A Mobility-Aware Channel Allocation Strategy for Clustered Ad hoc Network

Roni F Shigueta, Mauro Fonseca, Aline Carneiro Viana

HAL Id: hal-01676499

https://inria.hal.science/hal-01676499

Submitted on 5 Jan 2018

A Mobility-Aware Channel Allocation Strategy for Clustered Ad hoc Network

Roni F. Shigueta*, Mauro Fonseca†, Aline Carneiro Viana‡

*PUC-PR - Pontifical Catholic University of Parana, Brazil

†UTFPR - Federal Technological University of Parana, Brazil

‡INRIA, France

Email: *shigueta@yahoo.com, †maurofonseca@utfpr.edu.br, ‡aline.viana@inria.fr

Abstract—This paper presents a mobility-aware channel allocation strategy for clustered ad hoc network. Our main novelty is to consider the user mobility associated with traffic history and node popularity to guide the channel allocation process, while quickly responding to changes in the network topology. In our performance evaluation and contrarily to related works, we use a realistic mobility model based on user behavior and consider channels with largest spectral distance as well as the smallest number of occurrences. Obtained results show that our strategy presents throughput 15,43 % and 17,74% higher when compared with RANDOM and LD algorithms respectively, and lower overhead when compared with TABU algorithm.

Index Terms—Dynamic Spectrum Access, Distributed Channel Allocation, SLAW Mobility Model

I. INTRODUCTION

The growing use of mobile devices and network services have decreased the frequency spectrum availability. This scenario becomes more complicated with the development of news wireless devices for Internet of Things and Machine-to-Machine Communications that share the same frequency spectrum. In addition, the user mobility influences in the network behavior. Users have social behavior and tend to move resulting in the formation of clusters. This movement causes connections/disconnections, requiring that the resource allocation occurs in a manner quick, dynamic, and optimized. In this context, this paper presents a mobility-aware channel allocation strategy for clustered ad hoc network that considers the mobility associated with traffic history and node popularity. In addition, our approach is distributed and does not require complete knowledge of the network graph because we limit the exchange of channel information to 2-hops.

This paper is organized as follows. Section II address related work. Section III presents the system model. Section IV details the channel allocation strategy. Section V presents the scenario and performance evaluation and Section VI concludes the paper.

II. RELATED WORK

In the literature, some authors address the channel allocation in clustered ad hoc networks. The cluster is a set of nodes consisting of a cluster head (responsible for allocating resources) and ordinary nodes (that determine the boundary of the cluster). In [1] the authors present a scheme based on polling where the cluster head is responsible for establishes collision-free intra-cluster communications. For this, each cluster head learns the traffic pattern of its own cluster members and coordinates the transmissions to avoid collisions. This scheme has the limitation of only a single channel per cluster, with its bandwidth shared with the cluster members. In [2] the authors present a meta-heuristic based on Ant Colony Optimization (ACO) in which the network is modeled as a graph where the nodes are the cluster heads and the edges are the links among cluster. In this model, the channels are allocated among clusters and not for links inside the cluster. In [3] the authors propose a scheme in which a cluster head continuously monitors the power level in all channels. If the load on the channel increases beyond the capacity, the cluster head starts using an additional channel with the lowest power level measurement. This scheme allows only to detect the interference in the transmitter, and not in the receptor node. Our work differs from [1],[2],[3] because, in our scheme, the role of the cluster head is assigned dynamically to a node in the cluster according to its priority. The priority of the node is a function of its mobility, traffic, and popularity. In addition, we use a 2-hops interference model that allows to predict the interference in the transmitter and receiver and to allocate channel for the links inside the cluster.

III. SYSTEM MODEL

This section provides details on the models, considered in our approach, namely: network, interference and mobility model.

A. Network Model

The network model is based on an ad hoc network which uses the 802.11b/g spectrum. Each node in the network has two network interfaces: control interface and data interface. The control interface is responsible for exchanges information about available channels, mobility (position, acceleration, and direction), routing, and messages requesting channel switching (RTS/CTS) to a neighbor node. All network nodes are tuned to the same control channel to maintain minimal connectivity among nodes. Data interface sends and receives data packets and switches the channel according to channel allocation strategy.

