This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2007 proceedings.

CC-TDMA: Coloring- and Coding-based Multi-channel TDMA Scheduling for Wireless Ad Hoc Networks

Xuedan Zhang*, Jun Hong*, Lin Zhang*, Xiuming Shan* and Victor O. K. Li†
*Department of Electronic Engineering, Tsinghua University, Beijing, P.R.C.
†Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, P.R.C.

***Abstract*— This paper addresses the issue of transmission scheduling in multi-channel wireless ad hoc networks. We propose a multi-channel Time Division Multiple Access (TDMA) scheduling based on edge coloring and algebraic coding theory, called CC-TDMA. We categorized the conflicts suffered by wireless links into two types: explicit conflicts and implicit conflicts, and CC-TDMA utilizes two different strategies to deal with them. Explicit conflicts are avoided completely by a simple distributed edge-coloring algorithm μ-M, and implicit conflicts are minimized by using coding theory to assign channels to links. We evaluate CC-TDMA analytically and numerically, and find that it exhibits a better performance than previous work in terms of throughput and delay.**

I. INTRODUCTION

In this paper, we propose a novel distributed multi-channel TDMA scheduling method based on the concepts of edge coloring and algebraic coding theory. We categorize the conflicts suffered by wireless links into two types: explicit conflicts and implicit conflicts, and design different strategies, derived from edge coloring and algebraic coding, respectively, to deal with them. To our best knowledge, this is the first paper which utilizes both edge coloring and algebraic coding in the domain of multi-channel TDMA scheduling in wireless ad hoc networks. Our method gives a better performance than previous work in terms of throughput and delay.

The rest of the paper is organized as follows: Section II describes the network model and formally defines the problems. Section III presents the details of our scheduling algorithm. Section IV evaluates the performance of CC-TDMA through computer simulations. Section V concludes the paper.

II. MODEL AND DEFINITIONS

We represent a wireless ad hoc network as a directed graph $G = (V, E)$, where V is a set of vertices denoting the nodes comprising the network and E is a set of directed edges between vertices representing inter-node wireless links. Let $N = |V|$. Henceforth, the terms vertex and wireless node, as well as edge and wireless link, are used interchangeably in this paper.

This research is supported in part by the National Science Foundation of China (Grant No. 60672107), the Hi-tech Research and Development Program of China (Grant No. 10Z2), and China 973 Project (Grant No. 2007CB307105).

We denote D_{max} as the maximum number of neighbors, and Δ as the maximum degree of a node in G. We assume that each pair of neighboring nodes have bi-directional wireless links; hence the graph is not a simple graph but a multigraph and $\Delta = 2D_{max}$. Each node is assumed to be equipped with a single transceiver with M orthogonal channels. The transceiver, which is capable of switching channels dynamically, can only transmit or receive packets on one channel at a time. The transmission of a link between two nodes is successful if 1) the two nodes are within communication range of each other, 2) neither of the transceivers is being occupied by other links, 3) the transceivers are on a common channel, and 4) no node within a receiving node's interference range is transmitting using the same channel. Note that node pairs using different channels can transmit packets simultaneously without interference in a shared neighborhood. In our algorithm, we will use N and Δ as design parameters.

There are two kinds of conflicts suffered by wireless links in wireless ad hoc networks. The first kind of conflicts, called *explicit conflicts*, is caused when more than one link share a common node's transceiver at the same time. The second kind of conflicts, called *implicit conflicts*, is a consequence of wireless interference on the same channel in a neighborhood.

Consider the wireless network with a chain topology in Figure 1. There are four nodes in the network and the distance between the adjacent nodes is $100m$. The nodes in the network share four wireless channels and use the same fixed transmission power, with a transmission range of $100m$ and an interference range of $250m$. It is not difficult to see that the conflicts between $L1$, $L2$, $L3$ and $L4$ or between $L3$, $L4$, $L5$ and $L6$ are explicit conflicts and the conflicts between $L1$ and $L6$ or between $L2$ and $L5$ are implicit conflicts.

One can treat all the transmission time slots on all channels as a two-dimensional (time and channel) transmission scheduling problem. Thus, we use two orthogonal dimensions of the transmission scheduling space to solve the above two kinds of conflicts. Assigning different transmission time slots to links is the only effective way to avoid explicit conflicts. We use a distributed edge coloring algorithm which only requires local information, i.e. information from immediate neighbors, to assign time slots to links. Avoiding implicit conflicts needs topology information from nodes out of transmission range. Since it is difficult to get such information in wireless ad hoc networks, we propose a coding algorithm, which minimizes the implicit conflicts without requiring any topology information, to assign wireless links on different channels. We shall elaborate on the details of our algorithm in the next section.

