the structural importance of the node. A central node, typically, has a stronger capability of connecting other network members. There are several ways to measure centrality. The three most widely used centrality measures are Freeman’s degree, closeness, and betweenness measures [15].

Degree centrality is measured by the number of a given node’s direct links. A node with high degree centrality is popular, with many links to others. So, a central node occupies an important structural position for information exchange. In contrast, peripheral nodes with few or no relations are located at the margins of the network. Degree centrality for a given node \( p_i \) is calculated as:
\[
C_D(p_i) = \sum_{k=1}^{N} a(p_i, p_k)
\]
where \( a(p_i, p_k) = 1 \) if a direct link exists between \( p_i \) and \( p_k \), and 0 otherwise, \( i \neq k \).

Closeness centrality measures the reciprocal of the mean distance \( d(p_i, p_k) \), which is the shortest path between a node \( p_i \) and all other reachable nodes. Closeness centrality can be regarded as a measure of how long it will take information to spread from a given node to other nodes in the network. Closeness centrality for a given node is calculated as:
\[
C_C(p_i) = \frac{N - 1}{\sum_{k=1}^{N} d(p_i, p_k)}
\]
where \( N \) is the number of nodes in the network and \( i \neq k \).

Betweenness centrality measures the extent to which a node links other nodes. Betweenness centrality can be regarded as a measure of the extent to which a node has control over information flowing between others. A node with a high betweenness centrality has a capacity to facilitate interactions between linked nodes. In wireless networks, it can be regarded as how well a node can facilitate communication to other nodes in the network.
threecategories, namely the transmission range \( R \), duplex mode, i.e. only receive or transmit at a time, but not within negligible interval, and that each radio works in half-
WMN节点是 capable of switching between multiple channels using omnidirectional antennas. In addition, we assume that each multi-radio multi-channel (MRMC) communication capabilities

3.3. Network assumptions

We consider the network as a peer-to-peer WMN [2], in which every WMN node is both an Access Point and a mesh router with multi-radio multi-channel (MRMC) communication capabilities using omnidirectional antennas. In addition, we assume that each WMN node is capable of switching between multiple channels within negligible interval, and that each radio works in half-duplex mode, i.e. only receive or transmit at a time, but not both.

We model the effective ranges of radio communications in three categories, namely the transmission range \( R_{tx} \), the carrier sensing range \( R_{cs} \) and the interference range \( R_{i} \). The transmission range \( R_{tx} \) is mainly determined by transmission power and radio propagation, i.e. attenuation properties, and represents the range within which a packet is successfully received if there is no interference from other radios. The carrier sensing range \( R_{cs} \) is the range that a receiver detects the carrier as busy if there is an ongoing transmission within \( R_{cs} \). The interference range \( R_{i} \) is the range outside the carrier sensing range, and the received signal from other transmitter is treated as background noise.

The carrier sensing range \( R_{cs} \) is an adjustable parameter in the physical layer modules, and is usually set to 2 to 2.78 times of \( R_{tx} \) [49]. Therefore, the two-hop distance in WMNs is a good approximation of the carrier sensing range, and node access scheduling usually requires all neighbors of a node within two-hop range to be silent when the node transmits.

The network topology of a WMN is represented as an undirected graph \( G = (V, E) \), in which \( V \) is the set of WMN nodes, each assigned a unique ID number, and \( E \subseteq V \times V \) is the set of links between two nodes in \( V \) if they are within the transmission range \( R_{tx} \) of each other. The end-points of a link in \( E \) are called the one-hop neighbors of each other, and two nodes without a link between them but sharing a common one-hop neighbor are called two-hop neighbors of each other.

Betweenness centrality is calculated as:

\[
B(p_i) = \sum_{j=1}^{N} \sum_{k=1}^{j-1} \frac{g_{jk}(p_i)}{g_k}
\]

(4)

where \( g_k \) is the total number of paths connecting \( p_i \) and \( p_k \), and \( g_{jk}(p_i) \) is the number of those paths that include \( p_i \).

Freeman’s centrality metrics are based on analysis of a complete and bounded sociocentric network. Fig. 2 gives an example of a social connections diagram visualized by the social network analysis software UCINET [8], where nodes marked in red dots have high betweenness centrality and act as bridges between clusters in the network. In a social network, high betweenness individuals are often found at the intersections of more densely connected network communities.

4. LaSo specifications

LaSo is a Latin square Scheduling in multi-radio multi-channel (MRMC) wireless mesh networking using social centrality, which is mainly composed of following schemes: interference and bridge clustering, on-demand coloring based multiple access scheduling, Latin square based MRMC assignment, and coexistence with 802.11 DCF.

4.1. Clustering

It is well-known that CSMA/CA and RTS/CTS mechanisms cannot fully guarantee collision-freedom due to hidden and exposed terminal problems in multi-hop wireless networks. Instead, a hierarchically organized, cluster-based spatial division multiple access scheme has to be applied so as to resolve the hidden and exposed terminal problems at the macro-time scale, while the CSMA scheme is limited to the micro-time scale channel access control amongst nodes within individual clusters.

For clustering purposes, we use the interference graph and collision domain concepts to organize the network. Because the interference graph and collision domains of a WMN are not easily obtained, instead we derive the interference graph from the network topology of the WMN. That is, the interference relation is defined if two nodes are within two-hop range from each other in the WMN topology graph. So the interference graph is derived from \( G \), and defined as \( G^2 = (V, E^2) \), in which \( E^2 \) represents the one- or two-hop relations between nodes in \( V \).

For collision avoidance and communication purposes, we structure the WMN according to types of cluster, namely interference cluster (IC) and bridge cluster (BC). Nodes in the same interference cluster form a collision domain, and share a common channel. Nodes in the same bridge cluster form a collision domain as well, except that they are chosen among adjacent interference clusters so as to facilitate communications between interference clusters that use different channels.

In order to describe the clustering algorithms, we introduce the notations in Table 1.

4.1.1. Interference cluster

The interference graph \( G^2 \) is partitioned into a number of cliques for collision-free MRMC scheduling. In our paper, every clique is a

Fig. 2. Betweenness centrality in the social network.
fully connected two-hop subgraph of $G^2$, where each pair of nodes is either one hop or two hops away. We construct the interference clusters by finding the clique cover in $G$. However, it is well-known that the optimal clique cover problem is one of Richard Karp’s original 21 NP-complete problems [20]. Therefore, we simply provide a straightforward heuristic to form the interference clusters based on degree centrality, which is the number of one-hop neighborhoods. Here we start from a node that has the maximum degree$^1$ in the network as the seed to form a cluster from its one-hop neighborhood in $G$.

