

# A Framework for Topology-Transparent Scheduling in Wireless Networks

Qiong Sun, Victor O.K. Li, and Ka-Cheong Leung  
Department of Electrical and Electronic Engineering  
The University of Hong Kong, Hong Kong SAR, China  
E-mail: {joansun, vli, kcleung}@eee.hku.hk

**Abstract**—Transmission scheduling is a key design problem in wireless multi-hop networks. Many transmission scheduling algorithms have been proposed to maximize the spatial reuse and minimize the time division multiple access (TDMA) frame length. There exists some interesting scheduling algorithms called topology-transparent TDMA scheduling algorithms, which do not require the detailed topology information, and are suitable for the wireless environment. However, a framework to compare the performance of these algorithms properly and fairly is still lacking. The objective of this work is to propose a uniform framework for topology-transparent scheduling algorithms. Under some fundamental constraints, an optimal solution is provided to the scheduling problem of topology-transparent algorithms. Furthermore, under the proposed framework, we analyze the relationship among all existing topology-transparent algorithms. We then develop an adaptive topology-transparent algorithm, which can always give an optimal solution under a set of the system design parameters.

**Index Terms**—Adaptive, distributed scheduling, TDMA, topology-transparent, wireless networks.

## I. INTRODUCTION

Wireless networks have unique characteristics, which must be properly accounted for in transmission scheduling. This is especially true when the network topology is not fixed because of node movements and limited transmission power. Designing an effective distributed transmission scheduling algorithm in multi-hop mobile ad hoc networks is a challenging research issue.

A proper scheduling design not only allows the channel to be shared by all nodes in both time and space domains, but also maximizes the channel utilization and minimizes the packet delay. To achieve efficiency and robustness in the wireless environment, topology-transparent algorithms have been proposed [1], [2], [3], [4], [5], [6], [7]. In these algorithms, collisions may occur but no topology information is used. Hence, no topology updates are required. From the perspective of design methodologies, these existing topology-transparent scheduling algorithms can be classified into two categories. The first one is based on coding theory, such as Galois Field (GF) [1], [2], [3], [4], [6], and Latin Square (LS) [5], [8]. The other category is derived from the theory of Block Design (BD) [7]. All these algorithms aim to maximize the minimum throughput guarantee, but a common framework to compare them is still lacking.

In this paper, we propose a framework for these existing topology-transparent algorithms. The framework allows one to

not only compare the performance of these algorithms properly and fairly, i.e. under what scenarios a particular scheduling algorithm can give the best performance, but also to find an adaptive scheduling algorithm. It enables us to investigate different categories of the topology-transparent scheduling algorithms offering a certain minimum performance guarantee and to study the relationship among these algorithms. Under our framework, we can determine which algorithm performs better with the given set of the system design parameters. Based on the proposed framework, a node can automatically choose a proper algorithm addressing different network scenarios based on the given system design parameters.

The rest of the paper is organized as follows. The system model is introduced in Section II. Section III presents our proposed uniform framework for the topology-transparent scheduling algorithms in details. In Section IV, three representative topology-transparent scheduling algorithms are introduced and compared under the proposed framework. An adaptive scheduling scheme is also proposed. Finally, we conclude in Section V.

## II. SYSTEM MODEL

We consider a time slotted system where time is the resource to be shared among all nodes. In addition, we focus on the scheduling problem for a given transmission channel. The access method for the channel is time division multiple access (TDMA). We assume that the transmission channel is error-free and a reception failure is only due to packet collisions. All nodes are homogeneous, and cannot transmit and receive simultaneously.

A wireless multi-hop network can be modeled as a bi-directional graph  $G_v = (V, E)$ , where  $V$  is the set of nodes and  $E$  is the set of edges to indicate which pair of nodes is connected. The number of nodes  $|V|$  is denoted by  $N$ . The degree of a node  $v$  is defined as the number of its neighbors, and it is always less than or equal to the maximum node degree  $D_{max}$ . For simplicity, the transmission range of a node is the same as its interference range. All nodes suffer two types of conflicts [9]. A primary conflict occurs if two or more nodes transmit simultaneously to the same destination node, while a secondary conflict occurs when a packet reception is interfered by transmissions destined to other nodes.

### III. TOPOLOGY-TRANSPARENT SCHEDULING FRAMEWORK

Based on two system parameters  $N$  and  $D_{max}$ , the topology-transparent scheduling algorithms are designed such that each node in the network is guaranteed a minimum number of successful transmissions per TDMA frame no matter how the topology changes. The aim of these algorithms is to improve the performance of a node or the system as a whole under certain minimum performance requirements. In this section, we are going to devise a framework for topology-transparent scheduling so as to satisfy the given performance requirements.