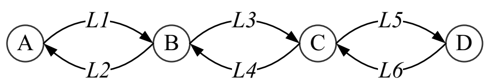


Fig. 1. A chain-topology wireless ad hoc network

III. SCHEDULING ALGORITHM

A. Explicit conflicts

Explicit conflicts are the consequence of the competition of multiple simultaneous wireless links for the single transceiver at a node. Assigning distinct channels to the links cannot solve this type of conflicts. The only way to avoid explicit conflicts is to assign different transmission time slots to each link. In Figure 1, $L1$, $L2$, $L3$ and $L4$ must be allocated to different time slots, as must $L3$, $L4$, $L5$ and $L6$. In CC-TDMA, time is divided into transmission slots of the same length, and grouped into sub-frames and frames. A frame consists of n sub-frames, and each sub-frame consists of l time slots. Each link is assigned an integer between 0 and $l-1$, which represents the assigned time slots in the n sub-frames of the frame, so that the explicit conflicts are avoided completely. The frame structure of scheduling for the links in Figure 1 is shown in Figure 2.

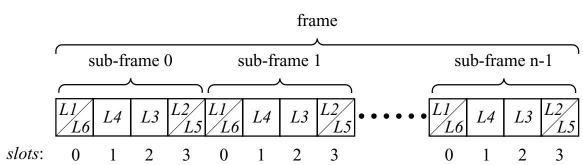


Fig. 2. An simple of time-slotted frame structure of CC-TDMA

The time slot assignment problem for avoiding explicit conflicts can be modeled as a standard *edge coloring* problem. In time slot scheduling, two links that have explicit conflicts cannot be assigned the same time slot. Similarly, in a valid edge coloring, no two edges incident on the same node are assigned the same color. A time slot assignment for explicit conflicts can be obtained by mapping each time slot to a color and thus the minimum edge coloring is equivalent to the optimal scheduling for this type of conflicts.

Vizing's theorem [1] states that a valid edge coloring for a graph can be obtained by using at most $\Delta + \mu$ colors, where μ is the maximum multiplicity of an edge. However, the existing distributed edge coloring algorithms cannot achieve the optimal. Panconesi and Srinivasan [2] proposed a distributed edge coloring algorithm that uses at most $(2\Delta - 1)$ colors and then they proposed a better distributed algorithm [3] that requires at most $1.6\Delta + O(\log^{1+\Delta} N)$ colors. Grable and Panconesi [4] designed a randomized distributed edge coloring algorithm which uses at most $(1+\varepsilon)\Delta$ colors, where $\varepsilon > 0$ is an input parameter. Their algorithm computes nearly optimal edge colorings very quickly but may fail to assign a valid color to each edge in the graph. Gandham *et al.* [5] developed a distributed implementation of Misra and Gries's centralized edge coloring method [6]. Their algorithm can obtain a valid edge coloring of acyclic graphs using at most $(\Delta + 1)$ colors. But their algorithm includes two complicated coloring rounds and needs large numbers of message exchanges by the nodes. The algorithms above are not suitable for distributed slot assignment in wireless ad hoc networks.

Algorithm **M** designed by Marathe *et al.* [7] inspired our proposed algorithm, which is a simple modification of the algorithm in [4] for edge-coloring simple graphs but with substantially improved performance. Unlike the original algorithm, this modified algorithm never fails. If the graph $G = (V, E)$ satisfies the condition $\Delta \gg \log N$, **M** can obtain the nearly optimal edge-coloring of G. Its implementation is rather simple and nodes work in parallel by exchanging information only with their immediate neighbors. Besides, its performance in terms of both running time and number of colors used is quite good.

We develop an implementation of **M** for edge-coloring multigraphs at each wireless node and call it μ-**M**. The sender of each directed edge is in charge of coloring. Each edge has its own palette whose initial size is Δ, which is the number of colors allowed to be used. μ-**M** executes the following four steps repeatedly until all edges are colored.