Algorithm 1 specifies the procedures to create $IC$s for channel access purposes using the original graph $G$ and the interference graph $G^2$. Line 1–8 select an unclustered seed node with maximum one-hop degree centrality in $G$ to construct a new $IC$. Line 3 establishes a flag variable $Stalemate$, which terminates the program if no nodes remain unclustered. Lines 5–7 merge all the unclustered one-hop neighbors of the seed node to the new $IC$. Now the new $IC$ contains the seed node and its all one-hop neighbors. We next look at the unclustered two-hop neighbors of the seed node. In Line 9–22, for each two-hop neighbor of the seed node, if it has one-hop or two-hop links in $E^2$ with all the one-hop neighbors of the seed node, we add this two-hop neighbor into the new $IC$. The order on the selection of the two-hop neighbor is based on the degree of its one hop links to all the one hop neighbors of the seed node. The condition $Number$ $of$ $clustered$ $nodes$ $< Upper$ $bound$ $number$ in line 9 limits the size of each $IC$ so as to reduce the channel resource contentions in each cluster. We keep iterating the above procedures until every node in $G^2$ has been assigned to an $IC$.

Fig. 3 provides an example to illustrate the steps of creating $IC$s for intra-cluster medium access. Given an original graph $G$ in Fig. 3 (a), Fig. 3 (b) merges the one-hop neighbors of selected seed node $c$ to create the initial $IC$, and then Fig. 3 (c) merges the two-hop neighbors of seed node $c$ from the interference graph $G^2$. Fig. 3 (d) is the final result of iterating above steps, and we get the clustered interference graph $IG$. For a better visual effect, we just keep one-hop links in the following figures of clustered interference graphs.

4.1.2. Bridge cluster

In LaSo, the interference clusters are individually allocated to channels, and allowed to access the channels only during certain time periods. For communication between different interference clusters, we derive bridge clusters for interconnections between interference clusters.

$^1$ We use maximum degree here to describe our clustering algorithms. The evaluation of different degree criteria will be discussed in Section 5.1.

### Table 1: Notation and meaning.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$IG = {U, E}$</td>
<td>Clustered interference graph, in which $U$ is the set of clustered WMN nodes, each forming a clique composed of nodes with one hop or two-hop links to each other, and $E$ is the set of edges, each indicating that there exist two WMN nodes, each of which on one end-point and could be connected in one hop range.</td>
</tr>
<tr>
<td>$SG = {I, J}$</td>
<td>Subgraph of clustered interference graph $IG$ by removing all the one- or two-hop links within each cluster, and the nodes without one hop links to other clusters. $I$ is the set of WMN nodes with one hop links to other clusters, and $J$ is the set of edges, which indicate the two end-points located in two different clusters within one hop range.</td>
</tr>
<tr>
<td>$IC_u$</td>
<td>Interference cluster of seed node $u$. Seed node $u$ is selected by its degree centrality.</td>
</tr>
<tr>
<td>$BC_u$</td>
<td>Bridge cluster of seed node $u$. Seed node $u$ is selected by its betweenness centrality.</td>
</tr>
<tr>
<td>$N_u^1$</td>
<td>The set of one-hop neighbors of node $u$.</td>
</tr>
<tr>
<td>$N_u^2$</td>
<td>The set of two hops neighbors of node $u$.</td>
</tr>
</tbody>
</table>

### Algorithm 1: Interference clustering

**Input:** Graphs $G = \{V, E\}$ and $G^2 = \{V, E^2\}$.  
**Output:** Clustered interference graph $IG = \{U, E\}$.

1. if node $a$ has the maximum degree centrality in unclustered nodes then
   2. select node $a$ as the seed node;
   3. Stalemate $\leftarrow$ false;
   4. /* Merge node $b$ to node $a$’s interference cluster. */
   5. foreach $b \in N_a$ do
     6. $IC_a \leftarrow IC_a + \{b\}$; // Merge
   7. end
   8. end
9. while Stalemate $\equiv$ false and Number of clustered nodes $< Upper$ bound number do
   10. Stalemate $\leftarrow$ true;
   11. /* Find all candidate nodes for merging. */
   12. $M \leftarrow \{c|c \in N_a^2\}$;
   13. foreach $c \in M$ do
     14. foreach $b \in N_a$ do
       15. if $c \in N_b$ or $c \in N_b^2$ then
         16. $IC_a \leftarrow IC_a + \{c\}$; // Merge
         17. Stalemate $\leftarrow$ false;
         18. continue $c$; // Examine the next $c$.
       19. end
     20. end
     21. end
   22. end

### Algorithm 2: Bridge clustering

**Input:** Graphs $G = \{V, E\}$, $G^2 = \{V, E^2\}$ and $IG = \{U, E\}$.  
**Output:** Bridge cluster $BC$.

1. /* Derive the subgraph $SG$ of the clustered interference graph $IG$ for bridge clustering operations. */
2. foreach $IC_u \in IG$ do
   3. $SG \leftarrow$ remove edges and nodes without inter-cluster links from $IG$; // Form $SG$
4. end
5. if node $a$ has the maximum betweenness centrality in unclustered nodes then
   6. select node $a$ as the seed node;
   7. Stalemate $\leftarrow$ false;
   8. /* Merge node $b$ to node $a$’s bridge cluster. */
   9. foreach $b \in N_a$ do
     10. $BC_a \leftarrow BC_a + \{b\}$; // Merge
   11. end
   12. end
13. Add all links in $G^2$ of node $a$ to $SG$;
14. Run 9–20 lines of Algorithm 1 on $SG$ to find a dominating bridge cluster $BC$;

The procedure to form $BC$s is similar to that of deriving $IC$s. Algorithm 2 specifies the steps to generate $BC$s. We first remove all the one- or two-hop links within each cluster, and the nodes without one-hop links to other clusters from $IG$. After removed those edges and nodes, we have a subgraph $SG$ which only contains edges and nodes supporting inter-cluster links. We use betweenness centrality, which is the number of one-hop clusters of

---

1. We use maximum degree here to describe our clustering algorithms. The evaluation of different degree criteria will be discussed in Section 5.1.
a node in SG, to form BCs. Here, we form the BCs by choosing the unclustered node with maximum betweenness centrality\(^2\) as the seed node. After selecting the seed node of a BC, we add links in \(G^2\) to SG in line 13 only if both end nodes of the links are in SG. The purpose of adding the links back is to find a wider coverage and reduce the number of BCs in the network. After this, line 14 in Algorithm 2 runs the lines 9 to 22 of Algorithm 1 to further construct BCs.

Fig. 4 provides an example to illustrate the steps of bridge clustering for inter-cluster medium access. Fig. 4(a) is the clustered interference graph \(IG\) generated from interference graph \(G\) by Algorithm 1. Fig. 4(b) removes the intra-cluster links and keeps the inter-cluster links from \(IG\) to create the subgraph SG. Fig. 4(c) merges one-hop neighbor \(m\) of seed node \(f\) chosen on SG. Fig. 4(d) adds other links from \(G^2\) to SG, and then generates the BC.