#### A. Problem Formulation

A scheduling policy is a rule that specifies which node is scheduled to transmit at which time slot. For the existing topology-transparent scheduling policies, time is divided into equal-length slots, grouped into equal-length frames. During one time frame, a node is scheduled based on a unique codeword assigned to it.

**Definition 1:** A codeword is defined as the sequence of the selected time slots at which a node can transmit.

The codeword assigned to Node  $v$  is denoted as  $CD_v$ . During a time frame, if Node  $v$  is scheduled in time slots 1, 5, 12, and so on. Its codeword  $CD_v = (1, 5, 12, \dots)$ .

Although these topology-transparent scheduling policies have different methods to schedule nodes, they share the same objective, say, to maximize the minimum guaranteed throughput of each node or of the system as a whole.

**Definition 2:** The throughput of a node  $G$  is defined as the ratio of the number of successful transmissions for the node in each frame to the frame length  $L$ .

The minimum throughput of each node is:

$$G_{min} = \frac{T_{min}}{L} \quad (1)$$

where  $T_{min}$  is the minimum number of guaranteed successful transmissions for each node in a time frame. Thus,  $T_{min} = T - CD_{max}$ .  $T$  is the number of transmissions for a node in a time frame.  $C$  is the maximum number of collisions between any two nodes per time frame. They can be determined by the selected scheduling policy.

In order to guarantee the minimum throughput for a node, the following two constraints must be satisfied.

**Constraint 1:** The required number of unique codewords should be at least the total number of nodes in the network.

Each node should be assigned a distinct codeword to transmit. Otherwise, transmissions from nodes with the same codeword may collide with each other all the time. Thus, the number of unique codewords should be at least the total number of nodes in the network. We use  $Q$  to indicate the code family.

**Definition 3:** A code family is defined as a set including all possible codewords.

It is also decided by the selected scheduling policy. Hence, for a scheduling policy, its code family can be written as  $Q =$

$\{CD_i, i = 1, 2, \dots, N, \dots\}$ . The number of codewords is the size of the code family, denoted by  $|Q|$ . From Constraint 1,  $|Q| \geq N$ .

**Constraint 2:** The number of conflict-free transmission opportunities for a node is at least  $\alpha$  ( $\alpha \geq 1$ ) within a time frame.

Constraint 2 can guarantee that each node can have at least one conflict-free transmission opportunity in a time frame.

Thus, the problem maximizing the minimum guaranteed throughput using Policy  $S$  with Constraints 1 and 2 can be stated as follows:

$$\max_{S \in \Theta} G_{min}(S) = \frac{T - CD_{max}}{L} \quad (2)$$

subject to

$$\begin{cases} |Q| \geq N \\ T \geq CD_{max} + \alpha \end{cases} \quad (3)$$

where  $S \in \Theta$ ,  $\Theta$  is the set of all topology-transparent scheduling policies, and  $\alpha \geq 1$  so that the number of conflict-free transmission opportunities for a node should be at least one within a time frame.

#### B. Optimal Policy

We define a policy  $S^*$  for Node  $i$  as follows:

$$S^*(\vec{X}) = \operatorname{argmax}_i (G_{min}(S_i)) \quad (4)$$

where  $\vec{X} = (x_1, x_2, \dots, x_N)$  represents a set of the decision variables utilized by Policy  $S^*$ .

**Proposition 1:** Policy  $S^*$  is a solution to the problem defined in (2). That is, it maximizes the minimum guaranteed throughput of a node under the given constraints.

With the given values of  $N$  and  $D_{max}$ , the optimal scheduling policy  $S^*$  can be found by maximizing the minimum guaranteed throughput. This  $S^*$  is defined based on the relationship among  $C$ ,  $L$ ,  $|Q|$ ,  $N$ , and  $D_{max}$ . In other words, we can find a scheduling policy  $S^*$  to maximize the minimum guaranteed throughput with the given  $N$  and  $D_{max}$  by (2).

### IV. COMPARISON OF SCHEDULING POLICIES UNDER THE FRAMEWORK

As far as we know, there are three different types of topology-transparent scheduling policies based on Galois Field (GF), Latin Square (LS), and Block Design (BD). In this section, we first briefly introduce these three scheduling policies (where details can be found in [4], [5], and [7]) and then compare the performance of these policies under the proposed framework.