B. Interference Model

The interference model considered is the 2-hops interference model [4]. This model is used to estimate the interference among channels and only for the channel allocation process. In addition, we use the default interference model of the NS-v3 simulator to avoid the interference in the data transmission process. The 2-hops model is used predictively and allows to select a combination of channels that generates less interference in the 2-hops neighboring nodes. In 2-hops model, two links are considered interfering if they are exactly 2-hops away from each other and use the same channel. We justify the choice of this model by the existence of only one radio interface for data communication. Thus, a node can communicate with only one of its neighbors at given time. We consider the 1-hop interference as contention (managed by the sending of RTS/CTS messages), which is not possible to be eliminated. Thus, it remains to mitigate the interference at 2-hops.

In addition to the 2-hops interference model, we consider the adjacent channel interference [5]. In Wi-Fi networks each channel has a bandwidth of 22 MHz, but they are separated by only 5 MHz as shows Figure 1.
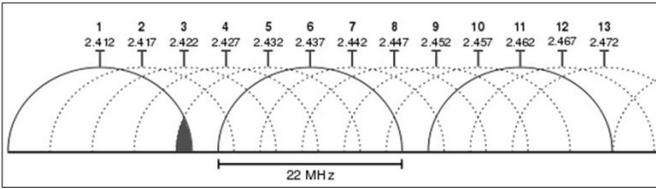

Fig. 1. Channels overlapping in Wi-Fi spectrum

The low spectral distance causes overlapping and consequently interference among adjacent channels. The band overlap factor is proportional to channels distances [6]. For example, channels 1 and 2 have high overlap factor. Channels 1 and 5 have low overlap factor (as showed in grayscale in Figure 1). For a better use of channels, we use a channels combination in the links that result in a lower overlap factor.

C. Mobility Model

The node mobility in the network is determined by a mobility model called SLAW (Similar Least-Action Human Walk) [7]. This model is based on real GPS traces collected from five outdoor sites (two campuses, a metro, Disney World and a state fair scenario). This model considers some fundamental statistical properties of human mobility: 1) straight line trips without directional change or pause have a truncated power-law distribution; 2) people mostly move only within their own confined areas of mobility and different people have different mobility areas; 3) times elapsed between two successive contacts of the same person can be modeled by a truncated power law distribution; 4) way-points of humans can be modeled by fractal points (people are more attracted to more popular places); 5) people more likely visit destinations nearer to their current waypoint when visiting multiple destinations in succession. In this model, every mobile node chooses a set of clusters and a fraction of waypoints to visit from each of the selected clusters. The node visits all of the selected waypoints, pausing in every waypoint. After traveling all the waypoints, the node randomly selects another cluster and a new subset of waypoints, starting another trip.

IV. CHANNEL ALLOCATION STRATEGY

The channel allocation strategy exchanges information about available channels and mobility among nodes using `Hello` messages. This information is used by the channel allocation algorithm to propose a channel for each link of the node. The proposed channels are then sent to neighboring nodes using `Interaction` messages. A neighbor node upon receiving an `Interaction` message evaluates if it accepts or rejects the proposed channels according to the message priority. The following subsections detail the channel allocation strategy.

A. Hello Messages

`Hello` messages are sent periodically to notify neighboring nodes about the available channels and mobility information (position, speed, and direction) of the node. Upon reception of a `Hello` message, a neighboring node obtains the available channels list and calculates the common channels list. This list results of the intersection between the received available channel list of a neighbor node and the local channels list. The common channels list contains the channels set that are used by the channel allocation mechanism to select a candidate channel for data interface. The mobility information is used to calculate the relative position among nodes and to estimate their future positions, as discussed in the next subsection.

B. Channel Allocation Algorithm

The channel allocation algorithm is executed whenever the node is initialized and when the topology or channel changes. Each node sorts the channels (in a channels list) according to their quality. The channel quality is determined by selecting channels that generate less than 2-hops interference. The 2-hops model is used due to its simplicity and due to the possibility of estimating which channel will be allocated to the link. Our algorithm prioritizes not used channels with largest spectral distance or interfering channels with a smallest number of occurrences (channels less used). After channels sorting, the links are evaluated and ranked according to the behavior (mobility, recent traffic history, and node popularity) of the neighbor node which belongs to the current link. For this, we compute a link priority function $p(t+\Delta)$ that uses link's behavior parameters (represented by $b(t+\Delta)$) in a future time window Δ. The priority function for the k-th link is:

$$p_k(t+\Delta) = \begin{cases} 0 & , \quad \text{if} \quad |d_k(t+\Delta)| > r_m, \\ 1 - b_k(t+\Delta), & \text{if} \quad |d_k(t+\Delta)| \le r_m. \end{cases} \quad (1)$$

Where $d_k(t+\Delta)$ is the Euclidean distance between link's nodes in the future time window Δ, r_m is the maximum transmission radius of the node, and $b_k(t+\Delta)$ is the behavior parameter of link.

