Multi-Channel Wireless Networks: Capacity and Protocols*

Technical Report

April 2005

Pradeep Kyasanur
Dept. of Computer Science, and
Coordinated Science Laboratory,
University of Illinois at Urbana-Champaign
Email: kyasanur@uiuc.edu

Nitin H. Vaidya
Dept. of Electrical and Computer Engineering, and
Coordinated Science Laboratory,
University of Illinois at Urbana-Champaign
Email: nhv@uiuc.edu

***Abstract*— Wireless technologies, such as IEEE 802.11a, provide for multiple non-overlapping channels. Typical multi-hop wireless network configurations have only used a single channel for the network. The available network capacity can be increased by using multiple channels, and nodes can be equipped with multiple interfaces to utilize the available channels. However, the number of interfaces per node is expected to be smaller than the number of channels. We establish the capacity of multi-channel networks under this scenario. We develop novel link layer and routing protocols that are designed specifically for multi-channel operation. Simulation results demonstrate the effectiveness of the proposed approach in significantly increasing network capacity, by utilizing all the available channels, even when the number of interfaces is smaller than the number of channels.**

I. INTRODUCTION

Wireless technologies, such as IEEE 802.11 [1], provide for multiple non-overlapping channels. Multiple channels have been utilized in infrastructure-based networks by assigning different channels to adjacent access points, thereby minimizing interference between access points. However, typical multi-hop wireless network configurations have used a single channel to ensure all nodes in the network are connected. For meeting the ever-increasing throughput demands of applications, it is necessary to utilize all of the available spectrum, and this requires the development of new protocols specifically designed for multi-channel operation.

Wireless hosts have typically been equipped with one wireless interface. However, a recent trend of reducing hardware costs [2] has made it feasible to equip nodes with multiple interfaces. Nevertheless, it is still expensive to equip a node with *one dedicated interface for each channel*, as the number of channels may be large. Even if each channel does not have a dedicated interface, currently available commodity wireless interfaces (such as IEEE 802.11 wireless interface cards) can be *switched* from one channel to another, albeit at the cost of a switching latency, thereby allowing all channels to be potentially utilized. Thus, it is of practical interest to develop an architecture and protocols for the scenario wherein the *number of interfaces per node is smaller than the number of channels*.

Past research on wireless network capacity [3], [4] has typically considered wireless networks with a single channel, although the results are applicable to a wireless network with multiple channels as well, provided that at each node there is a dedicated interface per channel. When nodes are not equipped with a dedicated interface per channel, then *capacity degradation* may occur, compared to using a dedicated interface per channel. In this paper, we characterize the impact of number of channels and interfaces per node on the network capacity, and show that in certain scenarios, *even with only a single interface per node, there is no capacity degradation.* This implies that it may be possible to build capacity-optimal multi-channel networks with as few as *one interface per node.* When interface switching latency is accounted for, capacity-optimal performance can be achieved by using a few interfaces per node instead of just a single interface.

---

[*] This research was supported in part by NSF grant ANI-0125859 and a Vodafone Graduate Fellowship.

We also develop link layer and routing protocols for multi-channel networks. Our solution requires at least two interfaces per node to simplify protocol design. We propose a novel interface assignment strategy that keeps one interface fixed on a specific channel, while other interfaces can be switched, as necessary, among the remaining channels. The use of a fixed interface simplifies coordination, while the switchable interfaces enable the utilization of all the available channels.

Traditional routing protocols do not account for channel diversity. For example, when shortest-path routing is used, a k hop route that traverses all the hops on a single channel has the same cost as an alternate k hop route that uses different channels for each hop. However, the throughput of a route that uses a single channel on all hops can be substantially smaller than a route that uses multiple channels, on account of *self interference* along the route. Furthermore, when the switching latency is non-negligible, the routing protocol has to account for the cost of interface switching when selecting routes. Our proposed routing protocol is designed to choose *channel-diverse routes*, while accounting for the cost of switching latency.

Evaluations show that our proposal is effective in utilizing multiple channels. For example, our results show that even with two interfaces, a five channel network can offer more than five-fold improvement over a single channel network. The proposed link layer and routing protocols are similar to the algorithms used in the constructive proof that achieves the upper bound for multi-channel capacity. Hence, the proposals are validated to be an effective choice by theory as well.

The rest of the paper is organized as follows. We describe related work in Section II. We present our theoretical results in Section III. Certain design issues are described in Section IV and assumptions made are discussed in Section V. Sections VI and VII describe the details of the proposed link and routing protocols. We evaluate our protocols in Section VIII. We discuss possible extensions to our protocols in Section IX, and conclude in Section X.

II. RELATED WORK

A. Capacity of wireless networks

The detailed capacity results and proofs are included in another technical report [5].

In their seminal work, Gupta and Kumar [3] derived the capacity of ad hoc wireless networks. The results are applicable to single channel wireless networks, or multi-channel wireless networks where every node has a dedicated interface per channel. We extend the results of Gupta and Kumar to those multi-channel wireless networks where nodes may not have a dedicated interface per channel, and we also consider the impact of interface switching delay on network capacity.

Grossglauser and Tse [4] showed that mobility can improve network capacity, though at the cost of increased end-to-end delay. Subsequently, other research [6], [7] has analyzed the trade-off between delay and capacity in mobile networks. Gamal et al. [8] characterize the optimal throughput-delay trade-off for both static and mobile networks. In this paper, we adapt some of the proof techniques presented by Gamal et al. [8] to the multi-channel capacity problem.

Recent results have shown that the capacity of wireless networks can be enhanced by introducing infrastructure support [9]–[11]. Other approaches for improving network capacity include the use of directional antennas [12], and the use of unlimited bandwidth resources (UWB) albeit with power constraints [13], [14].

Li et al. [15] have used simulations to evaluate the capacity of multi-channel networks based on IEEE 802.11. Other research on capacity is based on considerations of alternate communication models [16]–[18].

B. Multi-channel MAC and link layer protocols

Several researchers have proposed MAC protocols based on IEEE 802.11 for utilizing multiple channels. Nasipuri et al. [19], [20], and Jain et al. [21] propose a class of protocols where all nodes have an interface on each channel. The protocols differ in the metric used to choose a channel for communication between a pair of nodes. The metric may simply be to use an idle channel [19], or the signal power observed at the sender [20], or the received signal power at the receiver [21]. Wu et al. [22] propose a MAC layer solution that requires two interfaces. One interface is assigned to a common channel for control purposes, and the second interface is switched between the remaining channels and used for data exchange. Hung et al. [23] propose a similar two-interface solution that uses a channel load-aware algorithm to choose the appropriate data channel to be used for data exchange. So et al. [24] propose a MAC solution for multiple channels that uses a single interface.

All the multi-channel MAC proposals described above require changes to IEEE 802.11, and therefore cannot be deployed by using commodity hardware. In contrast, our proposal can be implemented with standard 802.11 interfaces.

Adya et al. [25] propose a link-layer solution for striping data over multiple interfaces. The proposal does not use interface switching, and for full utilization of available channels, an interface is necessary for each channel. Bahl et al. [26] propose SSCH, a link-layer solution that uses a *single interface*, and can run over unmodified IEEE 802.11 MAC. In this paper, we propose a new interface assignment strategy designed for *multiple interfaces* that can be implemented at the link-layer.

Multi-channel solutions implemented at the MAC or the link layer are not sufficient for effectively utilizing multiple channels, as the routing protocol may select routes wherein successive hops interfere with each other. In this paper, we propose a *multi-channel aware routing protocol* that complements our proposed interface assignment strategy.

In the context of *wired local area networks*, Marsan et al. [27] have studied the performance of multichannel CSMA/CD MAC protocols, and shown that significant reduction in delay average and variance is possible even when the number of interfaces is less than the number of channels. This paper is motivated by the need to answer a similar question with multi-channel CSMA/CA based *wireless networks*. Our evaluations show that in the case of multi-channel wireless networks as well, significant performance improvement is possible, even if the number of interfaces is less than the number of channels.

C. Multi-channel routing protocols

Shacham et al. [28] have proposed a architecture for multi-channel wireless networks that uses a single interface. Each node has a default channel for receiving data. A node with a packet to transmit has to switch to the channel of the receiver before transmitting data. However, the proposal does not consider the impact of switching latency. Furthermore, the routes used in the architecture may not utilize all the available channels.

So et al. [29] have proposed a routing protocol for multi-channel networks that uses a single interface at each node. We propose to use multiple interfaces, which can offer better performance than a single interface solution. Furthermore, the routing protocol requires complex coordination among communicating nodes, when setting up routes.

Existing routing protocols for multi-hop networks such as DSR [30] and AODV [31] support multiple interfaces at each node. However, those protocols typically select shortest-path routes, which may not be suitable for multi-channel networks [32]. Furthermore, the protocols cannot exploit all the available channels, if the number of interfaces is smaller than the number of channels.

We are aware of two routing protocols specifically designed for multi-channel, multi-interface wireless networks. Draves et al. [32] have proposed LQSR, a source routing protocol for multi-channel, multi-interface networks. LQSR uses WCETT, a new metric designed for multi-channel networks, and ensures "high-quality" routes are selected. Our proposal differs from LQSR in the following key aspects:

- LQSR assumes the number of interfaces is equal to the number of channels used by the network. In contrast, our proposal is designed to handle the scenario where the number of available interfaces may be smaller than the number of available channels, and therefore uses *interface switching*.
- LQSR is designed for *static*, multi-hop wireless networks, such as mesh networks, and does not account for the impact of node mobility on the routing heuristic. Our proposal is designed for general *mobile* ad hoc wireless networks, and can be used in mesh networks as well.

Raniwala et al. [33], [34] propose routing and interface assignment algorithms for static networks. Their goal is similar to our work in addressing the scenario where the number of available interfaces is less than the number of available channels. However their approach is different in the following key aspects:

- The protocols are designed for use in static networks where traffic is directed toward specific gateway nodes. The communication pattern that arises in such networks is a tree that is rooted at each gateway node. In contrast, our proposal is designed for a more general communication pattern, where any node may communicate with any other node.
- Raniwala's protocol assumes nodes are *stationary* and traffic load between all nodes are known. Using the load information, interface assignment and route computation is intelligently done. In contrast, we assume no such load information is available, as in an ad hoc network, nodes may frequently move, resulting in changing load conditions over time. Thus, we consider the multi-channel, multi-interface routing problem in more general *mobile* ad hoc networks.