#### A. Three Types of Topology-Transparent Scheduling Policies

1) *Galois Field (GF) Scheduling Policy:* In the GF scheduling policy, each time frame is divided into  $q$  subframes, each of which consists of  $p$  time slots. During a subframe, each node can select an assigned time slot to transmit. Thus, each node can have  $q$  transmissions within a time frame. The time frame structure is shown in Fig. 1. For a given node, its transmission

slots are determined by its assigned polynomial through the following rules, as discussed in [4].

**Rule 1:** For a given network, each node  $v$  chooses a unique time slot allocation function (TSAF)  $f_v(x) = \sum_{i=0}^k a_i x^i \pmod{p}$ , where  $v \in V$ . The function is used to calculate the position of a transmission slot selected in a frame for Node  $v$ .

**Rule 2:** Let a standard row vector  $S$  be  $(0, 1, \dots, q-1)$ . Define another row vector  $f_v(S) = (f_v(0), f_v(1), \dots, f_v(q-1))$ , known as the time slot location vector (TSLV) for Node  $v$ .

A TSLV indicates which time slots are selected for a node per frame. For example, in Fig. 1, for Node  $v$  in Subframe  $i$ , the selected transmission time slot is given by  $f_v(i) \pmod{p}$ . If  $f_v(0) = 2$ ,  $f_v(1) = 1$ , Node  $v$  chooses the third time slot in Subframe 0 and the second time slot in Subframe 1. Thus, the TSLV of Node  $v$  is  $f_v(S) = (2, 1, \dots)$ .

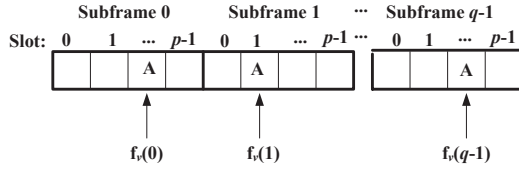


Fig. 1. The frame structure of the GF scheduling policy.

**Property 1:** For a set of TSAFs with degree  $k$ , if  $q \leq p$ , any two TSAFs have the same time slot selection for at most  $k$  times. This indicates that the number of collisions for any two nodes per frame is at most  $k$  [4].

In order to ensure that every node has at least the minimum guaranteed throughput, Constraints 1 and 2 must be satisfied. Thus, for the GF scheduling policy, the maximum value of the minimum guaranteed throughput is:

$$\max_{GF \in \Theta} G_{min}(GF) = \frac{q - kD_{max}}{pq} \quad (5)$$

subject to

$$\begin{cases} p^{k+1} \geq N \\ q \geq kD_{max} + \alpha \end{cases} \quad (6)$$

**2) Latin Square (LS) Scheduling Policy:** A Latin Square of order  $n$  is an  $n \times n$  square array composed of  $n$  symbols from one to  $n$  such that each symbol appears once in each row and once in each column. Two distinct  $n \times n$  Latin Squares  $A = (a_{i,j})$  and  $B = (b_{i,j})$ , where  $a_{i,j}, b_{i,j} \in \{1, 2, \dots, n\}$ , are said to be orthogonal if the  $n^2$  ordered pairs  $(a_{i,j}, b_{i,j})$  are all different. The square arrays A and B shown below are examples of the orthogonal Latin Squares of order four.

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

The scheduling policy using LS is constructed as follows:

- Each row in a Latin Square corresponds to a subframe. If a Latin Square has  $q$  rows, it can correspond to at most  $q$  subframes.
- A symbol corresponds to a transmission slot pattern. If a Latin Square has  $p$  symbols, it can associate up to  $p$  time slots in a subframe.
- A time frame is divided into  $q$  time subframes. Each subframe has  $p$  time slots. Each node will be assigned a time slot for possible transmission based on the allocated symbol.

For example, if  $p = 4$ ,  $q = 3$ , and Node  $v$  corresponds to the symbol "2" in Latin Square A, the transmission schedule is: Slot 2 in Subframe 1, Slot 1 in Subframe 2, and Slot 4 in Subframe 3.

**Property 2:** From the above scheme, it is clear that if two nodes correspond to two different symbols from the same Latin Square, their transmissions will never be in conflict at any time. If two nodes corresponds to two symbols from two different Latin Squares, their transmissions will have at most one conflict in any frame.

