On the Capacity Region of Multi-Radio Multi-Channel Wireless Mesh Networks

Murali Kodialam Thyaga Nandagopal

Bell Laboratories, Lucent Technologies
101 Crawfords Corner Road
Holmdel, NJ 07733, USA
{muralik, thyaga}@bell-labs.com

***Abstract*— Next generation fixed wireless broadband networks are being increasingly deployed as mesh networks in order to provide and extend access to the internet. These networks are characterized by the use of multiple orthogonal channels and nodes with the ability to simultaneously communicate with many neighbors using multiple radios (interfaces) over orthogonal channels. Networks based on the IEEE 802.11a/b/g and 802.16 standards are examples of these systems. However, due to the limited number of available orthogonal channels, interference is still a factor in such networks. Recent results have attempted to characterize the achievable capacity of such networks, both in the deterministic[1] and asymptotic[2] sense. In this paper, we build on these previous works and provide an experimental verification of asymptotic capacity result in [2] by using the deterministic model first proposed in [1]. Our simulation results are based on a novel *joint channel assignment and scheduling* algorithm which is shown to achieve near-optimal capacity limits in random network topologies. The results in this paper are the first to provide an experimental verification of theoretical results in [2] and give an insight into the behavior of the capacity region of a multi-hop multi-radio mesh network.**

I. Introduction

Recent research and commercial efforts in the wireless networking industry have focused on a new class of ad hoc networks: *mesh networks*. These networks are characterized by a set of fixed nodes with multiple wireless interfaces utilizing multiple orthogonal channels over a given swath of spectrum. IEEE 802.11 a/b/g [3] and 802.16 [4] are both examples of standards which are used to operate such networks. Mesh networks are being deployed as a solution to extending the reach of the last-mile access to the internet, using a multi-hop configuration. One of the popular deployment methods is to use one standard, for e.g., IEEE 802.16, for back-hauling traffic on the multi-hop wireless relay backbone and to use another standard such as 802.11 a/b/g to carry traffic over the last-hop to the user, as shown in Figure 1. This ensures that traffic on the wireless backbone is isolated from the fluctuating load and interference from the last-hop end-users. We are concerned with the capacity of the wireless backbone among multiple mesh network nodes.

While there can be multiple orthogonal channels for a mesh network to operate upon, the number of such channels may not be enough to prevent interference-free operation in a given neighborhood. In addition, each mesh network node may have multiple radio transceivers which allows it to communicate, simultaneously with more than one neighbor at the same time using different channels, with each radio interface tuned to a different channel.

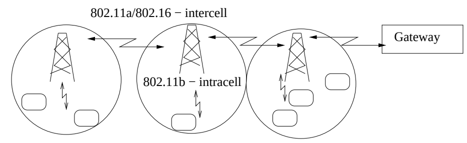

Fig. 1. Network Example

Full duplex operation is possible at each node, i.e., a node can be receiving from or transmitting to a neighbor i on channel A, while transmitting to or receiving from neighbor j on channel B, where $A \neq B$. At different time slots, a node can tune its radio interfaces to different channels. There is a switching penalty associated with this switching, typically one or more time-slots. In this paper, we assume that this switching overhead is potentially negligible. for purposes of verifying the achievable capacity. A similar assumption is made in [1], [2]. In the rest of this paper, we refer to such fixed multi-channel multi-hop wireless networks with multiple radios per node as *MC-MR* networks.

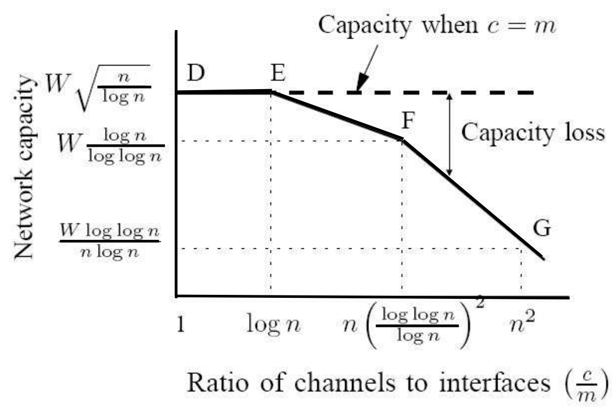

Fig. 2. Asymptotic Capacity

The above figure is taken from [2] and describes the asymptotic capacity of a random network.