We have presented some of the ideas discussed in this report in an earlier paper [35]. This report also extends an earlier technical report [36].

III. CAPACITY ANALYSIS OF MULTI-CHANNEL NETWORKS

In this section, we present the results of our capacity analysis. The detailed proofs are included in another technical report [5].

A. Modeling multi-channel multi-interface networks

We consider a static wireless network containing n nodes. We use the term "channel" to refer to a part of the frequency spectrum with some specified bandwidth. There are c channels, and we assume that every node is equipped with m interfaces, $1 \leq m \leq c$. We assume that an interface is capable of transmitting or receiving data on any one channel at a given time. We use the notation (m, c)-network to refer to a network with m interfaces per node, and c channels.

We define two channel models to represent the data rate supported by each channel:

Channel Model 1: In model 1, we assume that the total data rate possible by using all channels is W. The total data rate is divided equally among the channels, and therefore the data rate supported by any one of the c channels is W/c. This was the channel model used by Gupta and Kumar [3], and we primarily use this model in our analysis. In this model, as the number of channels increases, each channel supports a smaller data rate. This model is applicable to the scenario where the total available bandwidth is fixed, and new channels are created by partitioning existing channels.

Channel Model 2: In model 2, we assume that each channel can support a fixed data rate of W, independent of the number of channels. Therefore, the aggregate data rate possible by using all c channels is Wc. This model is applicable to the scenario where new channels are created by utilizing additional frequency spectrum.

The results presented in this paper are derived assuming channel model 1. However, all the derivations are applicable for channel model 2 as well, and the results for model 2 can be obtained by replacing W in the results of model 1 by Wc.

B. Definitions

We study the capacity of static multi-channel wireless networks under the two settings introduced by Gupta and Kumar [3].

Arbitrary Networks: In the arbitrary network setting, the location of nodes, and traffic patterns can be controlled. Since any suitable traffic pattern and node placement can be used, the bounds for this scenario are applicable to any network. The arbitrary network bounds may be viewed as the *best case* bounds on network capacity. The network capacity is measured in terms of "bit-meters/sec" (originally introduced by Gupta and Kumar [3]). The network is said to transport one "bit-meter/sec" when one bit has been transported across a distance of one meter in one second.

Random Networks: In the random network setting, node locations are randomly chosen, and each node sets up one flow to a randomly chosen destination. The network capacity is defined to be the aggregate throughput over all the flows in the network, and is measured in terms of bits/sec.

We use the following notation to represent bounds:

1) $f(n) = O(g(n))$ implies there exists some constant d and integer N such that $f(n) \leq dg(n)$ for $n > N$.
2) $f(n) = o(g(n))$ implies that $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$.
3) $f(n) = \Omega(g(n))$ implies $g(n) = O(f(n))$.
4) $f(n) = \omega(g(n))$ implies $g(n) = o(f(n))$.
5) $f(n) = \Theta(g(n))$ implies $f(n) = O(g(n))$ and $g(n) = O(f(n))$.
6) $MIN_O\,(f(n), g(n))$ is equal to $f(n)$, if $f(n) = O(g(n))$, else, is equal to $g(n)$.

The bounds for random networks hold *with high probability (whp)*. In this paper, *whp* implies with "probability 1 when $n \to \infty$."

C. Main Results

Gupta and Kumar [3] have shown that in an arbitrary network, the network capacity scales as $\Theta\left(W\sqrt{n}\right)$ bit-meters/sec, and in a random network, the network capacity scales as $\Theta\left(W\sqrt{\frac{n}{\log n}}\right)$ bits/sec. Under the channel model 1, which was the model used by Gupta and Kumar [3], the capacity of a network with a single channel and one interface per node (that is, a $(1, 1)$-network in our notation) is equal to the capacity of a network with c channels and $m = c$ interfaces per node (that is, a (c, c)-network). Furthermore, under both channel models, the capacity of a (c, c)-network is at least as large as the capacity of a (m, c)-network, when $m \leq c$ (this is trivially true, by not using $c - m$ interfaces in the (c, c)-network). In the results presented in this paper, we capture the impact of using fewer than c interfaces per node by establishing the *loss in capacity*, if any, of a (m, c)-network in comparison to a (c, c)-network.

The goal of this work is to study the impact of the number of channels c, and the number of interfaces per node m, on the capacity of arbitrary and random networks. Our results show that the capacity is

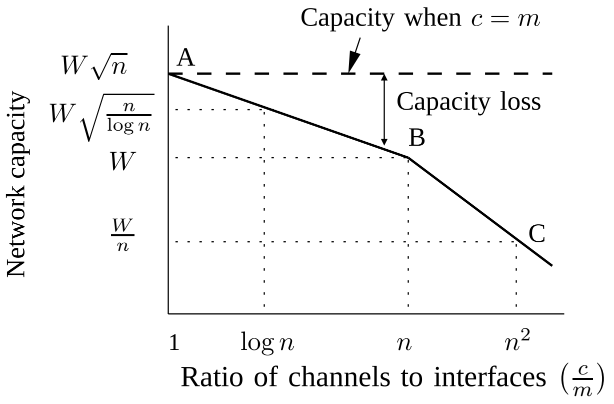

Fig. 1. Impact of number of channels on capacity scaling in arbitrary networks (figure is not to scale)

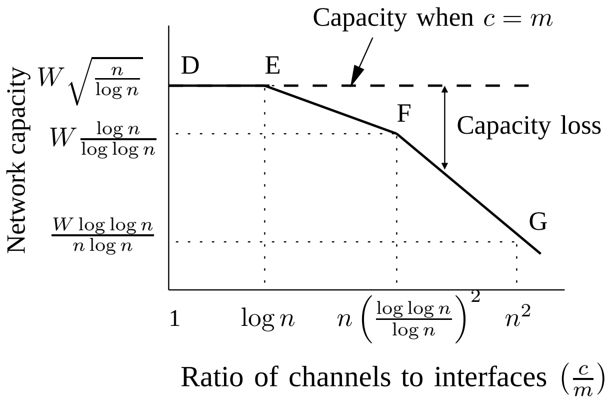

Fig. 2. Impact of number of channels on capacity scaling in random networks (figure is not to scale)

dependent on the ratio $\frac{c}{m}$, and not on the exact values of either c or m. We now state our main results under channel model 1.

1. Results for arbitrary network: The network capacity of a (m, c)-network has two regions (see Figure 1) as follows:

1) When $\frac{c}{m}$ is $O(n)$, the network capacity is $\Theta\left(W\sqrt{\frac{nm}{c}}\right)$ bit-meters/sec (segment A-B in Figure 1). Compared to a (c, c)-network, there is a capacity loss by a factor of $1 - \sqrt{\frac{m}{c}}$.
2) When $\frac{c}{m}$ is $\Omega(n)$, the network capacity is $\Theta\left(W\frac{nm}{c}\right)$ bit-meters/sec (line B-C in Figure 1). In this case, there is a larger capacity degradation than case 1, as $\frac{nm}{c} \leq \sqrt{\frac{nm}{c}}$ when $\frac{c}{m} \geq n$.

Therefore, there is always a capacity loss in arbitrary networks whenever the number of interfaces per node is fewer than the number of channels.

2. Results for random network: The network capacity of a (m, c)-network has three regions (see Figure 2) as follows:

1) When $\frac{c}{m}$ is $O(\log n)$, the network capacity is $\Theta\left(W\sqrt{\frac{n}{\log n}}\right)$ bits/sec (segment D-E in Figure 2). In this case, *there is no loss* compared to a (c, c)-network. Hence, in many practical scenarios where c may be constant or small, *a single interface per node suffices.*
2) When $\frac{c}{m}$ is $\Omega(\log n)$ and also $O\left(n\left(\frac{\log\log n}{\log n}\right)^2\right)$, the network capacity is $\Theta\left(W\sqrt{\frac{nm}{c}}\right)$ bits/sec (segment E-F in Figure 2). In this case, there is some capacity loss. Furthermore, in this region, the capacity of a (m, c)-random network *is the same* as that of a (m, c)-arbitrary network (segment E-F in Figure 2 overlaps part of segment A-B in Figure 1), implying "randomness" does not incur a capacity penalty.
3) When $\frac{c}{m}$ is $\Omega\left(n\left(\frac{\log\log n}{\log n}\right)^2\right)$, the network capacity is $\Theta\left(\frac{Wnm\log\log n}{c\log n}\right)$ bits/sec (line F-G in Figure 2). In this case, there is a larger capacity degradation than case 2. Furthermore, in this region, the capacity of a (m, c)-random network *is smaller than* that of a (m, c)-arbitrary network, in contrast to case 2.

3. Other results: The results presented above are derived under the assumption that there is no delay in switching an interface from one channel to another. However, it can be shown [5] that even if interface switching delay is considered, the network capacity is not reduced, provided a few additional interfaces are provisioned for at each node. This implies that it is possible to *hide the interface switching delay* by using extra interfaces in conjunction with carefully designed routing and transmission scheduling protocols.

D. Implications of capacity results

A common scenario of operation is when the number of channels is not too large ($\frac{c}{m} = O(n)$). Under this scenario, the capacity of a (m, c)-network in the arbitrary setting scales as $\Theta\left(W\sqrt{\frac{nm}{c}}\right)$ under channel model 1. Similarly, under channel model 2, the capacity of the network scales as $\Theta\left(W\sqrt{nmc}\right)$. Under either model, the capacity of a (m, c)-network goes down by a factor of $1 - \sqrt{\frac{m}{c}}$, when compared with a (c, c)-network. Therefore, doubling the number of interfaces at each node (as long as number of interfaces is smaller than the number of channels) increases the channel capacity by a factor of only $\sqrt{2}$. Furthermore, the ratio between m and c decides the capacity, rather than the individual values of m and c. Increasing the number of interfaces may result in a linear increase in the cost but only a sub-linear (proportional to square-root of number of interfaces) increase in the capacity. Therefore, the *optimal number of interfaces to use may be smaller than the number of channels* depending on the relationship between cost of interfaces and utility obtained by higher capacity.