To guarantee that each node will have the required minimum number of successful transmissions per time frame, Constraints 1 and 2 must be satisfied. Thus, the maximum value of the minimum guaranteed throughput for the LS scheduling policy is as follows:

$$\max_{LS \in \Theta} G_{min}(LS) = \frac{q - D_{max}}{pq} \quad (7)$$

subject to

$$\begin{cases} pN(p) \geq N \\ q \geq D_{max} + \alpha \end{cases} \quad (8)$$

where  $N(p)$  is the number of orthogonal Latin Squares of order  $p$ .

**3) Block Design (BD):** A block is simply a subset of a set of treatments. In topology-transparent scheduling algorithms, treatments are just time slots. The objective of a block design is to choose the blocks with certain properties. A block design is called incomplete if at least one chosen block does not contain all treatments. A balanced incomplete block (BIB) design [10] is an arrangement of  $a$  treatments into  $b$  blocks such that:

- 1) each block contains  $j$  ( $< a$ ) distinct treatments,
- 2) each treatment appears in  $r$  blocks, and
- 3) any pair of treatments appears together in  $\lambda$  blocks.

An ordered set of integers  $(a, b, j, r, \lambda)$  is called the parameter set of the BIB design. These parameters are not independent, since  $ar = bj$ ,  $\lambda(a-1) = r(j-1)$ , and  $b \geq a$  [10]. The BD design is used for the allocation of the transmission time slots such that:

- Each time frame has  $a$  time slots.
- Each node can transmit up to  $j$  time slots in a frame.
- There are  $b$  distinct blocks to be assigned to all  $N$  nodes.
- For any two nodes, there are at most  $S_U$  collisions.

**Property 3:** The symmetric block design [10] is used to schedule nodes, so that every block contains  $j$  treatments.

Every treatment occurs in  $j$  blocks. Any pair of elements co-exists in  $\lambda$  blocks. Every pair of blocks have  $\lambda$  treatments in common. Hence, the number of collisions between any two nodes is  $\lambda$ , i.e.,  $S_U = \lambda$ .

Suppose a series of symmetric BIB [7] is defined as follow:

$$\begin{cases} a = 2f(2f\lambda - 1) + 1 \\ b = a \\ r = j = 2f\lambda \end{cases} \quad (9)$$

where  $f, \lambda \in \mathbb{N}$ .

In addition, Constraints 1 and 2 should be satisfied to guarantee the minimum throughput. Hence, the maximum value of the minimum guaranteed throughput is:

$$\max_{BD \in \Theta} G_{min}(BD) = \frac{2f\lambda - \lambda D_{max}}{2f(2f\lambda - 1) + 1} \quad (10)$$

subject to

$$\begin{cases} b = 2f(2f\lambda - 1) + 1 \geq N \\ j = 2f\lambda \geq \lambda D_{max} + \alpha \end{cases} \quad (11)$$

### B. Relationship Among Scheduling Policies

Now, we are going to analyze these three types of topology-transparent scheduling policies introduced so as to evaluate and compare their performance through our proposed framework.

1) *GF and LS Policies*: By investigating the mathematical expressions for the maximum value of  $G_{min}$  and the corresponding constraints for GF and LS as shown in (5) and (7), we find that they can fit into our framework in (2) directly by the following configuration:

$$\begin{cases} T = q \\ C = k \quad (GF) \\ C = 1 \quad (LS) \\ L = pq \\ |Q| = p^{k+1} \\ \alpha = 1 \end{cases} \quad (12)$$

Since GF is more general and LS is just a special case of GF when  $k = 1$ , we shall simply discuss the GF policy from now on.

2) *BD Policy*: The design of the BD policy follows a different approach from GF and LS. We shall show that it can also be included into our framework after some transformations. The maximum value of  $G_{min}$ , as shown in (10), can be expressed in the form of (2) with the following configuration:

$$\begin{cases} T = 2f\lambda \\ C = \lambda \\ L = 2f(2f\lambda - 1) + 1 \\ |Q| = 2f(2f\lambda - 1) + 1 \\ \alpha = 1 \end{cases} \quad (13)$$

3) *Relationship Analysis*: Up till now, we can see that even though the design strategies for GF and BD are quite different, they can both fit into our framework properly.

