A Cross-Layer Architecture to Exploit Multi-Channel Diversity with a Single Transceiver

Jay A. Patel, Haiyun Luo, and Indranil Gupta
Department of Computer Science, University of Illinois at Urbana-Champaign
201 N. Goodwin Ave, Urbana, IL 61801
E-mail: {jaypatel, haiyun, indy}@cs.uiuc.edu

***Abstract*—The design of multi-channel multi-hop wireless mesh networks is centered around the way nodes synchronize when they need to communicate. However, existing designs are confined to the MAC layer – they are based on either negotiation on a rendezvous control channel, or on optimistic synchronization. Both approaches scale poorly as the network grows in coverage and density. The rendezvous control channel may become the bottleneck, while optimistic synchronization may incur substantial overhead – especially amongst nodes close to a gateway, where the mesh traffic converges. In this paper, we describe Dominion – a cross-layer architecture that includes both medium access control and routing. At the MAC layer, a node switches channels in a *deterministic* manner to address the scalability issue. At the network layer, a Dominion node routes traffic along the shortest distance across both *spatial and frequency domains*, based on the deterministic channel-hopping schedule and network connectivity. Since the shortest distance path across the frequency domain is time variant, Dominion naturally spreads the packets of the same flow across multiple paths, relieving the intra-flow and inter-flow contention, and improving throughput. Through QualNet simulations we show that Dominion is able to achieve, on average, 1813% higher aggregate distance-normalized throughput than IEEE 802.11, while being 1730% fairer (using Jain's fairness index) with 50 simultaneous random flows.**

I. Introduction

Wireless mesh networks [1], [2], [3] are emerging as a "*killer app*" for the traditional self-organized wireless ad hoc networks. However, wireless mesh networks have a scalability problem. Network flows are prone to intra-flow and inter-flow interferences, limiting their geographic reach. This problem is further exaggerated as most mesh networks feature only a small number of *gateway* nodes, where traffic to and from the Internet converges. The interference substantially reduces the actualized network capacity, and hence, most deployments do not span greater than a few hops.

One method to scale a community wireless mesh network is to exploit the off-the-shelf IEEE 802.11 (hereon, simply 802.11) transceivers' capability to hop among several orthogonal wireless channels. Multichannel MAC (MMAC) [4] and Slotted Seeded Channel Hopping (SSCH) [5] are two protocols that exploit multi-channel diversity using a single transceiver. (Alternate proposals are discussed in Section V). Based on the 802.11 Power Saving Mechanism (PSM), MMAC nodes periodically converge on a rendezvous control channel to exchange meta data about channel synchronization. MMAC scales poorly as the number of channels increase due to the need for all nodes to converge on a control channel.

SSCH uses randomized channel hopping and optimistic synchronization, therefore eliminating the need of a pre-defined control channel. In SSCH, a node switches its channel every time slot using a cyclical *schedule*, which guarantees that each node converges with its neighbors once every schedule cycle. Nodes learn (or update) the schedule of their neighbors during the periodic convergence as each node locally broadcasts its schedule every time slot. To maintain network flows, a node synchronizes its schedule with its neighbor so that they stay on the same channels – and exchange packets continuously in the following time slots. To communicate with multiple neighbors simultaneously, SSCH permits partial synchronization by empirically hopping across 4 different schedules. When the network density and traffic load are high, e.g., areas close to the gateways, optimistic synchronization with more than 4 neighbors becomes increasingly difficult – due to continuous synchronization and de-synchronization. A node may synchronize with a neighbor, only to find that the neighbor has deviated away from that schedule when the neighbor has its own local traffic to satisfy.

In this paper we propose Dominion[1], a cross-layer architecture that spans both the MAC and the network layers to scale a wireless mesh network to high density and large coverage. At the MAC layer, Dominion divides the network into distinct logical subnetworks, and assigns each subnetwork a static and deterministic channel hopping schedule from which the member nodes do not deviate. Hence, Dominion avoids the bottleneck rendezvous control channel and the contention of schedule synchronization. Dominion routing leverages the knowledge of channel-hopping schedule, and routes traffic along the shortest distance through both the spatial (intra-subnetwork) and frequency (inter-subnetwork) domains. Due to the channel hopping schedule, the shortest frequency domain path between a source and a destination node is time variant, Dominion routing spreads the packets of the same flow across multiple paths – naturally reducing the intra-flow and inter-flow interference.