Different network architectures have been proposed for utilizing multiple channels when the number of available interfaces is smaller than the number of available channels [32], [34], [35]. The construction we use in proving lower bound [5] implies that maximal capacity is achieved when *all channels are utilized*. One architecture used in the past [32] is to use only m channels when m interfaces are available, leading to wastage of the remaining $c - m$ channels. That architecture results in a factor of $1 - \frac{m}{c}$ loss in capacity which can be significantly higher than the optimal $1 - \sqrt{\frac{m}{c}}$ loss (when $\frac{c}{m} = O(n)$). Hence, in general, *higher capacity may be achievable by architectures that use all channels, possibly by dynamically switching channels*.

Our results suggest that the capacity of multi-channel random networks with total channel data rate of W is the same as that of a single channel network with data rate W as long as the ratio $\frac{c}{m}$ is $O(\log n)$. When the number of nodes n in the network increases, we can also scale the number of channels (for example, by using additional bandwidth, or by dividing available bandwidth into multiple sub-channels). Even then, as long as the channels are *scaled at a rate not more than* $\log n$*, there is no loss in capacity even if a single interface is available at each node*. In particular, if the number of channels c is a fixed constant, independent of the node density, then as the node density increases beyond some threshold density (at which point $c \leq \log n$), there is no loss in capacity even if just a single interface is available per node. Thus, this result may be used to roughly estimate the number of interfaces each node has to be equipped with for a given node density and a given number of channels.

In a single channel random network, i.e., a $(1, 1)$-network, the capacity bottleneck arises out of the channel becoming fully utilized, and not because interface at any node is fully utilized. On an average, the interface of a node in a single channel network is busy only for $\frac{1}{X}$ fraction of the time, where X is the average number of nodes that interfere with a given node. In a $(1, 1)$-random network with n nodes, each node on an average has $\Theta(\log n)$ neighbors to maintain connectivity [3]. This implies that in a single channel network, each interface is busy for only $\Theta\left(\frac{1}{\log n}\right)$ time. Intuitively, our construction above utilizes this slack time of interfaces to support up to $O(\log n)$ channels without loss in capacity. In general, there is *no loss in capacity in a random network as long as the number of channels is smaller than the average number of nodes in the neighborhood*[1] *of a node*.

When the number of channels is large (specifically, $\omega(\log n)$) and each node has a single interface, there is a capacity loss when compared to a single channel network. This capacity loss arises because the number of channels is more than the number of interfaces in a "neighborhood". The lower bound construction [5] suggests that *an optimal strategy for maximizing capacity* when number of channels is large is to sufficiently increase the *transmission power used* to ensure neighborhood size is sufficiently large. This ensures that number of interfaces in a neighborhood will then be equal to the number of channels. However, there is still some capacity loss because larger transmission power (than that is needed for connectivity alone) lowers capacity by "consuming" more area.

Some of the implications of our capacity results on protocol design are discussed in Section IV-B.

IV. DESIGN ISSUES

In this section, we first motivate the benefits of using a multi-interface solution for exploiting multiple channels. We then identify the need for specialized routing protocols for multi-channel, multi-interface networks.

A. Benefits of using multiple interfaces

We define "interface" to be a network interface card equipped with a half-duplex radio transceiver, e.g., a commodity 802.11 wireless card. In most multi-hop networks, a single channel is used, and therefore a single interface suffices. However, when multiple channels are available, having more than one interface is beneficial.

Our capacity analysis has shown that when switching delay is negligible, or when there are no latency constraints, a single interface per node may suffice in achieving optimal capacity in random networks. However, in practice, switching delay is often non-negligible,

---

[1] The neighborhood of a node consists of all other nodes that may interfere with it.

and applications typically expect end-to-end latency to be reasonably small. Under these constraints, our analysis shows that multiple interfaces per node may be required for achieving optimal capacity. Apart from the theoretical need for multiple interfaces to achieve asymptotically optimal capacity, many practical concerns, described below, motivate the use of multiple interfaces as well.

When using a single interface per node, if the interfaces of two nodes are on different channels, then they cannot communicate. For reducing synchronization requirements and overheads, each interface has to stay on a channel for many packet transmission durations (100ms in [24] and 10ms in [26]). As a result, when packets are traversing multi-hop paths, packets may be delayed at each hop, unless the next hop is on the same channel as well. Thus, when a single interface is used, there is an increase in the end-to-end latency if different hops traversed are on different channels. Otherwise, if most hops are on the same channel, transmissions on consecutive hops interfere, reducing the maximum capacity. In either case, TCP throughput is significantly affected.

When at least two interfaces are available, we propose keeping one interface permanently assigned to a channel to greatly simplify coordination, while switching the second interface (based on traffic requirements) to avoid delaying a packet at each hop. We defer discussion of the proposed approaches till later in the paper, but multiple interfaces are required to derive both simplicity in coordination and minimal delays.

A second benefit is the ability to receive and transmit data in parallel. Half-duplex wireless interfaces cannot simultaneously transmit and receive data. However, when multiple (say two) interfaces and multiple channels are available, while one interface is receiving data on one channel, the second interface can simultaneously transmit data on a different channel. In many cases, this can double the maximum throughput achievable on a multi-hop route. Our proposed architecture exploits this benefit of using multiple interfaces as well.

B. *Insights obtained from capacity analysis*

The constructions used in establishing capacity results [5] offer insights into optimal link layer and routing strategies that need to be used for multi-channel networks.

In the transmission scheduling scheme used in our lower bound construction proof (presented in [5]), *it suffices for a node to always transmit on a specific channel without requiring to switch channels* for different packets. However, a node may have to switch channels for receiving data. An alternate construction is to use a scheduling scheme which ensures that a node receives all data on a specific channel, but may have to switch channels when sending data. It can be shown that the alternate construction is equivalent to the lower bound construction. This intuition can be used in developing a practical scheme that uses two interfaces per node. One interface can be used for receiving data and is always fixed to a single channel (for long time intervals). The second interface can be used for sending data and is switched between channels, as necessary. Existing multi-channel protocols have often required tight synchronization among nodes. The use of two interfaces, with a dedicated interface on a fixed channel obviates the need for tight synchronization as a node receives data on a well-known channel. Furthermore, using a fixed channel for reception does not degrade capacity since it is based on the (optimal) alternate construction.

The lower bound construction also suggests that *load balancing* (i.e., distributing routes) among nodes in a given neighborhood is essential for full utilization of multiple channels. Existing routing protocols for multi-hop networks such as DSR [30] and AODV [31] typically select shortest-hop routes, and do not incorporate load balancing. In addition, route selection does not consider the interface switching cost, and the chosen routes may require frequent channel switching, degrading network performance. In a single channel network, load balancing is sometimes used to balance energy consumption across nodes, or to improve resilience of the network. However, load balancing in the same neighborhood is not always required in single channel networks for maximizing capacity. Thus, there is a need for a customized routing protocol for multi-interface, multi-channel networks.

Figure 3 illustrates a scenario that highlights the need for specialized routing protocols for multi-channel networks. In the figure, node A is communicating with node D using route A-C-D. Node E wishes to communicate with node F, and either of B or C can be used as the intermediate node. Assume all nodes have a single interface, and assume C and B can relay at most w bytes per second. If node C is chosen as the intermediate node, then node C has to forward data along both routes A-C-D and E-C-F, and the throughput received by each flow is at most $w/2$. On the other hand if node B is chosen as the intermediate node, then both routes A-C-D and E-B-F can be simultaneously used (assuming channels used on routes A-C-D and E-B-F can be chosen to be orthogonal), and each flow receives a rate of w. Although

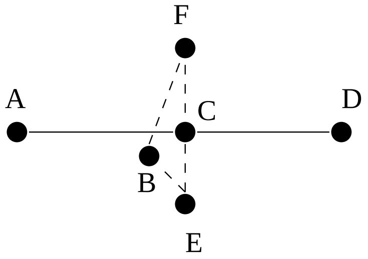

Fig. 3. Impact of route selection on effective utilization of multiple channels

this example assumed each node had a single interface, similar issues arise even when multiple interfaces are available.

The above scenario highlights the need for the routing protocol to appropriately *distribute* routes among nodes in the neighborhood. In the case of single channel networks, the throughput obtained is the same whether B or C is chosen as the intermediate node. When a single channel is available, and say, when C is transmitting a packet along route A-C-D, B cannot transmit a packet even if it is chosen as the intermediate node (as the common channel is busy). Consequently, routing protocols designed for single channel networks do not need to distribute routes within a neighborhood. However, to exploit the benefit of multiple channels, it is important for a routing protocol to ensure routes are carefully distributed in the network.

V. ASSUMPTIONS

In this section, we describe the assumptions we make while developing our protocols.

A. Interface switching cost

The ability to switch an interface from one channel to another is a key property we exploit to utilize all the available channels, even when the number of interfaces available is significantly lesser than the number of available channels. We assume that channels are separated in frequency, and switching an interface requires changing the frequency of operation. Switching an interface from one channel to another incurs some delay D which may be non-negligible. In the current literature, estimates for D (for switching between channels on the same frequency band) with commodity IEEE 802.11 hardware are in the range of a few milliseconds [37] to a few hundred microseconds [38]. It is expected that with improving technology, the switching delay will reduce to a few tens of microseconds [26]. Protocols that utilize interface switching need to be flexible enough to accommodate a range of switching delays. The routing protocol may have to account for the switching cost while selecting routes.

Interface switching is possible across different frequency bands as well. For example, wireless cards are currently available that support both IEEE 802.11a (operates on 5 GHz band) and IEEE 802.11b (operates on 2.4 GHz band), and can switch between the two bands. However, with the currently available hardware, switching across bands incurs a large delay, but the switching delay is expected to reduce in the future. The architecture presented in this paper allows for the utilization of channels on the same band as well as channels on different bands.

B. Orthogonal channels

IEEE 802.11a offers 12 non-overlapping channels, while IEEE 802.11b offers 3 non-overlapping channels. When a single node is equipped with multiple interfaces, it has been noted [32] that communication on different interfaces using adjacent non-overlapping channels may interfere. Thus, the number of available orthogonal channels may be smaller than the number of non-overlapping channels. Recently, Raniwala et al. [34] have experimentally shown that when separation between interfaces is increased, the interference between interfaces is reduced, allowing more channels to be used simultaneously. Furthermore, future hardware that employs better filters to reduce adjacent channel interference may allow any pair of non-overlapping channels to be simultaneously used. In this paper, we account for the possibility of interference among adjacent non-overlapping channels in our evaluations by assuming that the number of available orthogonal channels is smaller than the number of non-overlapping channels. We believe that a careful use of channels will offer at least 3 to 5 orthogonal channels in the 5 GHz band (IEEE 802.11a). Improved hardware in the future may enable all 12 channels to be used orthogonally. Future changes in FCC regulations may provide for more orthogonal channels as well.