When the distance between link's nodes is smaller than the maximum transmission radius, the node is considered to be inside the neighbor coverage radius. Otherwise, the node is outside and receives zero priority.

The behavior function $b(t+\Delta)$ for the k-th link is given by:

$$b_k(t + \Delta) = \alpha * M_k(t + \Delta) + \beta * T_k(t) + \gamma * D_k(t) \quad (2)$$

Where α, β, γ are weights applied in each of the parameters and depend on the network scenario.

The $M_k(t+\Delta)$ parameter represents the node displacement at the future time window Δ, $T_k(t)$ and $D_k(t)$ correspond respectively the traffic and the node degree (neighbor node to the local node and belonging to the k-th link). The node degree represents the node popularity.

After calculating the links priorities, the node sorts the links in a link list (according to the highest priority). Each link of this list is then associated with a channel in the channel list in their respective order. In this way, the highest quality channels will be proposed to the links of highest priority.

In addition to performing the allocation proposal to each of its links, a node calculates its priority. The node priority is used to decide if the proposed allocation sent by a node should be accepted or not by its neighbor node. The node priority is represented by the average value of the links priorities of each node.

The channel allocation proposal and the node priority are sent to neighboring nodes via `Interaction` messages.

C. Interaction Messages

`Interaction` messages are sent by a node that calculated a new channel allocation to inform the neighbor's nodes the new proposal of allocation. When a node receives an `Interaction` message, it first compares the priority $P(t + \Delta)$ of the neighbor node that sent the message with its local priority. If the received message priority is smaller than the local priority, the message is discarded. Otherwise, the node accepts the link assignment proposed by the neighbor node and recalculates the channel allocation for its other links (searching for a channel not used with largest spectral distance or, in the case that all channels are used, searching for a channel with the smallest number of allocation occurrences).

Algorithm 1 describes the channel allocation procedure in lower priority nodes.

In Algorithm 1, for each ordered link in L (line 1), ordered according to eq. 1, if the link is the link that received the `Interaction` message (line 3), it accepts the channel proposed by the `Interaction` message (line 4). Otherwise (line 5), the node searches a new channel (not allocated) that is different from the channels contained in the `Interaction` message (i.e., different channels to 2-hops) and with largest spectral distance (lines 6-8). If the channel is non-interfering (line 9), the channel is assigned to the link (lines 10-11). If there is no non-interfering channel, the link is classified as interferent and stored in *InterferentList* (line 14). For each *link* of the *InterferentList*, the algorithm selects a channel that has the least number of occurrences in the list of channels

---

Algorithm 1 Channel Allocation to Lower Priority Nodes

Input: $LinkList$, $\qquad$ $InterferentLinkList$, $InterferentChannelList$, C

Output: $AssignedLinkList$, $AssignedChannelList$
$\quad L \leftarrow$ ordered links of $LinkList$
$\quad InterferentLinkList \leftarrow$ interfering links contained in the interaction message
$\quad InterferentChannelList \leftarrow$ interfering channels contained in the interaction message

1: **for** each $element \in L$ **do**
2: $\quad link \leftarrow element$
3: $\quad$ **if** $link$ is the link that received the interaction message **then**
4: $\quad\quad ch \leftarrow$ channel proposed by interaction message
5: $\quad$ **else**
6: $\quad\quad$ **for** each $element \in C$ **do**
7: $\quad\quad\quad ch \leftarrow$ channel non-interfering with $InterferentChannelList$
8: $\quad\quad$ **end for**
9: $\quad\quad$ **if** ch is non-interfering **then**
10: $\quad\quad\quad AssignedLinkList \leftarrow link$
11: $\quad\quad\quad AssignedChannelList \leftarrow ch$
12: $\quad\quad\quad continue$
13: $\quad\quad$ **else**
14: $\quad\quad\quad InterferentList \leftarrow link$
15: $\quad\quad$ **end if**
16: $\quad$ **end if**
17: **end for**
18: **for** each $link \in InterferentList$ **do**
19: $\quad ch \leftarrow$ channel with the lowest number of occurrences in $InterferentChannelList$
20: $\quad AssignedLinkList \leftarrow link$
21: $\quad AssignedChannelList \leftarrow ch$
22: **end for**