There are three primary contributions of this paper. First, we identify the scalability problem of existing multi-channel wireless mesh network designs, and propose a new cross-layer architecture to scale a wireless mesh network. The MAC algorithm makes full usage of all available channels, and

[1]The name implies complete 'control' over the channel spectrum.

eliminates the need for maintaining schedules between neighboring nodes (i.e., no need to broadcast schedules). Second, we propose the first graph theoretic model for the multi-channel wireless mesh network, that characterizes not only spatial but also frequency domain connectivity. The Dominion routing algorithm is defined by our graph model, and is engineered towards optimizing throughput. Finally, we have implemented the Dominion architecture in the QualNet [6] simulator. Our simulations show that, on average, Dominion (with 11 channels), when handling 50 simultaneous random flows, improves the normalized throughput by 1813% and 86%, along with fairness by 1730% and 315%, over 802.11 and SSCH respectively.

The rest of the paper is organized as follows: the scheduling algorithm for deterministic channel hopping is described in Section II. Meanwhile, the routing algorithm is discussed in Section III. The experimental evaluation is carried out in Section IV. The related work is reviewed in Section V.

II. SCHEDULING

At the MAC layer, Dominion divides the network into distinct logical subnetworks, and assigns each subnetwork a static and deterministic channel hopping schedule. The key is to allow all pairwise subnetworks to converge on a periodic basis – allowing a node to communicate with all of its neighbors. Once a set of nodes converge on a given channel, the channel is still accessed via CSMA, akin to 802.11. Dominion relies on a globally-known subnetwork assignment scheme so that a node can locally determine any other node's channel hopping schedule. In this section, we present a channel hopping (scheduling) algorithm, fully generalizable to an arbitrary number of channels. Lastly, we mention how nodes are assigned to the various subnetworks.

For the preliminary schedule, with k available channels, we group the nodes into k subnetworks (labeled s_0 through s_{k-1}). We adapt the modulo progression scheme of SSCH [5] to facilitate guaranteed subnetwork convergence. Given k available channels (i.e., channels 0 through $k-1$), the following function c is used to determine the schedule for subnetwork s_i during time slot t:

$$c(s_i,t) = \begin{cases} 0, & \text{if } i=0,\ t=0; \\ c(s_0,t), & \text{if } i>0,\ t=i-1; \\ [c(s_i,(t-1) \bmod k)+i] \bmod k, & \text{otherwise} \end{cases} \tag{1}$$

The first subnetwork (s_0) remains on channel 0 at $t=0$.[2] Every other subnetwork s_i converges with the first subnetwork at $t=i-1$. The rest of the schedule (for other values of t) is calculated using modulo arithmetic with the subnetwork identifier as the seed. This implies that subnetwork s_0 remains on channel 0 for the entire schedule cycle. For all other subnetworks s_i, the channel schedule is calculated based on its channel selection during the previous time slot. When k is a prime:

- The *schedule cycle* repeats every k time slots.
- Each of the k subnetworks converges with every other $k-1$ subnetwork exactly once during the schedule cycle (i.e., every k time slots). This implies that each subnetwork spends one time slot (per schedule cycle) in *solitude* – i.e., not converging with a foreign subnetwork.

This schedule does have a drawback: since each subnetwork is always converging with another subnetwork (except for the one time slot spent in solitude), only $(k+1)/2$ channels are "occupied" during any given time slot. However, if the preliminary schedule cycle is generated with $2k-1$ channels (and assuming $2k-1$ is prime), all k channels can be utilized simultaneously. While this schedule uses $2k-1$ total channels, only $(2k-1+1)/2 = k$ channels are occupied during any given time slot. Thus the number of total channels can be reduced to k by eliminating all channels greater than or equal to k, and replacing them with unutilized channels less than k. For each time slot t, starting with s_0 (and downwards to s_{2k-2}), we can replace the values of $c(s_i,t)$ and $c(s_j,t)$ with a value from $[0,k-2]$ (in sequential order) if $c(s_i,t)=c(s_j,t)$ (i.e., if the subnetworks converge). As described before, a single subnetwork observing solitude will remain – this subnetwork is assigned to channel $k-1$. Assuming a uniform distribution of nodes amongst the subnetworks, only half as many nodes will be switched to channel $k-1$ as any other channel as every other channel will have exactly two converging subnetworks. To mitigate this slight discrepancy in the nodes-to-channels distribution, we "add" a new subnetwork (i.e., s_{2k-1}) that converges with this *solitudnal subnetwork*. Adding a new subnetwork improves the probability of nodes in the previously solitudnal network of having a currently convergent neighbor (i.e., a neighbor belonging to the new subnetwork s_{2k-1} or its own subnetwork) to $1/k$ from $1/(2k-1)$.