C. Problem Formulation

The protocols proposed in this paper are designed for a multi-hop wireless network. Nodes in the network can be mobile. We assume that the typical traffic pattern involves communication between arbitrary pair of nodes. Specifically, we *do not* require the presence of special gateway nodes that may be the source or destination of all traffic in the network, although our proposal can be used in that scenario as well.

We define the requirements of a multi-channel, multi-interface solution as follows:

1) Improve network capacity by utilizing all the available channels, even if the number of interfaces is smaller than the number of available channels. The solution must be flexible enough to accommodate different number of channels and interfaces, with channels potentially on different frequency bands.
2) Ensure that a network which is connected when using a single common channel, continues to be connected when multiple channels are being used.
3) Allow implementation using existing IEEE 802.11 hardware.

In Section VI we present the link layer "switching protocol" that manages interface switching. This link layer protocol can be used in conjunction with commonly used routing protocols such as AODV and DSR. However, to obtain the full benefits of using multiple channels, existing routing protocols are not sufficient (details in Section VII). We then present a new routing protocol, called Multi-Channel Routing Protocol in Section VII that is designed specifically for multi-channel networks, and operates in conjunction with the switching protocol.

VI. SWITCHING PROTOCOL

When the number of available interfaces is smaller than the number of available channels, an interface assignment strategy is required to assign interfaces to specific channels. Furthermore, for using all the available channels, a "switching protocol" is necessary to decide when to switch an interface from one channel to another. The switching protocol has to ensure that the neighbors of a node X can communicate with it on-demand, which requires all neighbors of X to be always aware of the channel being used by at least one interface of X. We first identify the different interface assignment strategies possible. We then describe our proposal and discuss issues involved.

A. Classification of interface assignment strategies

Interface assignment strategies can be classified into static, dynamic, and hybrid strategies.

1. Static Assignment: Static assignment strategies assign each interface to a channel either permanently, or for "long intervals" of time where "long interval" is defined relative to the interface switching time. For example, [32], [33] use static interface assignment. Static assignment can be further classified into two types:

1) Common channel approach: In this approach, interfaces of all nodes are assigned to a common set of channels (e.g. [32]). For example, if two interfaces are used at each node, then the two interfaces are assigned to the same two channels at every node. The benefit of this approach is that the connectivity of the network is the same as that of a single channel approach. Note that the scenario where a single channel and a single interface is used is a special case of the static, common channel assignment strategy.
2) Varying channel approach: In this approach, interfaces of different nodes may be assigned to a different set of channels (e.g. [33]). With this approach, there is a possibility that the length of the routes between nodes may increase. Also, unless the interface assignment is done carefully, network partitions may arise.

Static assignment strategies are well-suited for use when the interface switching delay is large. In addition, if the number of available interfaces is equal to the number of available channels, interface assignment problem becomes trivial. Static assignment strategies do not require special coordination among nodes (except perhaps to re-assign interfaces over long intervals of time) for data communication. With static assignment, nodes that share a channel on one of their interfaces can directly communicate with each other, while others cannot. Thus, the effect of static channel assignment is to control the network topology by deciding which nodes can communicate with each other.

2. Dynamic Assignment: Dynamic assignment strategies allow any interface to be assigned to any channel, and interfaces can frequently switch from one channel to another. In this setting, two nodes that need to communicate with each other need a coordination mechanism to ensure they are on a common channel at some point of time. For example, the coordination mechanism may require all nodes to visit a common "rendezvous" channel periodically (e.g. [24]), or require other mechanisms such as the use of pseudo-random sequences (e.g. [26]), etc. The benefit of dynamic assignment is the ability to switch an interface to any channel, thereby offering the potential to cover many channels with few interfaces. The key challenge with dynamic switching strategies is to coordinate the decisions of when to switch interfaces as well as what channel to switch the interfaces to, among the nodes in the network.

3. Hybrid Assignment: Hybrid assignment strategies combine static and dynamic assignment strategies by applying a static assignment for some interfaces and a dynamic assignment for other interfaces. Hybrid strategies can be further classified based on whether the interfaces that apply static assignment use a common channel approach, or a varying channel approach. An example of hybrid assignment with common channel at the MAC layer is [22], which assigns one interface of each node statically to a common "control" channel, and other interface can be dynamically switched among other "data" channels. We propose to use a hybrid channel assignment strategy with varying channel assignment. Hybrid assignment strategies are attractive as they allow simplified coordination algorithms supported by static assignment while retaining the flexibility of dynamic assignment.

B. Assigning interfaces to channels

We propose a new hybrid assignment strategy. We assume that there are M interfaces available at each node. The available interfaces are divided into two subsets.

- *Fixed Interfaces:* Some K of the M interfaces at each node are assigned for long intervals of time to some K channels, and we designate these interfaces as "fixed interfaces', and the corresponding channels as "fixed channels".
- *Switchable Interfaces:* The remaining $M - K$ interfaces are dynamically assigned to any of the remaining $M - K$ channels (over short time scales), based on data traffic. These interfaces are designated as "switchable interfaces", and the channel to which a switchable interface is assigned to is called a "switchable channel".

Different nodes may assign their K fixed interfaces to a different set of K channels. It is possible for each node to use a different value of K and M, and it is also possible to vary K with time. To simplify rest of the discussion, we assume $M = 2, K = 1$ for all nodes, i.e., there is one fixed, and one switchable interface (although the proposed protocol is applicable to any values of M and K).

The main idea of the interface assignment strategy is to receive data using the fixed interface. Figure 4 illustrates the protocol used for communication between nodes when using "fixed" and "switchable interfaces". Assume that node A has a packet to send to node C via node B. Nodes A, B, and C have their fixed interfaces on channels 1, 2, and 3 respectively. Initially, node A has its switchable interface on channel 3, node B has its switchable interface on channel 1, and node C has its switchable interface on channel 2. In the first step, node A *switches* its switchable interface from channel 3 to channel 2, before transmitting the packet, because channel 2 is the fixed channel of node B. Node B can receive the packet since its fixed interface is always listening to channel 2. In the next step, node B switches its switchable interface to channel 3 and forwards the packet, which is received by node C using its fixed interface. Once the switchable interfaces are correctly set up during a flow initiation, there is no need to switch the interfaces for subsequent packets of the flow (unless a switchable interface has to switch to another channel for sending packets of a different flow).

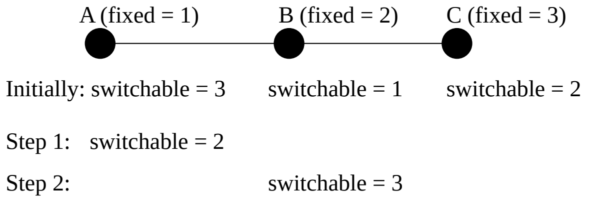

Fig. 4. Example of switching protocol operation with 3 channels, 2 interfaces

Fixed interfaces are assigned to a channel for long intervals of time. The channel to which the fixed interface of a node is assigned to is known by other nodes in the neighborhood (using a protocol described later). Thus, there is no need for special coordination between a sender and a receiver on *when to schedule transmissions to the receiver.* When a node X has to communicate with a node Y over channel i, if the fixed channel being used by X is also i, then the fixed interface is used for communication. Otherwise, the switchable interface of X is switched to channel i for communicating with Y. Therefore, the switchable interface enables a node X to transmit to any node Y in its neighborhood by switching (if required) to the fixed channel used by Y. Different nodes in the neighborhood choose different fixed channels, as we will elaborate later. This flexibility can be used to ensure that all channels in the network are utilized.

In summary, the proposed interface assignment strategy has the following benefits:

- A sender and a receiver do not have to synchronize for channel switching. Thus, the assignment strategy is designed to *not require* a coordination algorithm for ensuring the sender and receiver are on the same channel.
- By carefully balancing the assignment of fixed interfaces of different nodes over the available channels, all channels can be utilized, and the number of contending transmissions in a neighborhood significantly reduces.

- The protocol (described next) can easily scale if the number of available channels increases.

C. Switching Protocol: Fixed interface assignment

Fixed interface assignment involves two components - choosing the channel to be assigned to the fixed interface, and informing neighbors about the channel being used by the fixed interface. The interface assignment protocol has to ensure that fixed interfaces of nodes in a neighborhood are distributed across different channels. For example, suppose a node A uses channel 1 for the fixed interface. Then, all transmissions directed to A will be on channel 1. For balancing the usage of channels, it is beneficial if other nodes in the neighborhood use a different channel for their fixed interface. In general, all nodes within the interference range of a node can interfere with reception on its fixed channel, and it is important to balance the number of nodes that use each channel as their fixed channel.

We propose a localized protocol for fixed interface assignment. Each node maintains a *NeighborTable* containing the fixed channels being used by its neighbors. Nodes also maintain a *ChannelUsageList* containing a count of the number of nodes using each channel as their fixed channel. Initially, a node chooses a random channel for its fixed interface. Periodically, each node broadcasts a “Hello” packet on every channel. The hello packet contains the fixed channel being used by the node. When a node receives a hello packet from a neighbor, it updates its *NeighborTable* and *ChannelUsageList*. Information about the fixed channel used by neighbors can also be obtained by snooping on the route discovery packets (the contents of route discovery packet are described in Section VII).

Each node periodically consults its *ChannelUsageList* (the period chosen is large relative to packet transmission time). If the number of other nodes using the same fixed channel as itself is large, then a node with some probability p changes its fixed channel to a less used channel, and transmits a hello packet informing neighbors of its new fixed channel. The probabilistic approach is used to avoid frequent change of fixed channels by multiple nodes.

The *ChannelUsageList* maintained by a node only tracks the nodes present within its communication range. Nodes outside the communication range can be accounted for if the “Hello” packet also includes *ChannelUsageList*, thereby exchanging two-hop information, though at the cost of increased hello packet size.