After creating the interference clusters and bridge clusters, we can get the complete clustered network graph for MRMC communication in WMNs. Each cluster in the clustered network graph is assigned a number for identification purposes in further calculations in LaSo. In fact, each cluster defines a collision domain for channel access scheduling purposes.

4.2. On demand multiple access scheduling

In the previous section, we have constructed the interference clusters and bridge clusters for the WMNs. However, the channel access schedules are not discussed yet. To provide a collision-free environment in the network, we allocate different channels for neighboring clusters, and only one node in each cluster is allowed to transmit at any time. We utilize the conflict-free schemes of LaSo introduced in Section 3.1 for intra-cluster transmission by assigning Latin square indices to the nodes in each cluster. Because it is usually the case that only a subset of the network nodes require channel access for data forwarding purposes, instead of assigning a Latin square row to each and every node in the wireless networks, we map the Latin square row indices to different colors, and in turn assign colors to nodes with traffic forwarding demands using graph coloring schemes.

There are two types of graph coloring schemes in our MRMC mesh networks for channel access scheduling, which are node coloring and cluster coloring. Node coloring schedules channel access time for the nodes in the same collision domain. Cluster coloring is used to allocate channels for different clusters.

4.2.1. Node coloring

From our previous discussions, we have specifically limited the size of the cluster such that the nodes within each cluster can overhear each other. In this section, we describe our graph coloring scheme to achieve conflict-free color assignments to nodes within two-hop range of each node.

In our coloring scheme, each node maintains a coloring table, consisting of tuples with the following information:

- Node ID, denoted by nodeID, which is usually the MAC address of the node.
- Color assigned to node \(i\), denoted by myColor\(_i\). This field contains all 0’s except for a single bit if the node is assigned with a color. Otherwise, its myColor\(_i\)=0.
- The color bitmap of node \(i\)’s one-hop neighbors, denoted by oneHopColor\(_i\).
- Expiration time. Each coloring tuple is refreshed when the tuple is updated by the corresponding node.

The basic mechanism of the conflict-free color assignment is to exchange one-hop neighbor color assignments between adjacent nodes by piggybacking the color tuple (nodeID, myColor, one hop, expiration time) information in every data frame sent by node \(i\).

Fig. 5 illustrates the coloring information included in the header of data frames in LaSo. The colors are represented using two
The transmitter derives myColor or 0, respectively. rep represents a color that has been taken or not by setting the bit to 1 or 0, respectively.

The color assignment scheme is designed as follows. First, the transmitter derives myColor and onehop, by the logic "OR" operation in its color table, and attaches the coloring information in the outgoing data frame. The receiver records the incoming coloring information to its color table, and checks if there is any color conflict. If the receiver finds that its color has been used in the color bitmap of the received unicast packet, the receiver will reassign itself with another unused color.

Once we have the color assignment to each node, the Latin square row assignments to different nodes is automatically achieved by mapping the color information to the Latin square row indices.

In multi-hop wireless networks, the color assignment occurs when a routing path is established between a source and its destination. That is, when a unicast RREP (route reply) message is sent from the destination to the source. While in the first phase of route discovery that uses RREQ (route request) broadcast messages, color assignments are learned but not assigned to the nodes along the source to destination path.

Fig. 6 gives an example on the multi-hop coloring and how node 3’s coloring table evolves in this RREQ/RREP/DATA transmission process by AODV routing [32]. Since LaSo is independent of specific routing design, the coloring mechanism also works for other routing protocols. During the initial RREQ process, every node records its neighbors’ ID and initial color information by the broadcast message but does not assign color. When RREQ from source node 1 arrives at destination node 7 and node 7 finds a backward routing to node 1, nodes begin to implement coloring process in the RREP process. Every node in the RREP route obtains the color information \( C_i \) from the header of packet frames, records the information from its previous one-hop neighbor in the backward routing and computes its own color. When RREP packet arrives at source node 1, node 1 has the routing to destination node 7 and then begins to transmit data to destination through the forwarding nodes in the routing. From above coloring process, once data communication begins, every node will record all the color information of its neighbors in two-hop range. Thus, whenever routing is reestablished or new data flow is added to the networks, every node can update its color table and recompute its color if color collision happens with its neighbors in two-hop range.

4.2.2. Cluster coloring

Using the node coloring scheme, we can guarantee a collision-free environment within interference or bridge clusters. However, interference could still happen when two close nodes that are located in two different clusters with the same channel are scheduled to transmit at the same time. Cluster coloring is provided to allocate and schedule channels among close clusters. We will address the allocation and scheduling problems between colors and channels in clusters in Section 4.3.

To achieve conflict-free color assignments to clusters within two-hop range of each other, we map the Latin square rows to different colors, and in turn assign colors to the clusters. Heuristically, by replacing node entities by cluster entities in the above node coloring schemes, we can ensure that no clusters within two-hop range obtain the same color when a routing path is established.

**Proposition 2.** Given an interference graph \( G = (V, E) \), the clustering and coloring schemes in LaSo can achieve collision-free communication in multi-hop wireless mesh networks.

**Proof.** From Proposition 1, we know that through mapping the channel-access nodes/clusters and time slots/time frames to the rows and columns of a common Latin square, they will not have any chance to collide with each other. Clustering in LaSo forms multi-hop WMNs into multiple interference clusters and bridge clusters based on clique information. Proper social centrality and upper bound of clustered nodes are applied in the LaSo to balance cluster size and channel resource. Then coloring scheme in LaSo can assign distinct symbols to channel-access contenders (nodes or clusters), so that intra-cluster nodes can achieve collision-free communication using the symbol-based backoff scheme in Section 3.1 and inter-cluster nodes can achieve collision-free communication using the symbol-based MRMC assignment scheme in Section 4.3. \( \square \)

4.3. MRMC allocation and scheduling

So far, we have colored the network such that clusters within two-hop-cluster range will obtain different colors. Using the color information, we can now set up a collision-free multiple access schedule for different radios and channels in the network. In this section, we will first explain the schedules between clusters and channels. Because each neighboring cluster uses a different channel to avoid collisions, we then talk about how to utilize multiple radio interfaces to connect neighboring clusters.

4.3.1. Scheduling between clusters and channels

Once the color assignment of the clustered network is finished, LaSo distributes the color information to the clusters, including the color of every cluster and the total number of colors required in two-hop range. Using those color information and the number of available channels in the large-scale WMN system, each cluster can independently generate its channel access schedules in distributed environment. We address it in following cases:

- The number of colors is equal to the number of available channels. In this case, the clusters are one-to-one mapped to the channels, and each cluster 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 the number of available channels, which enables each cluster to be allocated with multiple channels. Since active nodes in the collision domain based cluster have different colors, multiple channels can be scheduled by mapping nodal colors to different Latin square values. Then nodes can achieve collision-free transmission using different channels in the same time slot.
- The number of colors is more than the number of available channels. In such situations, we have to apply Latin square schedules to share the channels in the same time frame according to the coloring information of clusters. Nodes in the same cluster transmit using the same channel in a time frame and only one node in the cluster is allowed to transmit in each time slot, so that there will be no transmission collision.