The schedule can be generalized by using $P(2k-1)$ as the number of channels, where $P(x)$ is the smallest prime number greater than or equal to x. Subnetworks higher than s_{2k-1} are discarded. Next, the number of channels is reduced to k. However, when $2k-1$ is not prime, in addition to the one solitudnal network, there may be a few additional (up to $P(2k-1)-2k-1$) pseudo-solitudnal networks (i.e., these subnetworks converge with the now discarded subnetworks). These remaining subnetworks can be converged on the remaining channels (a pair at a time), simply forcing some subnetworks to converge multiple times during a schedule cycle.

In summary, the deterministic schedule requires that the network be divided into $S=2k$ distinct subnetworks. Furthermore, it provides a compact schedule cycle of $T=P(2k-1)$, where $P(x)$ is the smallest prime greater than or equal to x. Since the schedule repeats every T time slots, we use the notation t_j to indicate all time slots $t \equiv j \pmod{T}$. As an example, we present a schedule in Table I for a network with 4 available channels. In this example, the schedule cycle duration T is 7 time slots. Each of the 8 subnetworks converge with the other 7 subnetworks during the schedule cycle.

[2] Channels are 0-offset.

TABLE I

SCHEDULE FOR A 4 CHANNEL NETWORK. DURING TIME SLOT t, SUBNET s_i SWITCHES TO CHANNEL $c'(s_i, t_j)$, WHERE $t \equiv j \pmod 7$.

$c'(s_i, t_j)$	t_0	t_1	t_2	t_3	t_4	t_5	t_6
s_0	0	0	0	0	0	0	3
s_1	0	3	1	1	1	1	0
s_2	1	0	1	3	2	2	1
s_3	2	1	0	1	2	3	2
s_4	3	2	2	0	1	2	2
s_5	2	2	3	2	0	1	1
s_6	1	1	2	2	3	0	0
s_7	3	3	3	3	3	3	3

An ideal subnetwork designation assigns an equal number of nodes to each of the S subnetworks either randomly or based on spatial reuse. A one-way uniform hashing function (for example, the SHA-1) may be used to determine a node's home subnetwork. Stated differently, a node A's home subnetwork may be determined by simply using the following mathematical operation: $\texttt{SHA1}(\texttt{MacAddress}(A)) \bmod S$. Another way is based on spatial reuse: using a globally optimal assignment [7] (possible for synthetic networks) or locally based on two-hop neighborhood [8]. However, the subnetwork assignment must be disseminated throughout the network.

III. ROUTING

Utilizing a general purpose routing strategy such as the HOP metric (i.e., fewest number of hops) or the ETX [9] metric with the deterministic MAC scheduler may yield undesirably high latencies or suboptimal throughput. An example is presented below. In this section, we present an abstract graph model of the physical topology, and in conjunction, present a link-state routing algorithm to select multiple routes that maximize the end-to-end throughput.

A. The Case for Cross-Layer Architecture

Using a 4 channel network (see Table I), in Fig. 1 we illustrate a sample scenario where simply using the shortest path would not maximize the available throughput. In the aforementioned scenario, Node A (belonging to s_3) sends packets to a neighboring node B (in s_4) starting at t_0. If node A relays the packets exclusively using the shortest route to node B (direct packet transmission at t_6), node A will not utilize the maximal opportunity to increase its end-to-end throughput. By selecting a second path, that relays packets via node C at t_0, node A can double its end-to-end throughput. An important observation: due to the constant channel hopping, multiple routes can be simultaneously used to exploit capacity gain.

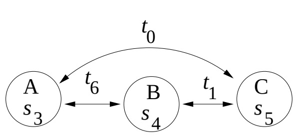


Fig. 1. **A longer route may minimize the end-to-end latency.**

B. Generalized Algorithm

Abstract Network Model: Each physical node A is replaced with $T + 1$ *virtual nodes* in the abstract model. The first T virtual nodes, labeled A^0 through A^{T-1}, are the *temporal nodes*. Each temporal node represents the physical node during each of the T distinct time slots. *Temporal edges* represent the passage of one time slot and cyclically connect the temporal nodes, i.e., there is a one-way temporal edge from each node A^t to $A^{(t+1) \bmod T}$ for all $t < T$. As an example, in Fig. 2, we expand node A (in a 4 channel network) from Fig. 1. The temporal nodes of node A, A^0 through A^6, are cyclically connected.