The frequency of hello packet exchange depends on the magnitude of average node mobility. A node moving into a new neighborhood cannot communicate with its neighbors until it has exchanged hello packets with them to learn about the fixed channels being used by neighbors. Hello packet exchange is used by many routing protocols (such as AODV) as well, and with moderate degrees of mobility, the overhead of hello packet exchange is not expected to be large.

We do not use channel load information to switch fixed channels. Using channel load may be beneficial if the load in the network does not change frequently. On the other hand, if the load in the network changes frequently, say when there are many short-lived flows, it may lead to frequent and unnecessary channel switching. For example, HTTP transfers often are less than a second, and if such short-lived flows dominate the network traffic, then it may lead to frequent channel switching. Basing fixed channel switching decisions on the network topology requires switching only when the topology changes, which is of the order of tens to hundreds of seconds even with moderate mobility. Hence, we have chosen to switch fixed channels based on the number of nodes using a channel in a given neighborhood.

D. Switching Protocol: Managing switchable interface

The switchable interface of a node X is used to transmit data whenever the fixed channel of the destination is different from the fixed channel of X. One issue to be resolved is how frequently to switch channels. For example, consider a stream of packets at a node X where the even-numbered packets are to destination A, and the odd numbered packets are to destination B, with A and B on different fixed channels. One possibility is to alternately switch between channels for forwarding *each packet*. However, such frequent switching may be very expensive when the switching delay is large. Another possibility is to switch over longer intervals of time, thereby amortizing the cost of switching among multiple packets. Thus, a policy is needed to decide when to switch an interface, and what channel to switch the interface to.

Each channel is associated with a packet queue, as shown in Figure 5. Based on the above discussion, we propose to transmit at most *BurstLength* queued packets on one channel, before switching to another channel (only if there are packets for some other channel). In addition, the switchable interface stays on a channel for at most *MaxSwitchTime* seconds, before switching to another channel (again, switching happens only if there

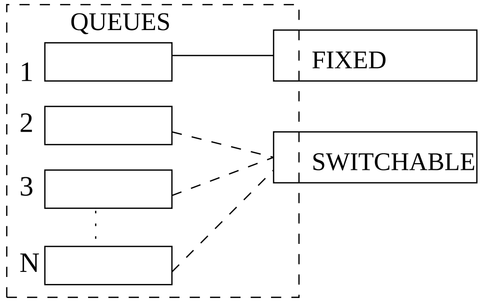

Fig. 5. Illustration of queues associated with interfaces

are packets for some other channel). The two conditions in conjunction ensure that the extra latency introduced by the switching protocol is bounded by *MaxSwitchTime*, while the switching cost is amortized among up to *BurstLength* packets. The parameters *BurstLength* and *MaxSwitchTime* can be suitably set to trade-off latency with throughput. Furthermore, to ensure fairness and to prevent starvation, when switching channels, the switchable interface is set to the channel having the oldest data packet in its queue.

In single channel networks, a packet broadcast on the channel can be received by all neighbors of the transmitter. However, when multiple channels are being used, a packet broadcast on a channel is received only by those nodes listening to that channel. Many higher-layer protocols (e.g., routing protocols) require broadcast packets to be received by all nodes in the neighborhood. Such neighborhood broadcast is supported in our case by transmitting the broadcast packet separately on every channel. A copy of the broadcast packet is added to each channel's queue, and sent out when that channel is scheduled for transmission by the switching protocol.

VII. MCR: MULTI-CHANNEL ROUTING PROTOCOL

In this section, we describe the details of the proposed on-demand Multi-Channel Routing (MCR) protocol that operates over the proposed switching protocol. Popular on-demand routing protocols such as AODV and DSR use the shortest-path metric for route selection. Shortest-path metric assigns a unit cost for each hop in a route, and does not distinguish between a route that uses many channels, and a route that uses few channels. In the proposed architecture, the routing protocol has to consider the cost of interface switching as well. MCR protocol uses a new routing metric that considers the "channel diversity", and "interface switching" costs, in addition to the number of hops in a route.

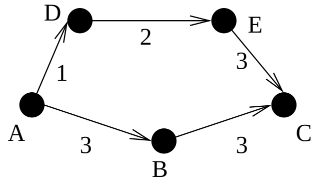

Fig. 6. Need for a diversity cost metric

A. Measuring channel diversity cost

We design a channel diversity cost metric that assigns smaller cost to routes using many channels (called "channel diverse" routes) than routes using few channels. Figure 6 illustrates the need for considering diversity cost. In the figure, node A is setting up a route to node C, and there are two possible routes: A-B-C, and A-D-E-C. Each link is labeled with the channel used to transmit along that link (the channel used on a link is the fixed channel that is being used by the destination of the link). Assume that each link can support a maximum data rate w. When the shortest-path metric is used (as is the case in DSR and AODV), route A-B-C is preferred, as it requires fewer hops than A-D-E-C. However, both links on route A-B-C use channel 3, and at any time only link A-B or link B-C can be active, resulting in a maximum end-to-end throughput of $w/2$. On the other hand, links on route A-D-E-C use different channels, allowing all links to be active simultaneously, resulting in a maximum end-to-end throughput of w. Using the proposed switching protocol, when route A-D-E-C is used, the switchable interface of A is set to channel 1, D to channel 2, and E to channel 3 for sending the first packet, and interface switching is not required for subsequent packets (unless a switchable interface has to switch to another channel for sending packets of a different flow). The availability of two radios allows multiple channels to be used by the switching protocol, provided the routing protocol carefully selects the routes.

We develop the diversity cost metric by noting that a *link* i *in a route is interfered by other links in the route that use the same channel, and are in its interference range.* The interference range is typically assumed to be a small multiple (say, 3) of the communication range. We define a parameter called *InterferenceLength (IL)* (set to 3 in simulations) that is used to identify which links along a route interfere. The i^{th} link is considered to interfere with k^{th} link on a route, for $i{+}1 \le k \le (i{+}IL)$ (we do not consider $k \le i$ to avoid counting links twice), if links i and k use the same channel. Suppose the channel being used by link i is $C(i)$, and suppose there are n links in a path. Then, the diversity cost, DC, of a route is defined to be

$$DC = \sum_{i=0}^{n-1} \sum_{j=i+1}^{max(i\,+\,IL,\;n)} I(C(i) == C(j))$$

where $I(C(i) == C(j))$ is a indicator function that is equal to one when channels being used by link i and link j are the same, else it is 0.

Intuitively, when n interfering links share the same channel, at any time only one of the n links can be active, reducing the path throughput to $1/n^{th}$ of the individual link throughputs. Routes with smaller diversity cost have lesser interference among the links of the route, and have the potential for achieving higher throughputs. The diversity cost is measured as the sum of diversity costs of individual links in the route. An alternate measure is to use the maximum diversity cost of any link as the diversity cost of the route, and this accounts for the *bottleneck* link along the route. However, we found from initial simulations that only tracking the bottleneck link was not effective, as it did not discriminate between two routes that had the same bottleneck link cost, say c, but one route had all other links with cost 1, while the other route had all links with cost c. Higher diversity cost of a link implies higher contention at the MAC layer, and therefore it is more important to reduce the total diversity cost of a route, than to reduce the cost of only the bottleneck link.

B. Measuring interface switching cost

Interface switching is used to enable a small number of interfaces to utilize a large number of channels. The switching protocol switches the switchable interface at the source node to the fixed channel of the destination node. When multiple routes share the same node, and the next-hop node along each route uses a different fixed channel, the switchable interface has to frequently switch from one channel to another. Figure 7 illustrates the need to account for the switching cost. Node B is transmitting data to node E. Node A is setting up a route to node C, with two possible routes: A-B-C, and A-D-C. Both routes A-B-C, and A-D-C have the same diversity cost, and use the same number of hops. However, if route A-B-C is chosen, node B has to frequently switch between channels 2 and 3 when sending data to node E and node C respectively. Thus, frequent switching incurs a *switching overhead*, and the throughput over both flows A-B-C and B-E reduces, because the switchable interface on B becomes a bottleneck to performance.

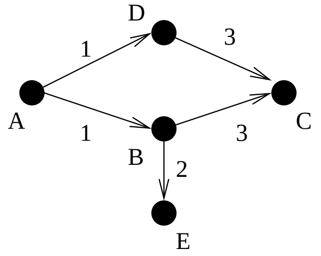

Fig. 7. Need for a switching cost metric

We set the switching cost for using a fixed channel to be 0, as interface switching is not required for fixed channels. For computing the switching cost of switchable channels, we define the notion of an *active channel*. Intuitively, *active channel* is a channel that is already being used for communication, and has a switchable interface assigned to it most of the time. If a new route that is being discovered requires the use of a channel that is not active, then supporting the new route may require switching between the active channel and the new channel. For identifying the active channel of a node, we associate each switchable channel i with a parameter called the *ChannelUsageFraction(i), CUF(i)* that indicates the fraction of total traffic sent by the node, using the switchable interface, over channel i. We use a smoothening factor α to update the *ChannelUsageFraction* for all channels j each time a packet (including broadcast packets) is sent on any channel i as follows.

$$\textit{CUF(j)} \;=\; \alpha * \textit{CUF(j)} + (1-\alpha) * I(j == i)$$

where I(j == i) is a indicator function that is equal to 1 if $j = i$, else is equal to 0. Parameter α maintains a weighted history of channel usage. We set α to 0.9 in the simulations, which corresponds to keeping a channel usage history of last 10 packets sent by the node.

A channel i is defined to be an *active channel* if *ChannelUsageFraction(i)* is more than a specified threshold *ChannelUsageThreshold* (which we set to 0.5 in simulations). When there is no active traffic going through a node, then all switchable nodes will have an equal (and small) *ChannelUsageFraction* that is below the usage threshold. Hence, it is possible that at some point a node has no *no active channels*. However, when one channel starts being used continuously for transmitting data, the *ChannelUsageFraction* of that channel rises above the threshold, and the channel is marked as a *active channel*. On the other hand, once an *active channel* becomes idle, it is marked as *inactive* after approximately $1/(1-\alpha)$ number of packets are sent out from the node on other channels (around 10 packets in our simulations). It is also possible to use a timeout, in addition to usage history, to identify when an active channel has turned inactive.

The switching cost of using a link X-Y along a route is defined as follows.