Note that each time frame is composed of multiple time slots. We have two methods to assign channels, namely the one based on a Latin square and the other based on a Latin cube.
methods can support both the second case with micro-time (timeslot scale) scheduling, and the third case with macro-time (time-frame scale) scheduling. These methods can also be extended to support the first case.

In LaSo, multi-channel are firstly one-to-one assigned to different clusters to guarantee network connectivity for multi-hop transmission, and then if there are extra channels, cluster can have multiple channels.

Here we use an example partially from the clustered WMN as shown in Fig. 7 to explain our Latin squares based channel assignment scheme for clusters. Each interference cluster in Fig. 7(a) is circled by the black dashed lines. Two bridge clusters connecting the interference clusters are circled by the red dashed lines. Suppose that the clustered network is colored by 5 colors, and only 2 channels are available for channel allocation.

Fig. 7(b) shows the Latin squares based approach, which uses a Latin square of order 5. In order to assign the 2 channels to the clusters with different colors, we map the colors to the rows of the Latin square, and time frame numbers to the columns. In each time frame, we require that the clusters with Latin square values 1 and 2 are able to utilize channels 1 and 2, respectively, for communication purposes. For instance, clusters in colors 1 and 2 will access the channels 1 and 2, respectively, in time frame $T_1$.

Fig. 7(c) shows the Latin cube based approach, which uses a Latin cube of order 5, and maps channels, cluster 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 clusters with colors that maps to Latin square value 1. For instance, clusters in color 1 and 5 will access the channels 1 and 2, respectively, in time frame $T_1$.

Once a cluster in the clustered network graph is assigned a channel, the WMN nodes inside the cluster can communicate using the medium access control protocol, specified in Section 3. WMN nodes without any channel allocation have to remain silent during the corresponding time slots.

**Fig. 6. Coloring mechanism.**

**Fig. 7. MRMC scheduling in a clustered network graph.**

**4.3.2. Scheduling between clusters and radios**

As shown in Fig. 7, nodes in the bridge cluster can be connected with multiple neighbor interference clusters. If we have multiple radios on each node, inter-cluster and intra-cluster communications may be activated at the same time, and thus increase the network throughput and reduce the end-to-end delay.

According to the number of interference clusters which overlapped with the bridge cluster, we can assign different radio interfaces to support communication between clusters. Using the coloring information of each interference cluster, the maximum number of colors, and the number of available radios, each node in the bridge cluster can independently generate its radio access schedules. Again there are three cases that need to be considered:

- The number of colors is equal to the number of available radios. In this case, the colors are one-to-one mapped to the radios, and we can activate all inter-cluster transmissions concurrently. We do not elaborate on this scenario further in this section.
- The number of colors may be far less than that of available radios. In this case, some of the radios may not be assigned to any channels. By removing the extra radios, we can reduce this scenario to the case that the number of colors and radios are equal.
- The number of colors is more than the number of available radios. In such situations, We can use the previously mentioned Latin square/cube based methods to assign radios, just simply replacing the channel output by the radio output.

**4.4. Coexistence with 802.11 DCF**

If all network nodes are allocated with Latin square rows, and access the channel according to LaSo, then we completely eliminate packet collisions in LaSo. However, there are two reasons that we need the random channel access scheme based on IEEE 802.11 DCF, and enable a coexistence mechanism for DCF and LaSo:
For the combined service of heterogeneous wireless networks, mobile clients have to depend on a more flexible DCF channel access mechanism to contact the servers before Latin square row assignments are made.

In LaSo, the size of a Latin square is a per-cluster parameter derived from the routing discovery process as addressed in Section 4.2. It determines the number of WMN nodes that can be supported under current traffic using the LaSo channel access paradigm. Once the Latin square rows are distributed and assigned to the WMN nodes, the LaSo does not allow new nodes to access the channel but to hold till the new nodes initiate a new routing discovery request to reassign the size of Latin square for updated traffic nodes.

Therefore, a fallback mechanism is necessary in LaSo so that nodes without any Latin square assignment may still access the channel using normal IEEE 802.11 DCF mechanisms. For clarity, we refer to nodes that access the shared channel using LaSo as LaSo nodes, and those using DCF as DCF nodes.

We utilize the frame fragmentation and NAV (network allocation vector) mechanisms in IEEE 802.11 DCF to achieve the coexistence of LaSo and DCF channel access mechanisms.

Fig. 8 shows the original specification of IEEE 802.11 for transmitting a multi-fragment frame burst using the prioritized SIFS based channel access and NAV channel reservation, in which each prior fragment burst pair (fragment + ACK) sets the channel reservation for the following fragment burst.

In IEEE 802.11 DCF, NAV is a way for communicating WMN nodes to reserve the channel for a certain amount of time in order to exchange data and control frames, using the Duration field of the MAC header. Except for the communicating pair of WMN nodes, all other WMN nodes that observe the value of the Duration field shall remain silent during the subsequent period that lasts for Duration microseconds.

During the transmission of LaSo nodes, we add an imaginary fragment burst after each normal DCF data frame exchange by setting the Duration fields of the DATA and ACK frames, which lasts for a number of node-slots, equal to the order of the Latin square. In the subsequent channel idle period reserved by the NAVs, all WMN nodes except for LaSo nodes hold back their backoff timers. Therefore, the LaSo nodes carry out their backoff timer count-down operations until they may access the channel.

However, if no LaSo node accesses the channel once the NAV timers of the DCF nodes expire, the DCF nodes may get a turn to access the channel by decreasing their backoff window.

In fact, the LaSo/DCF coexistence mechanism has arranged two channel access stages, the first of which is for LaSo based access, and the second for DCF access.

Fig. 9 illustrates the channel access activities for exchanging three data frames between three pairs of source and destination nodes, in which both DCF and LaSo channel access functions operate. The first frame exchange sequence is carried out between DCF nodes using the normal DCF handshakes, and the last two are between LaSo nodes. In the normal DCF operations, the NAVs of overhearing WMN nodes are determined by the Duration fields of the RTS and CTS frames, which end at the end of the acknowledge frame in an RTS/CTS/DATA/ACK based handshake, and the Duration fields of the DATA and ACK frames are set to 0. In LaSo operations, the normal DCF based frame exchange is carried out except that the Duration fields of the DATA and ACK frames are extended, which correspond to a order of Latin square in Fig. 9.

In both Figs. 8 and 9, the RTS/CTS frames are optional, and do not affect the LaSo/DCF coexistence mechanism. Therefore, the LaSo/DCF coexistence mechanism is also suitable for transmitting broadcast traffic in a mixture of LaSo and DCF mechanisms because the NAV in the data frame for the imaginary fragment transmission is set in both DATA and ACK frames.