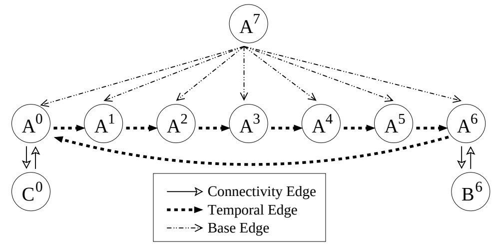


Fig. 2. **Expansion of a physical node in the abstract graph model.**

For each link in the physical topology, we add a virtual *connectivity edge* from node A^t to B^t for all t during which the physical nodes converge. In the illustrated example, the physical links from node A to nodes B and C are translated to connectivity edges from A^6 to B^6 and A^0 to C^0, respectively. The final virtual node, labeled A^T, represents the physical *base node* itself. The base nodes are connected to their respective temporal nodes via *base edges* – these edges represent the starting (or terminating) state of network flows. To reduce the number of edges in the abstract graph – only the source and destination node's base nodes are connected to their temporal nodes (as finding routes to other base nodes is unnecessary).

Assigning Edge Weights: Each temporal edge is given a weight of 0 (i.e., no transmissions taking place), whereas each connectivity edge is weighted according to its ETX [9] cost (i.e., ETX is the number of expected transmissions per packet). By giving a weight of 0 to temporal edges, it may seem that we are simply reducing the abstract graph to the normal network topology, but this is not the case – the abstract model allows us to select multiple routes.

Computation: Djikstra's algorithm is used to find the shortest path between the base nodes of the source and the destination in the abstract graph. This path represents the shortest route from (starting with any time slot) the source node A to destination node B. Once a route is located, all the virtual edges utilized by the path are pruned from the network. Djikstra's algorithm is run repeatedly to locate multiple paths, until no more routes can be found.

Maintaining Link-state: The algorithm described above is a centralized algorithm. The routing protocol is made distributed by maintaining a weakly consistent link-state at each node. Initially the nodes are bootstrapped with a set of high quality links. Whenever a link's "quality" changes substantially – for example, due to degradation in channel quality – the information is disseminated to the network. While the cost of propagating link changes is high, we believe that such situations can be minimized by maintaining only high quality links and by relaying information on demand, i.e., only when an actively used link changes. Each source node constructs the virtual graph based on its local link-state.

IV. Experiments

We evaluate Dominion against stock 802.11 and SSCH [5] using the Qualnet v3.9 network simulator [6]. The experiments evaluate the effect of random flows in a wireless mesh network, where a node may serve as either a source node or destination node of any number of flows. We find that Dominion outperforms both 802.11 (by an order of magnitude) and SSCH in aggregate normalized throughput and fairness.

Implementation: We implement Dominion using the QualNet v3.9 simulator [6]. Briefly, the implementation (i) uses source routing that describes each hop, i.e., when nodes should forward the packet, (ii) implements per-flow, per-time slot queuing to prevent head of line blocking, (iii) provokes a warning mechanism to prevent buffer overflows at intermediate nodes, (iv) attempts 1 DCF (vs. 7 for 802.11) transmission at a time for each packet – this reduces switching delays – and finally, (v), a packet is dropped after 14 successive DCF failures, akin to 2 802.11 retries.

We acquired the code for SSCH from its authors. The original SSCH code was implemented in Qualnet v3.6, which we ported to v3.9. SSCH has an intrinsic problem. It requires the number of channels to be prime. The suggested workaround is to use a large prime as the number of available channels, and then apply modulo reduction to map down to the actual number of available channels. However, currently SSCH does not implement this workaround. Thus, to make for a fairer comparison, we use 11 as the number of channels for all comparative experiments.

Methodology: The base topology is generated using uniform random placement, with 100 nodes placed in a square with 1000 m edges. The node radios transmit at 28 dBm using the 802.11 MAC at 54 mbits/s. Using the simulator, the maximum shortest distance between any two nodes is 7 hops. RTS-CTS is enabled. The radio interfaces are configured to utilize up to 12 channels, the number specified by 802.11a. The time slot for both Dominion and SSCH is set to 10 ms. The cost of a channel switch is 80μs [5], [10], therefore, the overhead of continuous channel switching is approximately 0.8%. Lastly, for Dominion, nodes were assigned to subnetworks randomly.