- If fixed channel of Y, say i, is an active or fixed channel of X, then switching cost of link X-Y is zero.
- Else, if X has no active channels, then the switching cost of link X-Y is zero.
- Else, switching cost of X-Y is equal to (*switchingDelay* / *estimatedPacketTransmissionTime*).

Here, *switchingDelay* is the time required for switching an interface from one packet to another, and *estimatedPacketTransmissionTime* is a rough estimate of the time required to transmit an average-sized data packet over a channel. In our simulations, we have chosen to use a pre-computed constant for *estimatedPacketTransmissionTime* assuming a 54 Mbps channel rate (IEEE 802.11a peak rate), and a 1000 byte packets. It is possible to use a more adaptive estimation technique based on available bandwidth measurements and average size of packets seen on the channel. However, our simulations suggested that the switching cost metric does not require an accurate estimation of the packet transmission time, and therefore we choose to use a simple pre-computed constant.

The switching cost of a link represents the delay, with respect to packet transmission times, of choosing that link. Intuitively, the impact of frequent switching may be viewed as a *lengthening of the route* because the switching delay manifests itself as *virtual hops* along the route that add to the path RTT. By using this representation for the switching cost, it can be easily integrated with the shortest-path cost metric. If the switching delay is large, the *route lengthening* on account of frequent switching is more, whereas when the switching delay is small, the *route lengthening* on account of frequent switching is small. Thus, the switching cost metric can account for a wide range of switching delays.

C. Combined routing metric

The routing metric we propose integrates the shortest-path metric with the diversity cost, and the switching cost. Shortest-path metric attempts to minimize the resources, in terms of nodes, used by a route. Diversity cost metric helps minimize the cost of not using the available channel diversity. Switching cost metric helps minimize the cost of frequent interface switching. The combined routing metric that we use is a *weighted linear combination of total hop count, total diversity cost, and total switching cost.* In our simulations, we have weighted all three components equally, and it is part of our future work to study the impact of different weights on the performance of the routing protocol.

D. Route discovery and route maintenance

The proposed MCR protocol is designed to be a source-initiated on-demand routing protocol, similar to DSR. The route discovery process is initiated by a source node, which broadcasts a Route Request (RREQ) packet over all channels. Each new route discovery initiated by a node uses a unique sequence number which is included in all RREQ packets. The RREQ packet sent over a channel i at a node X, contains the channel index i, as well as the switching cost of using channel i at node X. Intermediate nodes can compute the cost of a RREQ using the information included in the RREQ (diversity cost is computed based on the list of channels along the path; the switching cost is just the sum of all the link switching costs included in the RREQ). When a RREQ packet is received at an intermediate node, it re-broadcasts the request (after adding the channel index, and the switching cost) in the following two cases:

- The sequence number in the RREQ is being seen for the first time. In this case, the cost of the already traversed path is stored in a table, or
- The cost of the already discovered path in the RREQ is smaller than the cost seen in all earlier RREQs, if any, with the same sequence number.

A lower cost RREQ may traverse a longer path, and reach an intermediate node after a higher cost RREQ has been forwarded. The, second condition is required to ensure the least cost path is discovered.

When the destination receives a RREQ, it responds with a route reply (RREP) only if the cost of the received RREQ is smaller than other RREQs (containing the same sequence number) seen till then. This ensures that high cost paths are not unnecessarily sent back to the source node. The source node always uses the least cost route received from the destination for routing data packets.

Route maintenance involves two components. The first component, called “Route Refresh”, periodically initiates a new route discovery, even if a route is not broken, to update the costs of known routes. This mechanism ensures that the route cost information is never stale, and new lower cost routes, if any, are discovered. The second component, called “Route Recovery”, is used to repair broken routes. When a route breakage is discovered,

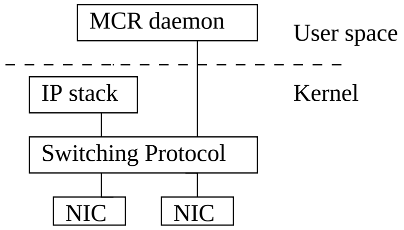

Fig. 8. Proposed architecture

route error (RRER) message is sent back to the source, and a new route discovery is initiated. We have not used optimizations such as route caches, and intermediate route repair, though it is possible to extend the protocol with those optimizations.

E. Overall architecture

In the previous sections, we have presented the details of the proposed switching and routing protocol. We now describe the architecture that can be used for implementing the proposed protocols. Figure 8 outlines the proposed architecture. The switching protocol can be implemented at the link layer and communicate with the network interface cards through the appropriate device drivers. The switching protocol maintains a neighbor table that contains information about the fixed interface being used by neighbors. The MCR routing protocol can be implemented as a daemon in the user space. The hello protocol can be implemented in the MCR daemon. The interaction between the MCR daemon and the switching protocol allows updating of the neighbor table, and enables the daemon to obtain information necessary to compute the switching cost. The switching layer chooses the channel to be used for transmitting a packet received from the network layer based on the next-hop address. If necessary, the switching protocol can change the channel of the switchable interface, before handing off a packet for transmission. It is part of our ongoing work to implement and evaluate the architecture in a Linux-based testbed.

VIII. EVALUATION

We have simulated the proposed architecture in Qualnet version 3.6 [39]. We added a layer above the MAC to implement the switching protocol, and the MCR protocol was implemented at the network layer. No modifications were required to the IEEE 802.11 MAC layer. In all simulations, nodes in the network were assumed to be equipped with two IEEE 802.11a interfaces. In our evaluations of MCR, we have varied the number of orthogonal channels from 2 to 5. All simulation results are run for 100 seconds. Unless otherwise stated, the interface switching delay is assumed to be 100 microseconds. The application data packet size for CBR and FTP traffic is set to 1500 bytes. The CBR bit-rates are always chosen to be large enough to saturate the network.

As discussed earlier, existing multi-channel, multi-interface proposals are either not designed for mobile ad hoc networks [34], or assume that the number of interfaces is equal to the number of channels [32]. Therefore, a fair comparison of existing proposals with MCR was not possible. Instead, we compared the performance of MCR with the performance of DSR protocol when using a single channel, to quantify the benefits of using multiple channels.

A. Performance of switching protocol

We first evaluate the performance of the proposed approach in simple chain topologies. The length of a chain is varied from 1 to 9 hops. A CBR flow is setup from the first node to the last node of the chain. We set the data rate of all channels to 54 Mbps, the maximum rate possible with IEEE 802.11a. Nodes in a chain are stationary, and direct communication is possible only between adjacent nodes on the chain (distance between adjacent nodes is 40m). Furthermore, all nodes in the chain are in the carrier sense range of each other. Hence, in this scenario, on each channel there can be at most one transmission going on at any time. This scenario tests the effectiveness of the switching protocol (routing metric is not tested here, as there is a single route between source and destination)

Figure 9 compares the throughput obtained with DSR using a single channel (curve labeled “DSR”), with the throughput obtained with the proposed MCR protocol when the number of channels is varied from 2 to 5. As we can see from the figure, the throughput of DSR rapidly degrades when the number of hops along a chain increase. The throughput of single channel, single interface protocols degrades because of two reasons. Firstly, intermediate nodes cannot simultaneously receive and forward data, cutting the achievable throughput by half. Secondly, since a single channel is used, transmissions on a hop will inhibit other transmissions on other hops that are within the carrier sense range, thereby further degrading the achievable throughput.

When multiple channels and multiple interfaces are used in MCR, the switching protocol assigns the fixed channel of successive nodes along the chain to different channels. Also, when an intermediate node is receiving

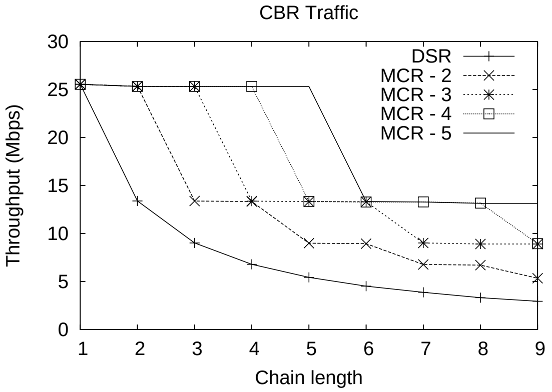

Fig. 9. CBR Throughput in chain topologies

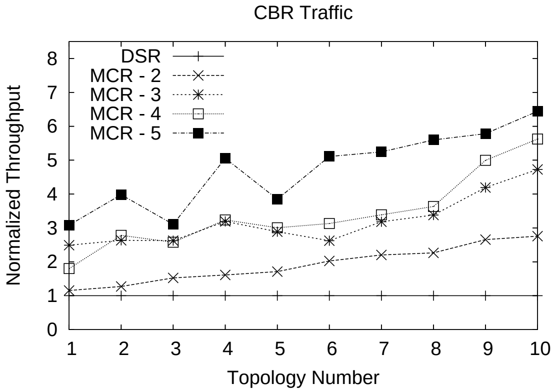

Fig. 10. CBR Throughput in a random topology

on the fixed interface, it can simultaneously forward data to the next node using the switchable interface. Consequently, MCR offers higher throughput over longer chain lengths. However, when the chain has only one hop, MCR uses a single channel only (the fixed channel of the destination), and hence achieves the same throughput as DSR. Over longer chains, MCR can better utilize the multiple channels. For example, in Figure 9, the throughput of MCR with 5 channels ("MCR 5" curve) stays the same for chains of length 1 to 5, because successive hops use different channels. When the chain length goes beyond 5, two hops along the chain have to be on some common channel, thereby degrading the throughput.

The key observation from Figure 9 is that multiple channels can significantly improve throughput in multi-hop scenarios. Furthermore, even a few interfaces (2 in this example), can utilize multiple channels (up to 5 channels in this example).

B. Performance of MCR routing protocol

In this section, we evaluate the performance of MCR in random topologies, with mobility. 50 nodes are placed in a 500m X 500m area. Nodes move using the random way-point mobility model, with both the maximum and the minimum speeds[2] set to 10m/s. 5 flows (either CBR or FTP) are setup between randomly selected pair of nodes. All results are plotted for 10 different random topologies. Since the aggregate throughput obtained depends on the topology, we normalize all results with the throughput obtained when using DSR on a single channel. The normalized throughput clearly quantifies the performance improvement obtained when using MCR.