1) Each sender chooses uniformly at random a tentative color from the corresponding palette for each of its uncolored links, and guarantees that any two links do not select the same color.
2) For each uncolored link, the sender sends a message describing the tentative colors to the receiver, and the receiver checks the tentative colors against the colors of its colored edges for possible color collision. If a collision occurs, the receiver sends a failure message back, with information on its occupied colors. If no collision occurs, the receiver sends a success message back and removes the color from the palettes of its uncolored links.
3) If a success message is received, the sender marks the link colored and removes the color from the palettes of its uncolored links. Otherwise, the sender removes the colors in the message (the receiver's occupied colors) from the palettes of the links.
4) If any palette is empty, one fresh new color is added to it.

In μ-**M**, an edge-coloring of G corresponds to a timeslot assignment for explicit conflicts in a sub-frame and the n sub-frames keep the same timeslot scheduling in a frame. Obviously, l is the upper bound of the number of colors used by μ-**M**. For the application of wireless ad hoc networks, if $\Delta \gg \log N$, l is set to $\Delta + \mu$. Otherwise, μ-**M** is expected to use between 5% and 20% more colors than $\Delta + \mu$ ($\mu = 2$), and hence l is set to 1.25Δ in CC-TDMA.

B. Implicit conflicts

Implicit conflicts are a consequence of wireless interference between nodes within interference range. It is very difficult to describe this interference relation accurately because topology information out of transmission range is hard to capture precisely, and interference range irregularity [8] may change the interference relations among nodes over time. Hence, it is a challenge to avoid implicit conflicts between interfering wireless links, and we rely on a coding-based algorithm, which assigns channels without any topology information, to minimize implicit conflicts.

A primary goal of algebraic coding theory is to construct error-correcting codes with the largest possible minimum distance with respect to the length over finite fields. That is analogous to the principle of finding separate but efficient ways for the nodes of the network to transmit in a multiple access system. The utilization of coding theory for scheduling is not new. Chlamtac and Farago [9] are the first to propose a framework for the application of code-based scheduling in the context of wireless ad hoc networks. Each node is assigned a distinct polynomial P characterized by a degree k and coefficients in $GF(q)$, and the evaluation of the polynomial is used to select the slots in a frame. Ju and Li [10] use the same construction as [9] to find an optimum value for the parameters of the code that maximize a lower-bound on the minimum throughput. Recently, Rentel and Kunz [11] indicate that the fundamental method in [9] and [10] is restricted to the code-words of a singly-extended Reed-Solomon (RS) code since polynomial evaluation in $GF(q)$ is one of the possible methods to construct RS codes. The scheduling strategy based on coding theory is topology-transparent and able to guarantee a minimum level of performance. Ju and Li [12] propose a multi-channel topology-transparent algorithm based on Latin Squares, but assume that each node is equipped with a single transmitter and multiple receivers, which is different from our assumption.

TABLE I

A $[4, 2, 4]$ EXTENDED RS CODE.

CW0	0000	CW4	1203	CW8	3021	CW12	3210
CW1	0123	CW5	2301	CW9	1032	CW13	1111
CW2	0231	CW6	3102	CW10	1320	CW14	2222
CW3	0312	CW7	2013	CW11	2130	CW15	3333

In this paper, the concept of algebraic coding is extended to multi-channel scheduling in wireless ad hoc networks. Our approach, which solves implicit conflicts, is to use coding theory to assign channels to links. Each link will be assigned one transmission time slot and one channel in each sub-frame. After implementing μ-**M** to assign time slots, each link in the network selects a code word of a structured code. The selected code word represents the channels assigned to the link in the corresponding sub-frames. For example, a link selects a code word $(c(0), c(1), \ldots, c(n-1))$, where $c(i)$ represents the channel on which the link transmits in the i-th sub-frame. We assume that L_2 of Figure 1 chooses the code word $CW11$ in Table I. Figure 3 shows the corresponding channels assigned to L_2 in each sub-frame.

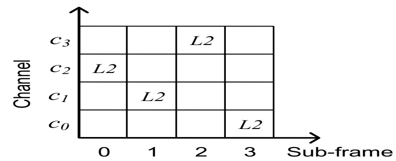


Fig. 3. The channel scheduling of L_2

As mentioned above, the links have been assigned time slots by μ-**M**. If two interfering links are assigned the same time slot and the elements at the corresponding position of the two links' code words have the same value, a collision in the corresponding sub-frame will occur because they transmit on one channel simultaneously. This implies that the larger the number of places where the two code words differ, the smaller the probability that the two links will collide. The Hamming distance measures the difference between two code words. We would like to find a code in which the Hamming distance of any two code words is as large as possible.