---

contained in the `Interaction` message (lines 18-22). A smaller number of occurrences results in a lower level of interference. In our work the level of interference among links is estimated by a channel quality metric (CQ) based on the number of 2-hops neighbors that use the same channel. This metric (adapted for 2-hops) is given by [8]:

$$CQ = 1/(1 + I_{i,c}), \quad (3)$$

where $I_{i,c}$ is the number of 2-hops neighbors from node i that use the channel c.

V. PERFORMANCE EVALUATION

This section describes the simulation experiments we have conducted using NS-v3 simulator [9] in order to assess the performance of our approach. The goal of our evaluation is to show how the design of our distributed mobility-aware channel allocation strategy enables flexibility in environments with varying network topologies due to mobility. For this, we have performed the following analysis.
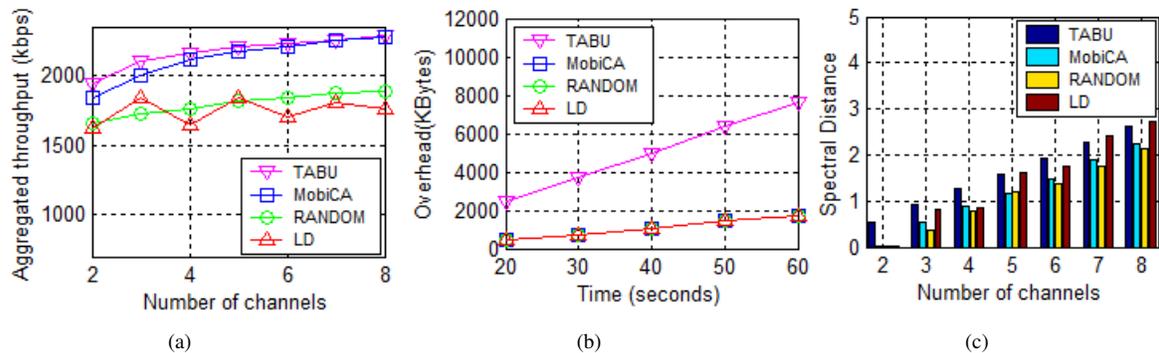
Fig. 2. (a) Aggregated throughput, (b) overhead, (c) spectral distance.

a) **Simulation setup::** The evaluation scenario corresponds to a simulated network consisting of 40 nodes distributed in an area of $800m^2$ and having a transmission range of 100m. The nodes distribution and mobility are given by the SLAW mobility model [7]. The SLAW mobility model divides the area into clusters and in each cluster, sets the way nodes will move inside the cluster.

In order to investigate the interference among links after the channel allocation is performed, we randomly select 5 links in the most populated cluster given by the SLAW mobility model. We then simultaneously generate traffic flows at each 20s. Each flow transmits 1000 data packets and each packet has a size of 1KBytes. We run 1000 simulations and show results in terms of mean values and confidence intervals, at a 95% confidence level (which is less than 2% around the mean).

b) **Compared approaches::** We compared our proposal, named here MobiCA (Mobility-aware Channel Allocation) with three other methods: centralized channel allocation (TABU) [10], randomly channel allocation (RANDOM), and largest distance (LD) among channels [5]. In TABU, the allocation is realized by a central unity in a heuristic manner. TABU algorithm is considered to provide the upper-performance limit. In RANDOM algorithm, one channel is randomly chosen for each link. RANDOM is considered to provide the lower performance limits. In LD algorithm, the criterion for channel allocation is based on the largest spectral distance among a candidate channel for the allocation and the channels allocated to the 2-hops neighbors. In LD, candidate channels with largest spectral distance have higher priority in the channel allocation process.