Figure 10 compares the throughput of DSR with MCR when using CBR traffic. The topologies are numbered from 1 to 10 in the increasing order of normalized throughput obtained when using 2 channels ("MCR 2" curve), and the same labeling is used for all graphs in this section. The throughput of MCR with 2 channels varies from 1.2 times of DSR to 2.75 times of DSR depending on the topology. Similarly, for MCR with 5 channels the throughput varies from 3 times to 6.5 times that of DSR.

The improvement obtained with MCR strongly depends on the underlying topology. If multiple routes are available between a pair of nodes, then MCR will choose a good *channel diverse* route, thereby utilizing the available channels. Otherwise, multiple routes are not available, then MCR is forced to use the available route, and the throughput improvement is less. Therefore, MCR is well-suited for higher density networks that may have multiple routes between a source and a destination. In some scenarios, MCR with k channels can offer more than a k fold improvement in performance by distributing load over multiple channels, thereby reducing contention overhead on any single channel.

In general, the results suggest that MCR can offer significant improvements even when using only two interfaces. Furthermore, the simulations involve moderate mobility, which indicates the suitability of MCR protocol in mobile ad hoc networks.

Figure 11 plots the normalized end-to-end delay of MCR, with respect to DSR, for different topologies. As we can see from the figure, by using multiple channels, MCR substantially reduces the end-to-end delay.

Figure 12 evaluates the performance of MCR with FTP traffic (sent over TCP). As we can see from the figure, MCR significantly improves the network capacity

---

[2] Random way-point based simulations have been shown to not stabilize when the maximum and minimum speeds are widely different.

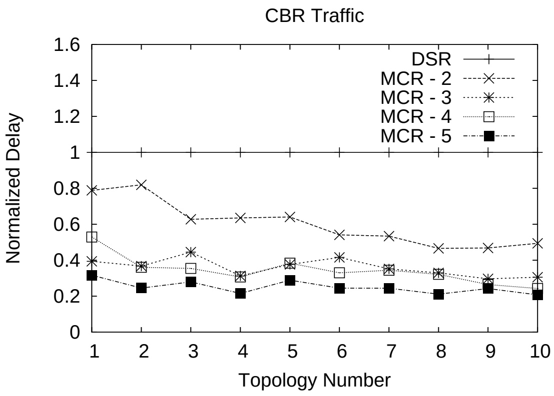

Fig. 11. CBR Delay in a random topology

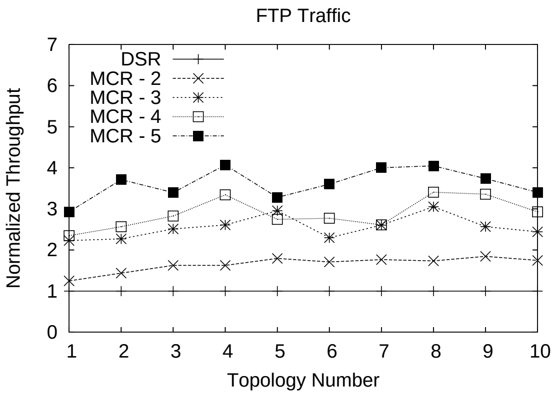

Fig. 12. FTP Throughput in random topologies

with TCP flows as well. The maximum performance improvement for FTP flows is smaller than the maximum performance improvement observed with CBR flows. For example, with 5 channels, CBR had maximum normalized throughput of 6.5, while FTP has a maximum improvement of 4.1. However, the minimum normalized throughputs are comparable (3 for CBR versus 2.9 for FTP).

TCP uses ACK-feedback from the destination to control the sending rate. The ACK packets form a *reverse traffic* from the destination to the source. The ACK traffic in the reverse direction may lead to more frequent switching if the same path is used for forward and reverse traffic, thereby reducing the maximum performance improvement. For example, consider an intermediate node X on a route, which has as previous hop node W, and as next hop node Y. When X is transmitting forward traffic to Y, it has to switch to the fixed channel of Y. On the other hand, when X is transmitting the reverse TCP ACK traffic to W, it has to switch to the fixed channel of W, potentially leading to frequent channel switching if W and Y use different fixed channels.

To mitigate this problem, MCR separately selects the (reverse) route from the destination to the source. Single channel protocols such as DSR reverse an existing route to transmit reverse traffic. In contrast, MCR does not cache the forward route at the destination, and therefore selects a new route for reverse traffic. Separately selecting the reverse route ensures that the chosen reverse route has the least cost from the destination to the source, and is possibly disjoint from the forward route. In some cases, when multiple disjoint paths are unavailable, the path used by reverse traffic will not be disjoint from the path used for forward traffic, degrading TCP throughput. We believe this is the reason for lower maximum performance improvement with FTP over CBR. If two switchable interfaces are available, in addition to a fixed interface, then both the forward and reverse traffic can use the same path without requiring interface switching.

C. *Impact of varying network contention*

In this section, we evaluate the impact of varying the number of flows in the network. 50 nodes are randomly placed in a 500m square area. CBR flows are set up between randomly selected pair of nodes. The number of CBR flows is varied between 1 to 15. Figure 13 compares the throughput of MCR with DSR. As the number of flows increases, MCR offers significantly better performance than DSR, especially when more channels are available. For example, MCR with 5 channels offers 1.5 times the throughput of DSR with 1 flow, but offers around 8 times DSR throughput with 12 flows.

When the number of flows is small, the throughput improvement with MCR depends on the channel diversity available on the best route between the source and the destination. Since the simulations involve mobility, the best route at different periods of time may offer different degrees of channel diversity. As a result, the full benefits of using a large number of channels (say, 5) is not realized when the number of flows is small. When the number of flows is large, at any point of time at least a subset of flows can utilize the available channel diversity. Furthermore, increasing the number of flows in the network increases the average contention at the MAC layer. When multiple channels are available, the fixed channels of various nodes are distributed across the available channels. Since the number of nodes using a specific channel decreases, MAC layer contention on

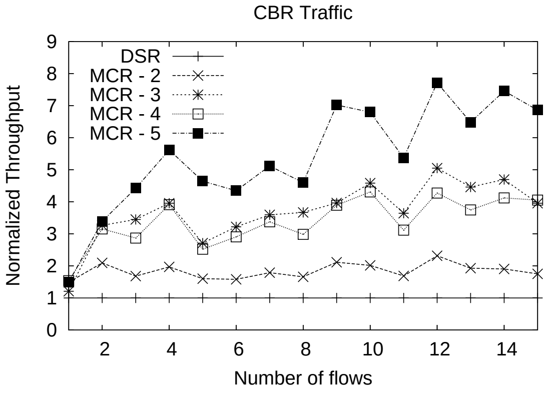

Fig. 13. CBR Throughput with varying number of flows

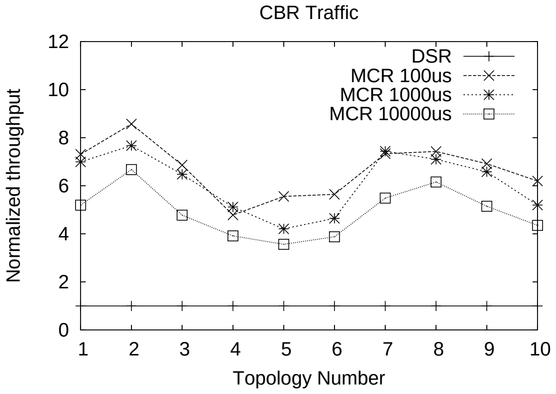

Fig. 14. CBR Throughput with varying switching delay

each channel reduces. As a result, with large number of flows, MCR with multiple channels offers significant performance improvement over DSR. Thus, MCR uses all the available channels to provide better scalability, with increasing network contention, than a single channel solution.

D. *Impact of switching delay*

In this section, we evaluate the impact of interface switching delay on the performance of the MCR protocol. We use the 10 random topologies described earlier. The impact of switching delay is more evident when there are more contending flows. Hence, we set up 15 CBR or FTP flows between randomly selected pair of nodes. Simulation results are for MCR protocol using 5 channels, and switching delay is varied from 100 microseconds to 10,000 microseconds (i.e., 10 milliseconds).

Figure 14 plots the throughput of MCR with CBR traffic, normalized with respect to DSR, over 10 random topologies. As we can see from the figure, the throughput obtained degrades when the switching delay increases. However, the degradation is not significant, and MCR continues to offer a large performance improvement over a single channel solution. The routing protocol attempts to find routes that do not require frequent switching. A route that requires frequent switching is chosen only if there is no other route with lower cost. Carefully considering the switching cost ensures minimal degradation in throughput even with large switching delays.

Figure 15 plots the normalized throughput of MCR with FTP traffic (sent over TCP). Higher switching delay degrades FTP throughput. However, the throughput degradation is minimal even with moderate delay (1000 microseconds). With very high switching delay (10 milliseconds), the throughput degradation is significant. When a route that requires frequent switching is used with TCP traffic, the path RTT increases. TCP throughput is inversely proportional to the RTT of the path, and therefore TCP throughput degrades with higher RTT. As long as the fraction of path RTT contributed by switching delay is small, switching delay has minimal impact on throughput (e.g., delays up to 1000 microseconds). When the switching delay starts contributing to a larger fraction of the path RTT, there is a more pronounced impact on throughput.

Some of the performance degradation with larger switching delays is due to the switching required when transmitting a broadcast packet on all channels. In our proposal, broadcast packets are used for route request and hello packets. Thus, the frequency of route refreshing (that is used to update route costs), and the frequency of hello packet exchange have to be tuned based on the the magnitude of the switching delay. However, as the simulation results demonstrate, even with fairly large switching delays, our proposed architecture offers significant performance improvement. Therefore, we conclude that it is possible to effectively utilize all the available channels, even if the number of interfaces is small, and the switching delay is large.