The mathematical expressions of maximizing the minimum guaranteed throughput using GF and BD become identical by setting the parameters as follows:

$$\begin{cases} q = 2f\lambda \\ k = \lambda \\ p = \frac{2f(2f\lambda - 1) + 1}{2f\lambda} \end{cases} \quad (14)$$

Therefore, the objective functions of maximizing  $G_{min}$  for these types of policies are the same. Now, we are going to examine the constraints for GF as shown in (6), while the constraints for BD is as follows:

$$\begin{cases} p \cdot 2f\lambda \geq N \\ q - kD_{max} \geq \alpha \end{cases} \quad (15)$$

We can see that, by comparing these two types of policies, their second constraints are the same and the only difference lies in the first constraints. For GF,  $p \geq N^{\frac{1}{k+1}}$  must be satisfied, while  $p \geq \frac{N}{2f\lambda}$  for BD. Therefore, we have transformed the comparison of these two types of scheduling policies into a simple optimization problem with different constraints on  $p$ . With the same objective function, Policy A with  $p \geq c_A$  performs at least as good as Policy B with  $p \geq c_B$  if and only if  $c_A \leq c_B$ . Indeed, the feasible region of A includes that of B. The size of the feasible region depends on the values of  $k, f$ , and  $\lambda$  under a given value of  $N$ . Let  $N^{\frac{1}{k+1}} = \frac{N}{2f\lambda}$ . That is,  $N = (2f\lambda)^{1+\frac{1}{k}}$ . GF and BD yield the same optimal performance since their feasible regions are the same. Similarly, when  $N > (2f\lambda)^{1+\frac{1}{k}}$  (i.e.  $N^{\frac{1}{k+1}} > \frac{N}{2f\lambda}$ ), the feasible region of BD includes that of GF. BD performs at least as good as GF does. Consider that  $q - kD_{max} \geq \alpha$  is the only constraint to the captioned maximization problem. As shown in Fig. 2, if the optimal value  $opt^*$  is located in the area R3, which is the overlapping region of the feasible regions of BD and GF, both BD and GF can achieve the same optimal performance. If  $opt^*$  is in R2, which is outside the feasible region of GF but within the feasible region of BD, only BD can yield the optimal performance. If  $opt^*$  is found in R1, which is outside the feasible regions of BD and GF, both cannot achieve that optimal performance. However, since the feasible region of BD includes that of GF, BD performs at least as good as GF does. When  $N < (2f\lambda)^{1+\frac{1}{k}}$  (i.e.  $N^{\frac{1}{k+1}} < \frac{N}{2f\lambda}$ ), the feasible region of GF includes that of BD, GF can perform at least as good as BD does. Thus, the relationship between the feasible regions of BD and GF, and their performance can be summarized as follows:

$$\begin{cases} N^{\frac{1}{k+1}} < \frac{N}{2f\lambda}, \text{ GF performs at least as good as BD does;} \\ N^{\frac{1}{k+1}} = \frac{N}{2f\lambda}, \text{ GF and BD have the same performance;} \\ N^{\frac{1}{k+1}} > \frac{N}{2f\lambda}, \text{ BD performs at least as good as GF does.} \end{cases} \quad (16)$$

### C. Illustration

Here, we use an example to illustrate the relationship between BD and GF with different system design parameters, namely,  $N$  and  $D_{max}$ .

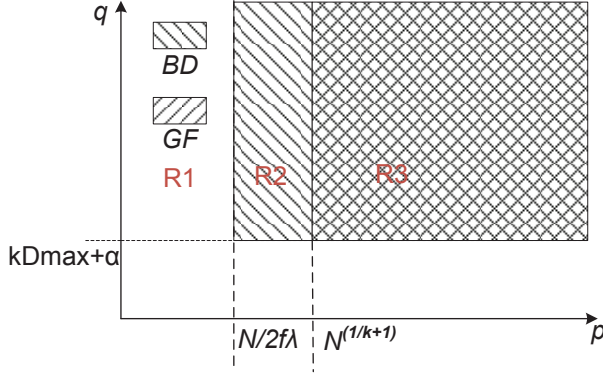


Fig. 2. The feasible regions of the BD and GF scheduling policies.

Let the set of system design parameters be configured as ( $N = 100, D_{max} = 8$ ). For GF, the maximum value of the minimum guaranteed throughput  $G_{min}(GF) = 0.0101$  with the optimal values  $k = 2, p = 11$ , and  $q = 9$  according to [1]. Thus, the feasible region of GF, denoted by  $FR(GF)$ , is  $\{(p, q) | p \geq 11, q \geq 9\}$ . For BD, the maximum value of the minimum guaranteed throughput  $G_{min}(BD) = 0.0332$  with the set of optimal parameters  $f = 8$  and  $\lambda = 1$  according to [7]. Hence, the feasible region of BD,  $FR(BD)$  is equivalent to  $\{(p, q) | p \geq 7, q \geq 9\}$ . Thus,  $FR(GF) \subset FR(BD)$ . This means that BD performs at least as good as GF does. Indeed, as shown in this example,  $G_{min}(BD) > G_{min}(GF)$ .