Under the LaSo/DCF coexistence mechanism, once LaSo 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 LaSo. In addition, the modified RTS and CTS frames include an additional flag NAV field as shown in Fig. 10, which tells the LaSo nodes to deduct the extended duration from their NAVs if present. So the LaSo channel access mechanisms operate before the declared NAV expires, and LaSo nodes achieve prioritized channel access over the DCF nodes.

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

5. Evaluation

In order to verify our MRMC access scheduling scheme under clustered interference graph in WMNs, we evaluate LaSo in two aspects:

(1) With regard to the clustering algorithms, we use degree centrality and betweenness centrality to form IC and BC. We compare the total number of clusters, the number of ICs and BCs, number of nodes per cluster, and standard deviation (std) of number of nodes per cluster in different sized networks. Also, we consider three criteria for degree centrality and betweenness centrality in our simulation to choose a seed node, namely maximum degree, minimum degree, and pure random in a network graph.

(2) For Latin square based multiple access scheduling, we used on demand coloring to assign Latin square indices to nodes in micro-time scale and clusters in macro-time scale. We evaluated LaSo under different network topologies, including a fully-connected network and a multi-hop multi-flow network. Using the proposed clustering algorithms, the network was partitioned into several interference clusters and bridge clusters. The channel/radio allocation and scheduling methods have been applied to IC and BC...
for inter- and intra-cluster communications. We collected the performance of throughput, delay and loss rate to compare with other protocols.

5.1. Clustering performance

We use two sets of simulations to evaluate our clustering algorithm. In both set of simulations, we deploy 100 to 1000 nodes in a 1000 m × 1000 m square area with 100 node increments. The only difference between the two set of simulations is the node deployment method. The network nodes in the first set of simulations are randomly deployed in the area. However, in the second set of simulations, the network nodes are organized using grids in the same area. We decrease the distance between each node to preserve the same node density as in the first set of simulations.

5.1.1. Random deployment

According to the first set of simulations where nodes are randomly deployed in the network, Fig. 11(a) gives the average number of IC and BC formed after running our clustering algorithm. As we can see, by choosing the seed node using the minimum degree, we could get the least number of IC. However, if we choose the seed node using maximum degree, we will obtain the least number of BC.

By looking at Fig. 11(b), we know that using maximum degree to choose the seed node will result in the least number of total clusters. It means that we could use fewer channels to achieve collision-free MRMC communication.

Fig. 11(c) shows the average number of nodes in IC and BC. Note that the fewer clusters formed in network, the more nodes each cluster contains. Fig. 11(d) gives the standard deviation of node distribution in each cluster, where the minimum degree criterion achieves the least standard deviation compared to the other two criteria.

5.1.2. Grid deployment

In the second set of simulations, the nodes are deployed using grids. We use the same performance metrics to evaluate our clustering algorithm in such network topology. The results are illustrated in Fig. 12. We can observe similar results appear as random deployment on the number of clusters and number of nodes per cluster. From Fig. 12(b), we can see that the maximum degree criterion still results in the least clusters formed in the network.

5.2. LaSo protocol performance

We implemented LaSo using the QualNet 4.5 network simulator [34], and evaluated its performance in terms of overall network throughput, average packet delay, and jitter by comparison with existing IEEE 802.11 standards, and some state-of-the-art multi-channel protocols, under a fully-connected network and a multi-hop multi-flow network with MRMC communication capabilities.

(a) RTS frame. (a) CTS frame.

Fig. 10. The extended RTS/CTS frames with NAV flags.

(a) Number of IC and BC. (b) Total number of clusters. (c) Number of nodes per cluster.

Fig. 11. Performance comparisons of degree metrics in random deployments.

(d) Std of nodes per cluster.
5.2.1. Fully-connected network

Fig. 14 evaluates the coexistence feature of LaSo, and the performance of channel access scheduling using Latin squares under fully-connected network. We randomly deployed 50 WMN nodes in a 500 m \( \times \) 500 m area, and created 25 flows between 25 distinct pairs of nodes. The transmission range of each node was 250 m, the packet size was 512 bytes, and the link bandwidth was 2 Mbps. All nodes that were within each other’s two-hop range form an interference cluster for data transmissions. We randomly chose 10 nodes to work under 802.11 DCF mode and other nodes to work under LaSo mode. Total network load rate increased from 0.4 Mbps to 2 Mbps (16 Kbps to 80 Kbps per node load rate). The number of available channels for communication within each cluster was 1. In comparison with IEEE 802.11e EDCA [18] and IEEE 802.11b DCF [19] in terms of average network throughput, end-to-end delay and jitter, LaSo performed well under fully-connected interference cluster due to the function of Latin squares on channel access collision avoidance.

5.2.2. Multi-hop multi-flow network

Because the deployment of multi-hop WMNs in an urban scenario is usually related to the population density and access frequency, we refer to the layout of the University Center, a neighborhood center adjacent to the University of California, Irvine (UCI), as the simulation scenario of a multi-hop multi-flow network with MRMC communication, and apply the social centrality in clustering for Latin square scheduling as shown in Fig. 13. There are over 50 various eateries, businesses and offices in the University Center. We placed 50 WMN nodes with a distribution by the frequented density over a shopping area of size 1500 m \( \times \) 1500 m, and randomly created 25 flows in the multi-hop wireless network. As shown in the Fig. 13, some popular places, i.e. University Office Tower, Edwards Theater, 24 Hour Fitness, are configured with more WMN nodes, which matches the pattern of human’s social behavior on wireless access. The black dashed lines indicate the interference clusters and the red dashed lines indicate the bridge clusters.

We ran the clustering algorithms and generated 13 interference clusters and 10 bridge clusters to cover the whole area. Under such situation, we assigned 9 colors to all nodes and 3 colors to all clusters by our coloring method. Each node had 2 radio interfaces and the number of available channels was 4. The transmission range of each node was set to 250 m by tuning the radio reception threshold values. The packet size was set to 512 bytes, and the link bandwidth for each channel was 2 Mbps. Each simulation was carried out for a duration of 30 s, and the end-to-end delay, jitter and network throughput were collected while we gradually increased the total network load from 0.5 Mbps to 5 Mbps (0.02 Mbps to 0.2 Mbps per mobile node). We compared our LaSo protocol with IEEE 802.11e EDCA and IEEE 802.11b DCF to evaluate its performance.

Fig. 15 shows the performance comparison of LaSo with IEEE 802.11e EDCA, and IEEE 802.11b DCF in the multi-radio
multi-channel simulation. Note that IEEE 802.11e EDCA and IEEE 802.11b DCF are all not capable of switching channels, therefore they operated over a single channel in our simulation. As we can see, because of the hidden terminal problems, IEEE 802.11e EDCA and IEEE 802.11b DCF all performed inefficiently and with limited stability in multi-hop scenarios.