IX. DISCUSSIONS AND FUTURE WORK

In this paper, we have not considered the problem of identifying the optimal number of fixed and switchable interfaces to use, when more than two interfaces are available. In our architecture, one fixed interface is always required to allow neighbors of a node to communicate with it. However, whether multiple fixed interfaces are beneficial depends on the traffic being forwarded by a node. For example, if M interfaces are

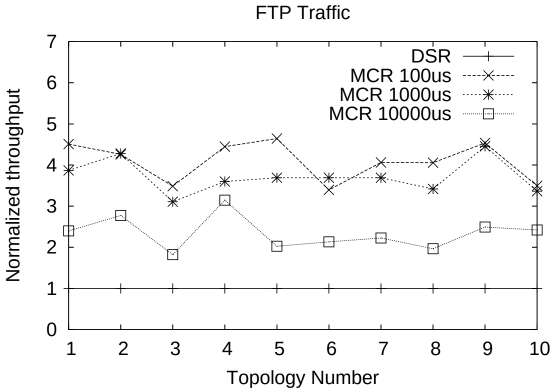

Fig. 15. FTP Throughput with varying switching delay

available, some K of the M interfaces can be chosen to be fixed interfaces. The fixed interfaces are mostly used for receiving data, while the switchable interfaces are mostly used for sending data. So, one choice for K will be to set it to approximately $M/2$ if the node receives and forwards nearly equal amounts of data. However, if a node is mostly a source (destination) of traffic, and therefore mostly transmits (receives) data, then it is better to use a smaller (larger) value of K. It is part of our future work to develop a mechanism for dynamically changing the number of fixed interfaces at a node, based on the amount of data being transmitted and received by the node.

The routing metric proposed in this paper does not currently consider the data rate supported by a channel. If the proposed MCR protocol is used with heterogeneous channels (for example, some channels are high data rate IEEE 802.11a channels, while others are low data rate IEEE 802.11b channels), then the diversity cost computation has to explicitly account for the data rate of a channel. Also, channels in different frequency bands may have different transmission ranges. For example, it may be appropriate to use a route with two hops that share a high data rate channel, than to use a route with a single hop, but over a low data rate channel. Draves et al. [32] have proposed a routing metric, WCETT, that accounts for heterogeneous channels, but does not consider the cost of interface switching. Furthermore, WCETT may not be suitable with node mobility [40]. It is part of our future work to combine WCETT with our proposed routing metric, to develop a new metric that accounts for channel heterogeneity, as well as node mobility and the cost of interface switching.

Multiple channels may be used to derive other benefits. For example, we have proposed to use a single-path routing algorithm. In single channel networks, multi-path routing algorithms are often not effective as the chosen paths have to be *interference-disjoint* (i.e., the paths should not interfere on the wireless channel), and it is often difficult to find such paths. On the other hand, if multiple channels are available, then it is sufficient for the paths to be *node-disjoint*, as it may be possible to select routes that use different channels. When the node density is high, the number of node-disjoint paths may be large, while the number of interference-disjoint paths is still small. Hence, multiple channels may simplify the use of multi-path routing algorithms.

X. CONCLUSION

In this paper, we have studied the problem of designing protocols for multi-channel wireless networks wherein the number of interfaces per node is smaller than the number of channels. We have analyzed the capacity of multi-channel networks. We have also developed link layer and routing protocols for multi-channel multi-interface ad hoc wireless networks. We have proposed an interface assignment strategy that uses the notion of *fixed* and *switchable* interfaces. The interface assignment strategy utilizes all the available channels even when the number of interfaces is smaller than the number of available channels. We have presented a routing protocol that selects high-throughput routes in multi-channel, multi-interface networks. The routing protocol uses routing metrics that account for channel diversity, and the cost of switching interfaces. It is part of our ongoing work to implement the proposed architecture in a Linux-based testbed.

REFERENCES

[1] *IEEE Standard for Wireless LAN-Medium Access Control and Physical Layer Specification, P802.11*, 1999.

[2] Paramvir Bahl, Atul Adya, Jitendra Padhye, and Alec Wolman, “Reconsidering Wireless Systems with Multiple Radios,” *ACM Computing Communication Review*, July 2004.

[3] Piyush Gupta and P. R. Kumar, “The Capacity of Wireless Networks,” *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, March 2000.

[4] Matthias Grossglauser and David Tse, “Mobility Increases the Capacity of Ad-hoc Wireless Networks,” in *Infocom*, 2001.

[5] Pradeep Kyasanur and Nitin H. Vaidya, “Capacity of Multi-Channel Wireless Networks: Impact of Number of Channels and Interfaces,” Tech. Rep., University of Illinois at Urbana-Champaign, March 2005.

[6] Nikhil Bansal and Zhen Liu, “Capacity, Delay and Mobility in Wireless Ad-Hoc Networks,” in *Infocom*, 2003, pp. 1553–1563.

[7] Eugene Perevalov and Rick Blum, “Delay Limited Capacity of Ad hoc Networks: Asymptotically Optimal Transmission and Relaying Strategy,” in *Infocom*, 2003.

[8] Abbas El Gamal, James Mammen, Balaji Prabhakar, and Devavrat Shah, “Throughput-Delay Trade-off in Wireless Networks,” in *Infocom*, 2004.

[9] Benyuan Liu, Zhen Liu, and Don Towsley, “On the Capacity of Hybrid Wireless Networks,” in *Infocom*, 2003.

[10] Ulas C. Kozat and Leandros Tassiulas, “Throughput Capacity of Random Ad Hoc Networks with Infrastructure Support,” in *Mobicom*, 2003.

[11] A. Agarwal and P. R. Kumar, “Capacity bounds for ad-hoc and hybrid wireless networks,” *ACM SIGCOMM CCR*, vol. 34, no. 3, pp. 71–81, July 2004.

[12] Su Yi, Yong Pei, and Shivkumar Kalyanaraman, “On the Capacity Improvement of Ad Hoc Wireless Networks Using Directional Antennas,” in *Mobihoc*, USA, 2003, pp. 108–116.

[13] Rohit Negi and Arjunan Rajeswaran, “Capacity of power constrained ad-hoc networks,” in *Infocom*, 2004.

[14] Honghai Zhang and Jennifer C. Hou, “Capacity of Wireless Adhoc Networks under Ultra Wide Band with Power Constraint,” in *Infocom*, 2005.

[15] Jiandong Li, Zygmunt J. Haas, , and Min Sheng, “Capacity Evaluation of Multi-Channel Multi-Hop Ad Hoc Networks,” in *International Conference on Personal Communications*, 2002.

[16] Michael Gastpar and Martin Vetterli, “On the Capacity of Wireless Networks: The Relay Case,” in *Infocom*, New York, USA, June 2002, pp. 1577–1586.

[17] Stavros Toumpis and Andrea J. Goldsmith, “Large Wireless Networks under Fading, Mobility, and Delay Constraints,” in *Infocom*, 2004.

[18] Stavros Toumpis, “Capacity Bounds for Three Classes of Wireless Networks: Asymmetric, Cluster, and Hybrid,” in *Mobihoc*, 2004.

[19] A. Nasipuri, J. Zhuang, and S.R. Das, “A Multichannel CSMA MAC Protocol for Multihop Wireless Networks,” in *WCNC*, September 1999.

[20] A. Nasipuri and S.R. Das, “Multichannel CSMA with Signal Power-based Channel Selection for Multihop Wireless Networks,” in *VTC*, September 2000.

[21] N. Jain, S. Das, and A. Nasipuri, “A Multichannel CSMA MAC Protocol with Receiver-Based Channel Selection for Multihop Wireless Networks,” in *IEEE International Conference on Computer Communications and Networks (IC3N)*, October 2001.

[22] Shih-Lin Wu, Chih-Yu Lin, Yu-Chee Tseng, and Jang-Ping Sheu, “A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks,” in *International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN)*, 2000.

[23] Wing-Chung Hung, K.L.Eddie Law, and A. Leon-Garcia, “A Dynamic Multi-Channel MAC for Ad Hoc LAN,” in *21st Biennial Symposium on Communications*, Kingston, Canada, June 2002, pp. 31–35.

[24] Jungmin So and Nitin H. Vaidya, “Multi-channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals using a Single Transceiver,” in *Mobihoc*, 2004.

[25] Atul Adya, Paramvir Bahl, Jitendra Padhye, Alec Wolman, and Lidong Zhou, “A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks,” in *IEEE International Conference on Broadband Networks (Broadnets)*, 2004.

[26] P. Bahl, R. Chandra, and J. Dunagan, “SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks,” in *ACM Mobicom*, 2004.

[27] Marco Ajmone Marsan and Fabio Neri, “A Simulation Study of Delay in Multichannel CSMA/CD Protocols,” *IEEE Transactions on Communications*, vol. 39, no. 11, pp. 1590–1603, November 1991.

[28] N. Shacham and P. King., “Architectures and Performance of Multichannel Multihop Packet Radio Networks,” *IEEE Journal on Selected Area in Communications*, vol. 5, no. 6, pp. 1013–1025, July 1987.

[29] Jungmin So and Nitin H. Vaidya, “A Routing Protocol for Utilizing Multiple Channels in Multi-Hop Wireless Networks with a Single Transceiver,” Tech. Rep., University of Illinois at Urbana-Champaign, October 2004.

[30] David B. Johnson, David A. Maltz, and Yih-Chun Hu, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR),” *Ietf Manet Working Group (Draft 10)*, 2004.

[31] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing,” in *Ietf RFC 3561*, July 2003.

[32] Richard Draves, Jitendra Padhye, and Brian Zill, “Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks,” in *ACM Mobicom*, 2004.

[33] Ashish Raniwala, Kartik Gopalan, and Tzi-cker Chiueh, “Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks,” *Mobile Computing and Communications Review*, vol. 8, no. 2, pp. 50–65, April 2004.

[34] Ashish Raniwala and Tzi-cker Chiueh, “Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network,” in *Infocom*, 2005.

[35] Pradeep Kyasanur and Nitin H. Vaidya, “Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks,” in *WCNC*, 2005.

[36] Pradeep Kyasanur and Nitin H. Vaidya, “Routing in Multi-Channel Multi-Interface Ad Hoc Wireless Networks,” Tech. Rep., University of Illinois at Urbana-Champaign, December 2004.

[37] Ranveer Chandra, Paramvir Bahl, and Pradeep Bahl, “MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card,” in *IEEE Infocom*, Hong Kong, March 2004.

[38] “Maxim 2.4 GHz 802.11b Zero-IF Transceivers,” http://pdfserv.maxim-ic.com/en/ds/MAX2820-MAX2821.pdf.

[39] Scalable Network Technologies, “Qualnet simulator version 3.6,” http://www.scalable-networks.com.

[40] Richard Draves, Jitendra Padhye, and Brian Zill, “Comparison of Routing Metrics for Multi-Hop Wireless Networks,” in *SIGCOMM*, 2004.