Table I shows the relationship between GF and BD with different configurations of ( $N, D_{max}$ ). According to the analysis, we can find whether GF or BD performs better with the given configuration of ( $N, D_{max}$ ).

TABLE I  
PERFORMANCE COMPARISON BETWEEN GF AND BD

( $N, D_{max}$ )	FR Comparison	$G_{min}(GF)$	$G_{min}(BD)$
(16, 3)	$FR(GF) \subset FR(BD)$	0.0476	0.0968
(64, 5)	$FR(GF) \subset FR(BD)$	0.0152	0.0549
(500, 7)	$FR(GF) \subset FR(BD)$	0.0095	0.0169
(1000, 5)	$FR(GF) \supset FR(BD)$	0.0083	0.0052

#### D. Adaptive Scheduling Scheme

According to the previous discussion, we know that a scheduling policy performs at least as good as another if its feasible region includes that of the one. Thus, we can design an adaptive scheduling algorithm, which can choose between GF and BD based on their feasible regions. This adaptive algorithm is summarized as follows:

- 1) Based on the values of  $N$  and  $D_{max}$ , the feasible regions of GF and BD are determined as  $FR(GF)$  and  $FR(BD)$ , respectively.
- 2) Comparing  $FR(GF)$  with  $FR(BD)$ . If  $FR(GF)$  includes  $FR(BD)$ , GF is selected; otherwise, BD is selected.
- 3) If either  $N$  or  $D_{max}$  changes, repeat 1) and 2).

#### V. CONCLUSION

In this paper, a framework for the topology-transparent scheduling algorithms is proposed. It aims to compare the topology-transparent algorithms properly and fairly. First, we introduce three representative types of topology-transparent scheduling algorithms based on Galois Field (GF), Latin Square (LS), and Block Design (BD). Moreover, under the proposed framework, we discuss the relationship among these three types of algorithms. Since LS is a special case of GF, we focus on finding the relationship between GF and BD. Their feasible regions are calculated and compared. With the analysis, we find that an algorithm with a larger feasible region likely yields a better performance. We also illustrate this observation with an example. Furthermore, we devise an adaptive topology-transparent scheduling algorithm, which can choose a suitable algorithm to maximize the system performance based on the feasible regions of GF and BD under the given system parameters,  $N$  and  $D_{max}$ .

#### REFERENCES

- [1] Z. Cai, M. Lu, and C. Georgiades, "Topology-transparent time division multiple access broadcast scheduling in multihop packet radio networks," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 4, pp. 970–984, Jul. 2003.
- [2] I. Chlamtac and A. Farago, "Making transmission schedules immune to topology changes in multi-hop packet radio networks," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 23–29, Feb. 1994.
- [3] F. Farnoud and S. Valaee, "Reliable broadcast of safety messages in vehicular ad hoc networks," in *Proc. IEEE INFOCOM'09*, Apr. 2009, pp. 226–234.
- [4] J.-H. Ju and V. O.K. Li, "An optimal topology-transparent scheduling method in multihop packet radio networks," *IEEE/ACM Transactions on Networking*, vol. 6, no. 3, pp. 298–306, Jun. 1998.
- [5] K. Kim and V. Prasanna, "Latin squares for parallel array access," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 4, pp. 361–370, Apr. 1993.
- [6] K. Oikonomou and I. Stavrakakis, "Analysis of a probabilistic topology-unaware TDMA MAC policy for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 7, pp. 1286–1300, Sep. 2004.
- [7] Y.-S. Su, S.-L. Su, and J.-S. Li, "Topology-independent link activation scheduling schemes for mobile cdma ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 599–616, May 2008.
- [8] J.-H. Ju and V. O.K. Li, "TDMA scheduling design of multihop packet radio networks based on Latin squares," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1345–1352, Aug. 1999.
- [9] A. Ephremides and T. Truong, "Scheduling broadcasts in multihop radio networks," *IEEE Transactions on Communications*, vol. 38, no. 4, pp. 456–460, Apr. 1990.
- [10] A. Dey, *Theory of Block Designs*. John Wiley & Sons, New York, 1986.