Our experiments were performed using constant bit rate (CBR) flows over UDP. A 1024-byte packet was offered every 100μs, safely saturating the maximum MAC bit rate. Each individual simulation run was executed for 30 seconds, with network flow(s) starting at the 15 second mark and terminating at the end. To reduce statistical anomalies, each comparative benchmark was re-evaluated 5 times using a different set of flows. All results are described using the average of the experimental runs.

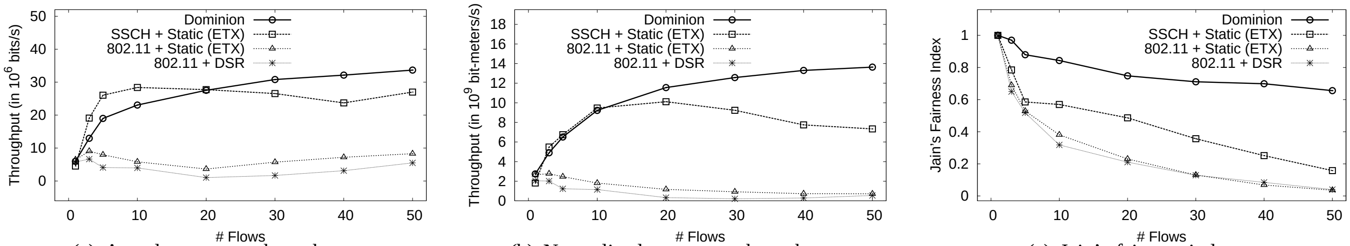
A bootstrap process was executed to gauge the quality of each link – as QualNet's physical model intrinsically supports variable link quality. During the process, each node broadcasts a few hundred HELLO message at random intervals. At the completion of the process, the quality of each link is calculated based on the proportion of received HELLO messages. With the link quality information, we calculate static routes for 802.11 and SSCH using ETX [9]. Static ETX routing eliminates route discovery and maintenance overhead. Since static routing is oblivious to a change in channel quality, we do not disseminate link change information with Dominion – hence, making the link-state "static". However, an initial set of high quality links (i.e., forward transmission success rate higher that 85%) was provided to all nodes. Using high quality links for Dominion may seem unfair, however, ETX naturally prefers high quality links. In our experiments, 96+% of links selected by ETX routes were of high quality (i.e., over 85%).

Results: Fig. 3 shows that both Dominion and SSCH achieve an order of magnitude higher aggregate throughput than 802.11 with random flows. The throughput increases substantially as the numbers of flows increase. While both Dominion and SSCH have similar aggregate throughput, Dominion sustains long flows better. The difference between the two is reflected when the throughput is normalized, i.e., each flow's throughput is given a weight equivalent to the distance between its end-nodes. With 50 simultaneous random flows, on average, Dominion achieves 1813% and 86% higher aggregate distance-normalized throughput than 802.11 and SSCH respectively. The discrepancy between normalized and actual throughput shows that smaller flows seem to dominate in SSCH. Furthermore, Jain's fairness index [11] was used to evaluate flow distribution – both SSCH and 802.11 fare poorly compared to Dominion. With 50 simultaneous random flows, Dominion is, on average, 1730% and 315% fairer than both 802.11 and SSCH respectively. SSCH does achieve higher actual throughput than Dominion for a small number ($<$20) of simultaneous flows, however, Fig. 3(b) shows that in terms of normalized throughput, both perform similarly (with Dominion performing much better with more flows). This implies that Dominion achieves lower throughput (for those scenarios) only because of its tendency to be fair.

V. Related Work

Previous research efforts can be divided into two categories: solutions that exploit multi-channel diversity, and others that exploit multi-user diversity with multi-path routing.

Multi-channel Networks: The first category of multi-channel network utilize a specialized transceiver. Nasipuri et al [12] propose a multi-channel CSMA protocol assumes that a transceiver can listen to all available channels simultaneously. A transmitting node chooses an idle (or least busy) channel to deliver data packets to its destination. The protocol is improved in [13], where the receiver selects the transmission channel. The next category of multi-channel networks utilize multiple commodity transceivers. Dynamic Channel Assignment (DCA) [14] requires two transceivers per node, one for the control channel communication and the other for actual transmission of data packets. Based on information



(a) Actual aggregate throughput. (b) Normalized aggregate throughput. (c) Jain's fairness index.