As in any network, one of the principal questions that arise here is regarding the achievable capacity of the mesh network. In a recent result, Kyasanur and Vaidya [2] have theoretically shown that the asymptotic capacity of a MC-MR network with c orthogonal channels and m radio interfaces per node is

$O(W\sqrt{\frac{n}{\log n}})$ when the ratio $c/m = O(\log n)$, as shown in Figure 2. W is the aggregate maximum data rate of all the channels. As the ratio c/m increases, of channels to radios becomes larger, the achievable capacity of the network decreases. One of the challenges in trying to make use of such asymptotic results is to figure out the hidden constants involved in the $O(...)$ notation. For example, in Figure 2, if the $O(\log n)$ function for the c/m ratio has a large associated constant, then a network designer can achieve the best possible capacity with a smaller number of radio interfaces per node, instead of trying to have as many radios as there are channels.

One of the main goals of this paper is to identify this constant for practical mesh network instances. In addition, we also want to identify a viable scheme that uses routing, channel assignment and scheduling to achieve the best possible capacity with a given number of radios for a particular instantiation of the network.

We address these key capacity planning questions, and propose an algorithm to jointly optimize the routing, link channel assignments and scheduling for such networks in order to obtain *upper and lower bounds for the capacity region* under a given objective function, which helps us to identify the capacity constants associated with large mesh network deployments.

We use the network model proposed recently by us in [1] to characterize the various aspects of a multi-hop mesh network with limited number of orthogonal channels and with multiple radios at each node. This model gives both *necessary and sufficient conditions* for a feasible channel and time schedule in the network, using the protocol model of interference [5]. This model can be used to compute upper and lower bounds on the feasible rate vector in a network where all nodes have data to send.

Our Contributions: Our contributions are as follows.

1) We propose a greedy link channel assignment and scheduling algorithm which gives an achievable lower bound on the capacity. We show that this algorithm performs within 90% of the capacity upper bound in practice. Note that computing the optimal schedule is a well-known NP-hard problem, with no known performance bounds even for the simple case of 1 channel and 1 radio per node for general graphs.
2) We evaluate the achievable capacity for many mesh network topologies with up to a 100 nodes. Based on our results, we show that the constants associated with the asymptotic capacity results in [2] are very small, i.e., one needs nearly as many radios per node as there are orthogonal channels in order to make effective use of multiple channels. Along the way, we also provide experimental confirmation of the results in [2].

The rest of the paper is organized as follows. In Section II, we present a summary of relevant related work in this area, and describe the basic network model, notations and design considerations used in this paper. We describe the constraints characterizing the *MC-MR* network model, describe the optimization functions and obtaining the upper bounds in Section III. We present the link channel assignment and scheduling algorithm in Section IV to derive the lower bounds. Section V describes our evaluation using simulations. The paper concludes with a discussion of issues in Section VI.

II. Background and Assumptions

We first briefly summarize the related work in mesh networks, and then proceed to describe our network model.

A. Related Work

There are numerous studies [6]-[19] that consider some combination of the routing, channel assignment and scheduling problems for wireless mesh networks. Some study the problem of finding efficient routes to maximize throughput [8], [9], [10], [11], [13], while some consider only channel assignment and scheduling [12], [18], [19] and others consider both routing and scheduling [6], [7], [14], [15], [16], [17]. All of these previous studies consider a subset of the capacity problem, with most of the papers only addressing the question of how to improve throughput compared to some other algorithm. We think that it is more important to seek the limits of the throughput performance first, so that we can quantify the performance of such algorithms.

Gupta and Kumar [5] discuss the problem of *asymptotic capacity* of a multi-hop wireless network under two different interference models: (a) the protocol interference model, and (b) the physical interference model. In [20], the asymptotic capacity of a relay network is shown to be $O(\log n)$. These studies assume a single channel, single radio-per-node-network, although it is possible to extend the results to multiple channels and multiple radios-per-node. There have been various other studies regarding asymptotic capacity of multi-hop wireless networks [21], [22], [23]. However, the most recent and complete work is reported in [2] by Kyasanur and Vaidya, where the asymptotic capacity of MC-MR networks is derived using similar methods as in [5]. We will use the simulation results from our algorithms to verify our conclusions with these results.

The work that is related to the network model proposed in this paper is the model in [1]. We present an overview of this modelling in subsequent sections, and refer the reader to consult [1] for further details.