The throughput comparison is shown in Fig. 15(a). When we increased the total network traffic load to 5 Mbps, the throughput of LaSo was about 1.5 times that of the other protocols. It is because that there are two types of nodes operating in LaSo. Some nodes are inter-cluster nodes while others are inter-cluster bridge nodes. The nodes for intra-cluster communication operate on one shared channel, and the bridge nodes for inter-cluster communication operate on two channels simultaneously to connect difference clusters. These differentiated services of WMN nodes under clustering improve the throughput of LaSo under MRMC schemes. If we also provide MRMC ability for the nodes to communicate within clusters, LaSo will further improve the throughput. However, it is not in the scope of this paper.

For the average delay and jitter metrics in Fig. 15(b) and (c), LaSo was also lower and more stable than IEEE 802.11e EDCA and IEEE 802.11b DCF.

We also compared the performance of LaSo with that of CoMTac [31], GAALS [45], LCM [29], MMRC [42] and Hyacinth [38] under same settings. Note that in the multi-hop WMNs simulation, we first consider to assign multi-channel one-to-one to different clusters, so that multi-channel protocols can guarantee network connectivity for multi-hop transmission. After this, if there are extra channels, cluster can have multiple channels later.

From simulation results as shown in Fig. 16(a) on throughput and Fig. 16(b) on average delay, it is obvious that all the multi-channel protocols can achieve better performance than single channel protocols, since they adopt multi-channel capabilities to schedule for communication. LCM, MMRC and Hyacinth are heuristic multi-channel scheduling protocols that assign channel and schedule channel access from local aspect without considering fully coordination of multi-hop neighbors. GAALS adopts grid-based structure to allocate multi-channel and scheduling transmission, therefore the multi-hop collision is reduced. CoMTac adds multi-radio capabilities to multi-channel scheduling, and it uses the cluster-based multi-path topology control and channel assignment scheme, which produce better performance than other multi-channel protocols. Our LaSo also fully utilizes and allocates MRMC in a clustered multi-hop wireless mesh network. Due to better clustering scheme to forward the data based on social centrality, and the fact that Latin square based scheduling can achieve collision-free intra-cluster and inter-cluster communications, LaSo significantly reduces the transmission failure, and therefore presents better performance than other multi-channel protocols, especially when the load rate is high.

6. Discussion

6.1. Time synchronization

Time is structured hierarchically in LaSo with four different levels of granularity, with two time slot concepts mapping to the micro-time scale and two time frame concepts mapping to the macro-time scale. Each time frames contains multiple time slots. The shortest time unit is called node-slot that has the same duration as the time slot defined by IEEE 802.11 DCF. In DCF based on DSSS, the slot duration is 20 µs, which is used as the backoff count-down unit in the CSMA algorithm. The next level time unit is called clique-slot, which last for a few milliseconds in this paper for reassigning channel access priorities among nodes. The next level is called cluster-frame, which is the time boundary that we generate new Latin squares for the next round of channel
access coordinations. The highest level time unit is called systemframe, which is another time boundary that we generate new Latin squares for changing channel allocations.

Although the deployment of wireless mesh networks is fixed and stable at most of the time, nodes may enter and leave the network with different local times. In LaSo, the access backoff by node-slot is based on the value of nodal Latin square assigned by on-demand coloring, and each clique-slot contains enough number of node-slots for the transmission of multiple packets. In addition, the collision domain based clustering provides a dynamic structure for distributed Latin square assignment by two-hop range, therefore, LaSo requires a coarse time synchronization between nodes to keep collision-free channel access.

The IEEE 802.11 standards support the peer-to-peer mode Independent Basic Service Set (IBSS). Lai et al. proposed very efficient algorithm that synchronizes clock within 100 µs when the number of nodes is more than 300 [26]. Such accuracy is sufficient for LaSo to be applied in both micro-time and macro-time division schemes. The multi-hop adaptive timing synchronization function (MATSf) has been presented lately that performs an accurate and scalable network synchronization even within highly dynamic structures [50]. Since there is a common channel for WMNs nodes to exchange control information in LaSo, the coarse time synchronization can be achieved through the common channel by listening to transmission and reception of data packets, the so-called beacons, that are carrying time-stamps and are processed in order to gather asynchronous information.

Specifically, each node runs an oscillator that drifts according to physical variations. This time is a direct function of the oscillator vibrations and therefore named the physical time which is denoted by \( p_i(t) \). Furthermore, the virtual time \( v_i(t) \) describes the artificial model of synchronized nodes which takes asynchronism information of adjacent neighbors into account. Network synchronization bases on the beacons that are carrying time-stamps and are processed in order to gather asynchronism information. One beacon generates at least two events: when being transmitted by one node and received by a neighbor. The event when node \( i \) generates its \( n \)-th beacon is denoted by \( T_i(n) \) while the event when node \( i \) receives it's \( n \)-th beacon is denoted by \( T_e(i, n) \). For instance, when a beacon releases \( T_e(1, n) \) at node 1 causing \( T_r(2, n) \) at a neighbor node 2, the times of these events only differ in propagation delay.

There are also a number of proposed time synchronization protocols for multi-hop wireless networks [12,40,39], which could be used as better time synchronization solutions in LaSo.

6.2. Mobility and scalability

One critical issue in applying Latin squares in wireless mesh networks is to determine Latin square order and distribute Latin square values to nodes. To explain how Latin square is used in LaSo for mobility and scalability we specially consider the micro-time division scheme where the Latin square values are used for deferral and backoff intervals.

As described in Section 4.2, the size of a Latin square is a per-cluster parameter, which is derived from the routing discovery process by an on-demand coloring scheme under distributed environment. The graph coloring method can determine the number of active WMN nodes in each two-hop range to support current end-to-end communication.

If a new node which does not have a Latin square value assigned joins the network and has packet to transmit. It first chooses a random channel access delay, greater than the Latin square order by the coexistence method as described in Section 4.4. By this way, the transmission from the node least interferes other in-network nodes. The new node could initiate a new routing discovery process to reassign the Latin square values to all active traffic nodes. However, the routing discovery may induce redundant overheads and delay in multi-hop WMNs. In addition, the node also could listen to the channel and determine which Latin square values have been taken by other nodes only in two-hop range through the common channel. After a period of time, the node starts sending Latin square request packet to ask for a Latin square value assignment. Any node in the two-hop range hearing the request responds with a list of used Latin square values from local color table. After such information collection at the newly joining node, the node would have an idea of which value to choose for channel access. The size of Latin square in the two-hop range will be accordingly adjusted upon the reception of the request packet.

After making any choice about the Latin square value, the node reaffirms 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 in two-hop range, the node obtains the Latin square value. Otherwise, the node has to go through another round of requesting a Latin square value, or continue using the longer backoff interval than the Latin square order.