c) **Performance metrics::** To evaluate the approaches efficiency, we use four metrics: (1) aggregated throughput, which is the sum of packets delivered to all network nodes per unit of time, (2) spectral distance, which is the distance in the spectrum among an allocated channel and the others channels allocated in the 2-hops neighbors, (3) packet delivery rate, which is the number of packets received in relation to the number of packets sent, and (4) end-to-end delay, which is the time interval that a packet takes to reach its destination.

d) **Simulation results::** Figure 2(a) illustrates the aggregated throughput according to the number of channels. MobiCA algorithm presents a throughput about 10.14% and 11.58% higher (considering 2 channels) when compared to RANDOM and LD algorithms, respectively. When compared with TABU algorithm, MobiCA presents throughput 5,41% lower. As the number of channels increases, MobiCA maintains for all channels, a higher average aggregated throughput when compared to RANDOM (15.43% higher) and LD algorithms (17.74% higher). The better performance of MobiCA is explained by the fact that its channel assignment mechanism prioritizes links with closer neighbors, with higher traffic, and higher node degree. TABU algorithm presents a higher throughput (for all channels, about an average of 2.30% compared to MobiCA). TABU has the highest throughput due to its centralized channel allocation mechanism. Nevertheless, TABU search algorithm has the limitation of high overhead of messages in the network.

Figure 2(b) shows the overhead in the control channel varying with the time (considering 6 channels). We analyze the overhead in bytes (and not in packets) because there are several types of control messages (routing, RTS/CTS, `Hello`) with different sizes. TABU presents the higher overhead (an average of 5 MB in the 40 seconds interval) due to a number of hops for a node to find the central unit to be greater. MobiCA, RANDOM and, LD present the same overhead (an average of 1.1 MB in the 40 seconds interval), because they use the same mechanism for exchanging messages (only changes the channel number) and their overhead is lower because the exchange of messages is limited to 2-hops. We can observe in Figure 2(b) that the increase in overhead is much higher in TABU solution.

Figure 2(c) shows the comparison among TABU, MobiCA, RANDOM, and LD algorithms considering the spectral distance. In average, TABU algorithm presented the highest spectral distance (1,59), followed by LD (1,45), MobiCA (1,17), and RANDOM (1,08), respectively. Although, the LD algorithm uses as decision criterion the highest spectral distance, its simulation demonstrated that there is no throughput gain (Figure 2(a)) when compared with a hybrid criterion (i.e., selecting a channel with largest spectral distance if it was not
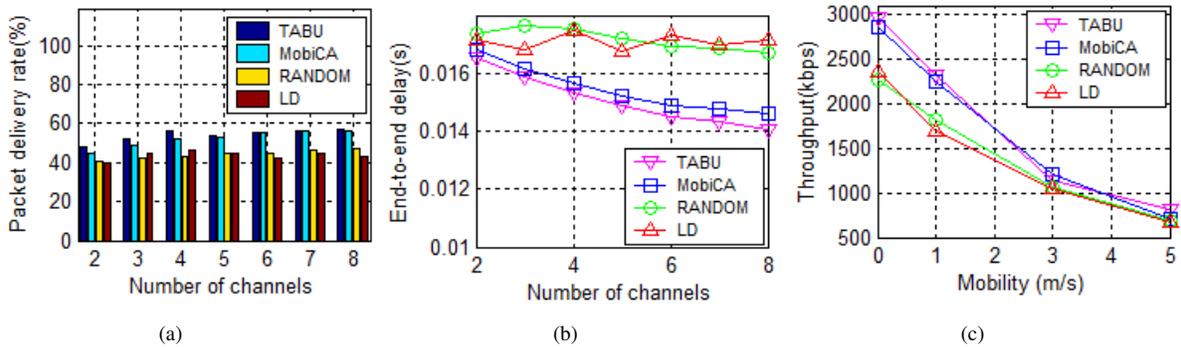
Fig. 3. (a) Packet delivery rate, (b) end-to-end delay, (c) mobility.

allocated to a 2-hop neighbor, or selecting the channel with the smallest number of occurrences if all channels are allocated to a 2-hop neighbor), as used in MobiCA algorithm.

Thus, a larger spectral distance does not guarantee a throughput gain because it also depends on the number of times that the channel with larger spectral distance is allocated.

Figure 3(a) shows the average packet delivery rate. TABU algorithm presented the highest packet delivery rate corresponding to 54%, while MobiCA, RANDOM, and LD algorithms presented 52.28%, 44.14%, and 42.85%, respectively.