The throughput G is defined as the number of successful packets in each time slot, i.e. the ratio of the number of successful packets in each frame to the frame length.

$$G = \frac{np - p_c}{nl} \tag{1}$$

where p is the average number of packets sent per node in each sub-frame, $p \leqslant D_{max}$ and p_c is the number of collided packets per node in a frame. According to [10] and [11], a lower-bound throughput of a node can be written as,

$$\begin{aligned} G_{min} &= \frac{np - \sum_{i=1}^{p}(n - d_{min})I'_i}{nl} \\ &= \frac{p}{l} - \frac{1}{l}(1 - \frac{d_{min}}{n})\sum_{i=1}^{p} I'_i \end{aligned} \tag{2}$$

where I'_i is the number of interferers of a node's i-th packet at its assigned timeslot. All the links in the network are assigned to l time slots, so the number of interferers are divided by l, i.e. $E(I'_i) = I_i/l$, where I_i is the number of links that a node's i-th packet interferes with, and d_{min} is the minimum Hamming distance of the code.

From (2), we find that a code with larger d_{min}/n is ideal. We are interested in RS code, a popular code, which is maximum distance separable (MDS). RS code does exhibit good performance in our algorithm, though it may not be the best choice in term of good d_{min}/n ratio property.

In this paper a code is denoted as $c(n, k, q)$, where n is the length of the code words, k is the rank of the code, and q is the dimension of the Galois field. A $c(n, k, q)$ code has q^k code words of length n in a Galois field $GF(q)$ [11].

According to the property of RS code, $d_{min} = n - k + 1$. When n and k are fixed and q varies, d_{min} remains the same. There are q^k code words in $GF(q)$. When q increases, the number of code words increases, so that the probability that two links choose the same code word (channel) decreases. As a result, q should be as large as possible. In this paper, q is set to the number of orthogonal channels in the network, i.e. $q = M$. The length of an extended RS code is q, i.e. $n = q = M$. Table I is an example of a $[4, 2, 4]$ extended RS code. We use a channel-time space to describe the scheduling of links. If two links are assigned to one channel-time grid, a collision will occur. Figure 4 shows a CC-TDMA scheduling of links in Figure 1. The links select code words in Table I stochastically, and there is only one collision, namely, between L_2 and L_5 at the $3rd$ time slot in sub-frame 2.

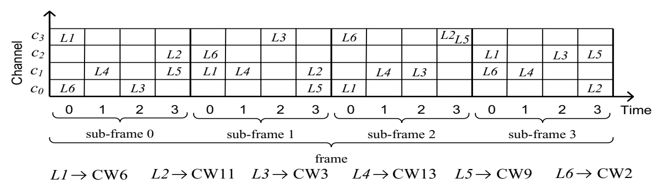


Fig. 4. CC-TDMA scheduling example

C. Analysis

CC-TDMA segregates the packets to l different time slots by μ-**M**, which eliminates the explicit conflicts completely and minimizes the implicit conflicts. Besides, we utilize multiple channels and coding theory to reduce implicit conflicts further.

Note that in Equation (2), G_{min} do not take into account the probability that two links choose the same code word. When k changes from 1 to n, the distance between the code words decreases, i.e. if two links choose different code words in a frame, the number of conflict sub-frames in which the links are assigned the same channel increases. However, the total number of code words increases so that the probability that two links choose the same code word decreases. There is a tradeoff between the number of code words and the number of conflicting sub-frames. According to our simulation, we find that the best value of k is 3.

Now we compare the minimum throughput of CC-TDMA with the existing algorithms.

$$\begin{aligned} E(G_{min}) &= E(\frac{p}{l} - \frac{n - d_{min}}{nl} \sum_{i=1}^{p} I'_i) \\ &= \frac{p}{l} - \frac{k-1}{nl} E(\sum_{i=1}^{p} \frac{I_i}{l}) = \frac{p}{l} - \frac{k-1}{nl}(\frac{p}{l}\bar{I}) \\ &= \frac{p}{l}(1 - \frac{k-1}{Ml}\bar{I}) \end{aligned} \quad (3)$$

The minimum throughput of the algorithm in [10] is:

$$E(G'_{min}) = E(\frac{r - (k-1)I}{r^2}) = \frac{1}{r}(1 - \frac{k-1}{r}\bar{I}) \quad (4)$$