Four types of messages are necessary in this process:

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

In mobility scenarios, if a node hears other nodes have the same transmission Latin square value, it needs to go through the same process to acquire a new Latin square value. Through above distributed online operation, the Latin squares can be scaled to support mobility of nodes in MRMIC mesh networks.
6.3. Relevance of centrality

Since social centrality could reflect the social behavior pattern of human’s access in WMNs, and then obtain good clustering results for MRMC allocation and access purposes, the relevance of centrality in LaSo has been analyzed through the relation between social centrality based clustering and multi-hop transmission performance on end-to-end packet loss probability and average delay. Some interesting results from the analysis are also shared in the section.

The multi-hop WMNs could be modeled as a tandem system with $L$ concatenated queues where traffic coming out of each queue is fed into the next queue in the chain, because the operation of multi-hop WMNs in LaSo where traffic from the source node usually traverses several hops to reach the destination node. The sequence of nodes that the traffic traverses is obtained from the routing related coloring algorithm as described in Section 4.2. We evaluate the multi-channel scheduling in multi-hop WMNs using an approximate discrete time $M/M/1$ queueing model with bulk arrivals [44]. Each WMN node in the wireless network maintains one queue for multi-channel, which is convenient for analysis but without generality. Traffics from different channels traversing through the node are buffered for transmission in a FIFO manner. In general, packets stored in the buffer may come from different channels.

Because of the wireless channel impairments, packets transmitted over each wireless link may be in error. In realistic situations, the transmission rate on each link in each time slot is determined by corresponding channel condition. Most current wireless standards employ adaptive modulation and coding (AMC) in the physical layer which essentially results in multi-rate transmission on wireless links. With AMC, each modulation and coding scheme (MCS) is called one mode and it corresponds to one particular interval of signal to interference plus noise ratio (SINR). Specifically, the SINR at the receiver is partitioned into a finite number of intervals with threshold values $X_0(=0) < X_1 < X_2 < \cdots < X_k+1(=\infty)$. If $X$ is the SINR at the receiver, transmission mode $k$ with its corresponding rate is employed if $X_k \leq X < X_{k+1}$ ($k = 0, 1, 2, \ldots, K$). Then we call the channel is in state $k$.

Since Latin square based intra-cluster and inter-cluster multiple access can achieve collision-free communication on wireless links by Propositions 1 and 2, then in LaSo a packet may be lost due to buffer overflow at any buffers or due to other transmission errors on the wireless links. We apply AMC and choose the SINR switching thresholds for multi-rate transmission, and then give an analysis of the relation between social centrality based clustering and multi-hop transmission performance on end-to-end packet loss probability and average delay in the Appendix. From the analysis, we learn that the performance of multi-hop WMNs is linearly related with the number of queues in the system.

Since LaSo applies the social centrality and two-hop collision domain based clustering scheme, the number of queues $L$ in a traffic from the source node to the destination node is closely related to the number of clusters in the routing path. Therefore, the analysis in the Appendix tells us in LaSo less number of clusters can achieve better multi-hop end-to-end performance if these clusters can be scheduled to transmit at the same time. In LaSo, if we use maximal degree of social centrality, then we can generate minimum number of clusters as shown in Section 5.1. Because of the MRMC capabilities, transmissions can occur simultaneously on all wireless links by using different wireless channels for these clusters according to the Latin square/cube method described in Section 4.3.

Meanwhile, we know that if we use maximal degree of social centrality for clustering, though we can achieve lower multi-hop delay and require less channels for multi-hop scheduling, we also increase the size of cluster with more nodes, which will result in the increase of intra-cluster access deferral and backoff intervals corresponding to the Latin square size. Therefore, we set a condition $N_c < UB$, where the $UB$ can be tuned depending on the various deployment of WMN nodes. Furthermore, as the second case in Section 4.3.1, when the number of cluster is reduced and less channels are used to connect multi-hop clusters, LaSo can assign extra channels to multiple nodes with different Latin square values in dense cluster to achieve collision-free intra-cluster transmissions using multiple channels in the same time slot, therefore guaranteeing the performance on network throughput and average delay.

7. Conclusion

We have presented LaSo, a novel multiple access scheduling protocol based on Latin squares and social centrality, for multi-radio multi-channel allocation and access scheduling in wireless mesh networks. In LaSo, the network is organized into clusters according to two hops interference information, and guarantees the network connectivity using the interference clusters based on degree centrality and bridge clusters based on betweenness centrality. The radio and channel access efficiency and fairness were achieved by using compact Latin square based MRMC access and channel allocation schedules. Especially, we challenged and revised a few design aspects of IEEE 802.11 DCF protocol, such as the CSMA/CA and RTS/CTS handshaking mechanisms to avoid the hidden and exposed terminal problems in multi-hop WMNs. In our simulation based studies, LaSo demonstrates strong advantages over existing IEEE 802.11 standards and other state-of-the-art multi-channel protocols, in multi-hop multi-flow WMNs with MRMC capabilities, therefore stands as a promising candidate in future MRMC-based WMN deployments.

Acknowledgments

We would like to thank Dr. Shih-Hsien Yang and Dr. Sung Park for their help and discussions during the process of designing LaSo protocol. This work was supported by the Raytheon Company under Grant No. RC-42621 and the National Science Foundation under Grant No. 0725914.

Appendix. Analysis of queueing and clustering in LaSo

We assume that transmissions occur in fixed-size time slot (clique-slot) and we model the M/M/1 queueing model in discrete time with one time step equal to one time slot. We assume that traffic arrives at the source node buffer according to a bulk Bernoulli arrival process where $i$ packets arrive in one time slot with probability $a_i$ ($i = 0, 1, 2, \ldots, M$). The system state is observed at the beginning of each time slot and node is scheduled to transmit the packets using Latin square symbols of the current time slot for backoff purposes as described in Section 3.1. Packets arriving in time slot $t$ can only be transmitted during time slot $t+1$ at the earliest.

The queues of the multi-hop WMN are numbered using an increasing sequence of integers where the source node maintains queue one and queue $i$ has buffer size of $Q_i$ packets. The transmission rate on each link in each time slot is determined by the corresponding channel condition. We assume that each wireless link employs AMC with the same number of $K$ modes, and the number of packets transmitted in mode $k$ is $c_{ki}$. Let the probability that the channel is in state $k$ on link $i$ be $p_{ki}$, which can be calculated as [4].
Let the probability that packets transmitted in error on link \( i \) and channel state \( k \) be \( P_e^{(i)} \), which can be calculated as in [6]. The channel state on each link is assumed to be stationary in each time slot but changes independently in consecutive time slots. We define \( q^{(i)}(n,m) \) as the probability that \( m \) packets are successfully received given \( n \) packets were transmitted over link \( i \) on channel state \( k \). Assuming packet errors occur independently, we can calculate \( y^{(i)}_k(n,m) \) as follows:

\[
y^{(i)}_k(n,m) = \binom{n}{m} \left( P_e^{(i)} \right)^n (1 - P_e^{(i)})^m. \tag{A.1}
\]

We assume that \( i \) packets arrive at queue \( k \) with probability \( q^{(i)}_k \). Let the maximum bulk size (maximum number of packets arriving in one time slot) captured in \( q^{(i)}_k \) for \( k \geq 2 \) is \( N \) (i.e., equal to \( c_k \)) while the maximum bulk size captured in \( q^{(i)}_1 \) is \( M \) (i.e., \( a^{(i)}_1 = M \)). Since the behavior of queue \( i \) can impact queue \( i + 1 \) in a Markov chain but the reverse is not true, we could find the queue length dynamics for one queue at a time where its input is the output of the previous queue in the chain.