Fig. 3. **Random flows in a multi-hop wireless mesh network.**

collected through the dedicated control channel transceiver, a node switches the second transceiver when it needs to transmit or receive data. Jain et al [15] propose a similar scheme to DCA, in which the receiver chooses the channel to commence data communication. Hyacinth [16] requires two transceivers per node, both of which are assigned different static channels. The nodes are assigned channels such that the network remains connected. Kyasanur and Vaidya [8] propose a routing protocol that works with two transceivers: the first transceiver is assigned a static home channel, whereas the second transceiver channel hops to transmit packets. The two primary research efforts that utilize only one commodity transceiver are MMAC [4] and SSCH [5].

Multi-path Routing: Opportunistic Multi-path Scheduling (OMS) [17] uses multiples routes via intelligent scheduling. Extremely Opportunistic Routing (ExOR) [18] forwards a series of packets through nodes, deferring the choice of next hop until after the previous node has transmitted the packet.

VI. Conclusion and Future Work

In this paper, we propose a new cross-layer architecture for wireless mesh networks. As a result, we presented the first graph theoretic model for the multi-channel wireless mesh network. In Section II, we introduced the MAC scheduler that makes full usage of all available channels. The Dominion routing algorithm presented in Section III is defined using an abstract graph model (in Section III-B). Finally, in Section IV, we implemented the Dominion architecture and protocols in the QualNet [6] simulator. Our simulation results show that, on average, Dominion (with 11 channels), when handling 50 simultaneous random flows, improves the normalized throughput, by 1813% and 86%, along with fairness by 1730% and 315%, over 802.11 and SSCH respectively.

Future Work: The biggest drawback of the Dominion architecture is possibly high end-to-end latency. End-to-end latency can be minimized by reducing the time slot duration. A related optimization is the run-time adjustment of the number of utilized channels based on spatial density and network utilization. Networks with low density and low parallelism (of flows) may achieve lower end-to-end latency with fewer channels. TCP performance also needs to be improved as multiple paths lead to out-of-order packet delivery. We believe a "middleware" packet re-sequencer should improve TCP performance[18]. Lastly, while experimenting, we discover that utilizing fewer routes during periods of high contention improves aggreagate throughput. We would like to investigate the run-time adjustment of the number of paths.

Acknowledgment: The authors are thankful to Ranveer Chandra for providing the original implementation of SSCH, and to Nathanael Thompson, Steve Ko, Ramses Morales, Vytautas Valancius, and Michael Earnhart for their comments and suggestions. This work was supported in part by NSF CAREER grant CNS-0448246, and in part by NSF ITR grant CMS-0427089.

References

[1] "Champaign-Urbana wireless network," http://www.cuwireless.net/.
[2] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *Proc. of ACM MobiCom,* 2005.
[3] J. D. Camp, E. W. Knightly, and W. S. Reed, "Developing and deploying multihop wireless networks for low low-income communities," in *Proc. of Digital Communities*, 2005.
[4] J. So and N. H. Vaidya, "Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *Proc. of ACM MobiHoc*, 2004.
[5] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *Proc. of ACM MobiCom*, 2004.
[6] "Qualnet user manual v3.9," http://www.qualnet.com/.
[7] A. Mishra, S. Banerjee, and W. Arbaugh, "Weighted coloring based channel assignment for WLANs," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 3, pp. 19–31, 2005.
[8] P. Kyasanur and N. H. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," in *Proc. of IEEE WCNC*, 2005.
[9] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. of ACM MobiCom*, 2003.
[10] F. Herzel, G. Fischer, and H. Gustat, "An integrated CMOS RF synthesizer for 802.11a wireless LAN," vol. 38, no. 10, Oct. 2003.
[11] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Digital Equipment Corporation, Tech. Rep. TR-301, 1984.
[12] A. Nasipuri, J. Zhuang, and S. R. Das, "A multichannel CSMA MAC protocol for multihop wireless networks," in *Proc. of IEEE WCNC*, 1999.
[13] A. Nasipuri and S. R. Das, "Multichannel CSMA with signal power-based channel selection for multihop wireless networks," in *Proc. of IEEE Vehicular Technology Conference (VTC)*, 2000.
[14] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks," in *Proc. of ISPAN*, 2000.
[15] N. Jain, S. R. Das, and A. Nasipuri, "A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks," in *Proc. of Intl. Conf. on Computer Communications and Networks (IC3N)*, 2000.
[16] A. Raniwala and T. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proc. of IEEE INFOCOM*, 2005.
[17] C. Cetinkaya and E. Knightly, "Opportunistic traffic scheduling over multiple network paths," in *Proc. of IEEE INFOCOM*, 2004.
[18] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," in *Proc. of ACM SIGCOMM*, 2005.