I_{max} is the maximum number of interferers of a link. According to [10], r is the number of slots in each sub-frame and $r = 2D_{max} = \Delta$. As mentioned above, $1.25\Delta \geqslant l \geqslant \Delta$ and $M \geqslant 1$. So $Ml \geqslant r$, and

$$1 - \frac{k-1}{Ml}\bar{I} \geqslant 1 - \frac{k-1}{r}\bar{I} \quad (5)$$

Hence,

$$E(G_{min}) \geqslant p\frac{r}{l}E(G'_{min}) \geqslant \frac{p}{1.25}E(G'_{min}) \quad (6)$$

where $\bar{I}$ is the average value of I. The analysis above verifies that our algorithm performs better than the existing algorithms. When p increases, which implies that the density of the nodes increases, our algorithm will have even better performance.

IV. PERFORMANCE EVALUATION

We simulated the following scenarios for comparison:

Scenario 1: 100 nodes are placed uniformly within a square area whose size varies from 3×3 to 40×40.

Scenario 2: 500 nodes are placed uniformly within a square area whose size varies from 8×8 to 100×100.

Scenario 3: 1000 nodes are placed uniformly within a square area whose size varies from 10×10 to 200×200.

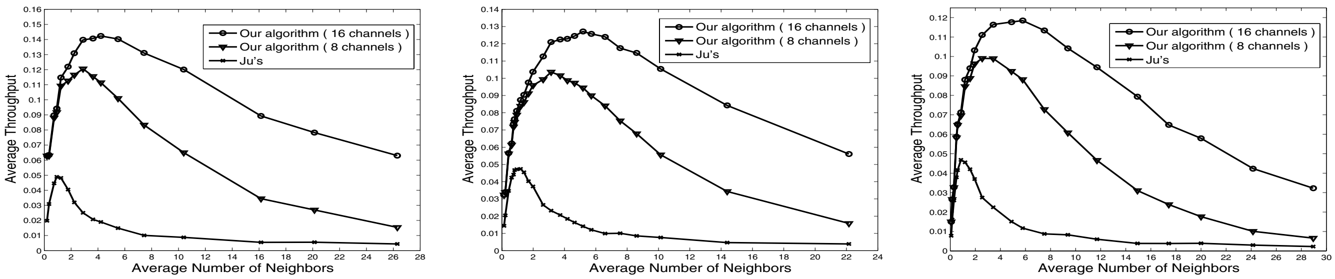
The transmission range of the nodes is 1 and the interference range is 2.5. There exist bidirectional links between any pair of neighboring nodes. The network traffic load is set to 1, i.e. there are always packets on each link waiting to be transmitted. The average number of neighbors $\bar{D}$, which reflects the density of the network, varies with the size of the square area. We simulate 8-channel CC-TDMA, 16-channel CC-TDMA and Ju's algorithm [10], where the bandwidth of each channel is the same. The simulations for all scenarios are implemented in C++ and the simulation results are averaged over 20 runs.

In the simulation, the throughput is defined as the average number of successful packets per node in each time slot, and the average transmission delay is defined as the average waiting time between two successive successful transmissions.

A. Throughput and Delay

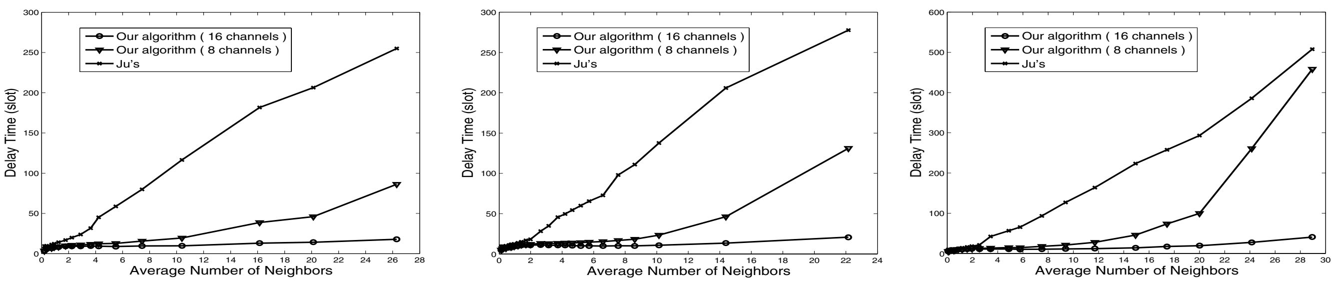
Figure 5 and Figure 6 show the average throughput and delay of CC-TDMA and Ju's algorithm for Scenarios 1, 2, and 3, respectively. When the density of the nodes is small, the total number of packets is so small that the throughput of the whole network is small. As the density of the nodes increases, the number of conflicts increases which affects the network throughput.