Suppose there are \( L \) separate queues for a traffic from the source node to the destination node in the multi-hop route. Let us consider a particular queue \( k \) of the chain and form the Markov chain \( X_k(t) = \{q_k(t), 0 \leq q_k(t) \leq Q_k\} \) where \( q_k(t) \) denotes the number of packets in queue \( k \) at time slot \( t \) with arrival process described by \( A^{(i)}_k \). The transition probabilities for this Markov chain can be found as follows. Let us consider a general transition probability \( P\{X_1 \rightarrow X_2\} \). Let \( s \) be the number of packets arriving at the queue and the transmission channel is in state \( l \) during the considered time slot. Then, we have \( x_2 = \min\{x_1 + s, Q_k\} - \min\{x_1, c_i\} \). Thus, the transition probability \( P\{X_1 \rightarrow X_2\} \) can be found as

\[
P\{X_1 \rightarrow X_2\} = \sum_{s} q^{(i)}_k p^{(i)}_s \tag{A.2}
\]

where all combinations of \( l \) and \( s \) such that \( x_2 = \min\{x_1 + s, Q_k\} - \min\{x_1, c_i\} \) are included in the sum.

Given the transition probabilities, we can easily calculate the steady state probability vector of this Markov chain \( \pi^{(i)}_k = \begin{bmatrix} \pi^{(i)}_1 & \pi^{(i)}_2 & \ldots & \pi^{(i)}_{Q_k} \end{bmatrix} \) where \( \pi^{(i)}_{Q_k} \) denotes the probability that there are \( i \) packets in queue \( k \).

Assume that we choose the SINR switching thresholds for the AMC. The buffer overflow probability for queue \( k \) of the multi-hop route can be calculated as a ratio between the average number of dropped packets due to overflow at queue \( k \) (denoted as \( \bar{a}_k \)) and the average number of packets arriving at queue \( k \) in one time slot (denoted as \( \bar{a}_k \)). Hence, the buffer overflow probability at queue \( k \) can be calculated as \( P_f^{(i)} = \frac{\bar{a}_k}{\bar{a}_k} \). The average arrival rate of traffic to queue \( k \) can be written as \( \bar{a}_k = \sum_{l=1}^{Q_k} \sum_{j=0}^{\min\{l, c_i\}} \pi^{(i)}_l p^{(i)}_{l,j} (A.3) \), where \( B^{(i)} \) is the maximum bulk size of the arrival process.

The probability that \( i \) packets are successfully transmitted at queue \( k \) and arrive at queue \( k + 1 \) can be written as

\[
a^{(i+1)}_{(i)} = \sum_{j=0}^{Q_k} \sum_{l=0}^{K} \pi^{(j)}_l p^{(j)}_{l,j} (A.3) \]

These arrival probabilities are used to derive the queueing solution for queue \( k + 1 \) as in the presented procedure. And the average number of dropped packets due to overflow at queue \( k \) can be calculated as

\[
\bar{a}_k = \sum_{i=1}^{Q_k} \sum_{j=0}^{\min\{l, c_i\}} \max\{0, i + j - Q_k\}. \tag{A.4}
\]

Finally, the end-to-end loss probability can be written as

\[
P_L = 1 - (1 - p^{(i)}_1) \prod_{i=1}^{k} (1 - p^{(i)}_1) \tag{A.5}
\]

where the loss due to overflow at both buffers and due to channel errors are taken into account.

The end-to-end delay is the sum of delays that any packet experiences in all queues and links along its routing path. We ignore the transmission delay and only include queuing delay in the calculation. Using the Little’s law, the end-to-end average delay can be written as

\[
D = \sum_{k=1}^{L} \bar{a}_k (1 - p^{(i)}_1) \tag{A.6}
\]

where the numerator of each term in the summation is the average queue length of each queue and the denominator is the average arrival rate considering packet loss due to overflow.

Above results can be used to support the multi-channel scenario in LaSo. Assume that there is another traffic from another channel to queue \( k (k \geq 2) \) with arrival probability vector \( e^{(i)} \) besides the endogenous relayed traffic with probability vector \( a^{(i)}_k \). Then, the queueing performance for queue \( k \) can be calculated with aggregate arrival probability vector \( e^{(i)} = a^{(i)} \otimes e^{(i)} \), where \( \otimes \) denotes the convolution operation.

References


Di Wu received the B.E. degree in Communication Engineering and the M.E. degree in Communication and Information Systems from Hunan University, Changsha, China, in 2004 and 2008 respectively. He was a staff research associate at the University of California, Irvine, USA, a visiting researcher at IBM Research—China, and a student research associate at the SRI International, Menlo Park, USA. He is currently working toward the Ph.D. degree in Computer Science at the University of California, Irvine, USA. His research interests include wireless networks and mobile computing, location-based services, cyberphysical systems, mobile social networks, cloud computing and crowdsourcing, etc.

Lichun Bao received the B.S. degree in Computer Science from the University of Science and Technology of China, Hefei, China, in 1994, the M.E. degree in Computer Engineering from Tsinghua University, Beijing, China, in 1997, and the Ph.D. degree in Computer Science from the University of California, Santa Cruz, USA, in 2002. He was an assistant professor at the University of California, Irvine, USA. He is currently a senior software engineer at F5 Networks, Inc., San Jose, USA. His research interests include networking software design and development, system architecture design and integration, etc.

Amelia C. Regan received a B.A.S. degree in Systems Engineering from the University of Pennsylvania, an M.S. in Applied Mathematics from the Johns Hopkins University and an M.S. and Ph.D. in Transportation Systems Engineering from the University of Texas, Austin. She is currently a professor of computer science and transportation systems engineering at the University of California, Irvine. Her research is focused on algorithm development for optimization of transportation and communication systems. She has also taught short courses at the Athens University of Business and Economics and the National Technical University of Denmark and was an operations research analyst with the Association of American Railroads and United Parcel Service prior to earning the Ph.D. degree.

Carolyn L. Talcott is a Program Director in the Computer Science Laboratory of SRI International. Dr. Talcott holds a Ph.D. in Computer Science from Stanford University (1985) and a Ph.D. in Chemistry from the University of California, Berkeley (1986). She has over 25 years experience in formal modeling and analysis. For the last 11 years she has also been working in computational modeling of biological systems. At SRI Dr. Talcott is leading research in symbolic systems biology, security protocol analysis, and formal analysis applied to cyber-physical systems and next-generation networks.