Figure 3(b) illustrates the average end-to-end delay. For 3 channels, RANDOM had the highest delay among all algorithms (17.66 ms). With the increasing of the number of channels, end-to-end delay values between RANDOM and LD become close (an average difference of 0.63% for all channels). This is due to a greater number of available channels which increases the spectral distance and decreases the number of occurrences of interferences. The lower average end-to-end delay for all channels (considering 8 channels) were presented by TABU (15.16 ms), and MobiCA (15.35 ms) algorithms, respectively. For few channels, the end-to-end delay value is higher due to the contention and higher waiting time in the transmission buffer. As the number of channels increases, the number of contention is decreased and consequently, is so the packet delivery delay.

Figure 3(c) shows the throughput variation in function of the mobility. The higher throughput for all algorithms is obtained in a static position (speed = 0 m/s). TABU and MobiCA present higher values (2968 kbps and 2857 kbps, respectively) while LD and RANDOM algorithms, values lower (2363 kbps, 2247 kbps, respectively). As the nodes move, the clusters become more sparse and for low speed (1 m/s), TABU and MobiCA present higher values (2300 kbps and 2247.27 kbps, respectively) than RANDOM and LD algorithms (1822 kbps, 1692 kbps, respectively). For higher speed, the node disconnection frequency increases, and the network throughput decrease rapidly.

VI. CONCLUSIONS

In this paper, we propose a mobility-aware channel allocation strategy for clustered ad hoc network. The main contribution of this work is to consider characteristics such as mobility, traffic history, and node popularity in the decision making of the channel allocation mechanism. In this context, the SLAW mobility model is used to generate realistic mobile simulation scenarios in our evaluation. When compared to two other approaches, named RANDOM and LD, results show that our method (MobiCA) presents greater values for aggregated throughput and packet delivery rate and lower end-to-end delay. Although MobiCA has lower performance than TABU algorithm, its distributed approach allows quickly respond to network topology changes due the exchange messages be limited to 2-hops), unlike TABU algorithm that uses a centralized approach and needs information about the entire network, resulting in higher overhead.

REFERENCES

[1] S. S. Dehkordi and J. Torkestani, "Channel assignment scheme based on learning automata for clustered wireless ad-hoc networks," in *Journal of Computer Science and Infomation Technology*. JCSIT, June 2014, pp. 149–172.

[2] M. Parsapoor and U. Bilstrupand, "Ant colony optimization for channel assignment problem in a clustered mobile ad hoc network," in *Proceedings of the 4th International Conference on Swarm Intelligence*. ICSI, June 2013, pp. 314–322.

[3] B. Karaoglu and W. Heinzelman, "Cooperative load balancing and dynamic channel allocation for cluster-based mobile ad hoc network," in *IEEE Transactions on Mobile Computing*. IEEE TMC, May 2015, pp. 951–963.

[4] A. Subramanian, H. Gupta, and R. Das, "Minimum interference channel assignment in multiradio wireless mesh networks," in *IEEE Transactions on Mobile Computing*. IEEE TMC, December 2008, pp. 1459–1473.

[5] E. G. Villegas, E. L. Aguilera, R. Vidal, and J. Paradells, "Effect of adjacent-channel interference in ieee 802.11 wlans," *International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, 2007.

[6] A. Mishra, E. Rozner, S. Banerjee, and W. Arbaugh, "Exploiting partially overlapping channels in wireless networks: turning a peril into an advantage ," in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, 2005, pp. 311–316.

[7] K. Lee, S. Hong, S. Kim, I. Rhee, and S.Chong, "Slaw:self-similar least-action human walks," *IEEE/ACM Transactions on Networking*, vol. 20, pp. 515–529, April 2012.

[8] Y. Saleem, A. Bashir, and E. Ahmed, "Spectrum-aware dynamic channel assignment in cognitive radio networks," in *International Conference on Emerging Technologies*. ICET, October 2012, pp. 1–6.

[9] T. R. Henderson, M. Lacage, and G. F. Riley, "Network simulations with the ns-3 simulator," *ACM SIGCOMM'08*, 2008.

[10] A. Subramanian, H. Gupta, and R. Das, "Minimum interference channel assignment in multiradio wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 1459–1473, December 2008.