From the simulation result, we can see that when $\bar{D}$ is between 5 and 15, the average throughput of 8-channel CC-TDMA can achieve 8-9 times that of Ju's, while the delay is only $1/8$ of Ju's. When $\bar{D}$ is between 10 and 30, the average throughput of 16-channel CC-TDMA can achieve 16-20 times that of Ju's, while the delay is less than $1/16$ of Ju's. This indicates that compared with Ju's algorithm utilizing single channel, CC-TDMA using multiple channels improves the network performance greatly. Ju's algorithm which maximized a lower-bound on the minimum throughput is optimal under the condition that none of the topology information is used at all, while CC-TDMA, whose performance per channel is



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Fig. 5. Performance Comparison with other algorithms (Throughput)



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Fig. 6. Performance Comparison with other algorithms (Delay)

better than that of Ju's in some cases, depends on the topology information of immediate neighborhood.

B. Communication Overhead

The communication overhead of CC-TDMA mainly comes from the messages of tentative color exchanged by the immediate neighbors in μ-**M**. In each μ-**M** round, there are only two messages exchanged between each pair of immediate neighbors, so the number of rounds can be used to measure the communication overhead. Figure 7 shows that μ-**M** colors all the links with no more than five rounds in all scenarios, which implies that CC-TDMA enjoys very low average communication overhead.

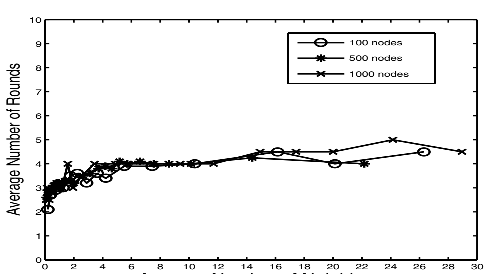


Fig. 7. Average number of rounds

V. Conclusion

In this paper, we propose a new distributed coloring- and coding-based TDMA scheduling that maximizes the throughput of multi-channel wireless ad hoc networks. We evaluated our scheduling algorithm analytically and numerically and showed that it exhibits better performance than existing algorithms.

References

[1] R. Diestel. Graph theory. Springer, New York, 2005.

[2] A. Panconesi and A. Srinivasan, "Improved Distributed Algorithms for Coloring and Network Decomposition Problems," In *Proc. of ACM Symposium on Theory of Computing*, 1992, pp. 581-592.

[3] A. Panconesi and A. Srinivasan, "Randomized Distributed Edge Coloring," *SIAM Journal on Computing*, 26(2):350-368, 1997.

[4] D. A. Grable and A. Panconesi, "Nearly optimal distributed edge colouring in $O(\log \log n)$ rounds," In *Proc. of ACM/SIAM Symposium on Discrete Algorithms*, 1997, pp. 278-285.

[5] S. Gandham, M. Dawande and R. Prakash, "Link scheduling in sensor networks: Distributed edge coloring revisited," In *Proc. of IEEE INFOCOM*, 2005, vol. 4, pp. 2492-2501.

[6] J. Misra and D. Gries, "A Constructive Proof of Vizing's Theorem," *Information Processing Letters*, 41:131-133, 1992.

[7] M. V. Marathe, A. Panconesi and L. D. Risinger JR, "An experimental study of a simple, distributed edge-coloring algorithm," *ACM Journal of Experimental Algorithmics*, vol. 9, No. 1.3, pp. 1-22, 2004.

[8] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio irregularity on wireless sensor networks," In *Proc. of ACM International Conference on Mobile Systems, Applications, and Services*, 2004, pp. 125-138.

[9] 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.

[10] J. H. Ju and V. O. K. Li, "An optimal topology-transparent scheduling method in multi-hop packet radio networks," *IEEE/ACM Transactions on Networking*, vol. 6, No. 3, pp. 298-306, Jun. 1998.

[11] C. H. Rentel and T. Kunz, "MAC coding for QoS guarantees in multi-hop mobile wireless networks," In *Proc. of ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks*, 2005, pp. 39-46.

[12] 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, iss. 8, pp. 1345-1352, Aug. 1999.