A parallel work in [24] attempts to solve the throughput maximization problem for a MC-MR network, where the network is restricted to be a superset of a disk graph, i.e., the interference range is assumed to be a fixed multiple of the communication range. In addition, radios are assumed to be unable to switch dynamically between multiple channels in [24].

B. Network Model and Design Considerations

We start with the underlying network model, and explain the definitions and terminology used in this paper. We also explain some design choices made in the paper.

1) Basics: We consider a fixed multi-hop wireless network with n nodes. We represent the network with a directed graph $G = (V, E, E^{\mathcal{I}})$ where V represents the set of nodes in the network, E the set of directed links that can carry data (*data links*), and $E^{\mathcal{I}}$ denotes the set of directed links that indicate interference (*interference links*), but cannot carry data. We assume that data links and interference links are bi-directional. This is a consequence of our choice of the protocol interference model described later.

If node u can transmit *directly* to node v (and vice versa), then we represent this by a link, $u \leftrightarrow v$, between node u and

V	set of vertices	n	number of nodes $\|V\|$
E	set of data links	$E^{\mathcal{I}}$	set of interference links
G	network graph	τ	length of a time slot
$t(e), e \in E$	transmitting node of link e	$h(e), e \in E$	receiving node of link e
OC	set of orthogonal channels, $\|OC\| = \mathcal{C}$	$\kappa(v)$	number of radios at node v
$c_i(e)$	capacity of channel i over link e	$f_i(e)$	flow rate of channel i over link e
$g_i(e)$	$= f_i(e)/c_i(e)$	$\varrho(e)$	max number of channels available for link e
$N(v)$	set of data link neighbors of node v	$N^{\mathcal{I}}(v)$	set of interference link neighbors of node v
$E(v)$	set of data links incident on node v	$E^{\mathcal{I}}(v)$	set of interference links incident on node v

TABLE I

INDEX OF SYMBOLS USED IN THE PAPER

node v, with the link belonging to the set E. If node u can only interfere with node v (and vice versa), but cannot transmit data to it, then we represent it by a link in $E^{\mathcal{I}}, u \leftrightarrow v$. Note that a link $u \leftrightarrow v \in E$ implies that u is within the communication range of v, while $u \leftrightarrow v \in E^{\mathcal{I}}$ implies that u is between the interference and carrier-sensing range of v. These ranges need not be fixed numbers, but can vary based on the network topology, potential obstacles and the terrain of deployment.

There are $\mathcal{C}$ orthogonal channels in the network, denoted by the set $OC = 1, 2, ..., \mathcal{C}$. In the IEEE 802.11a standard, $\mathcal{C} = 12$. Each node v has $\kappa(v)$ radios. One of the practical constraints on radios is that it is not useful to have two radios tuned to the same channel at a given node, since local interference at the node will ensure that only one of them is active at any time. Therefore, it is possible that $\kappa(v) \leq \mathcal{C}$, though this is not a restricting factor in our model.

Given a data link $e \in E$, we use $t(e)$ to represent the transmission end of the link and $h(e)$ to be the receiving end of the link e. We say that a data link e is *active* when there is a transmission from $t(e)$ to $h(e)$. Each data link e has capacity $c_i(e)$ on channel i. We assume that for a given topology, the capacity is fixed for any given channel across a link. In other words, $c_i(e)$ does not change over time [1]. A flow on data link e using channel i is denoted by $f_i(e)$. We define $g_i(e) = f_i(e)/c_i(e)$ as *the utilization of channel i over link e*.

Given a node $v \in V$, $N(v)$ denotes all neighbors of v with data links to and from v. These data links are denoted by the set $E(v)$. We also assume that all link capacities, flows and rates are rational numbers. Table I lists the notations used in the paper.

We assume that system operates in a synchronous time-slotted mode, where the length of a time-slot is τ seconds. It is easy to see that for an asynchronous system, the results in this paper will serve as an upper bound on the performance.

2) Interference Model: We use the protocol interference model [5]. In this model, a transmission on channel i over link e is successful when all potential interferers in the neighborhood of the sender $t(e)$ and the receiver $h(e)$ are silent on channel i for the duration of the transmission. This is similar to the model used in IEEE 802.11, based on a RTS-CTS-Data-ACK sequence [3]. The *interference neighborhood* of a node v is defined to be the set of nodes that can interfere with node v. This is the set of nodes that have either a data link or an interference link incident on v. The protocol model of interference captures the behavior of the CSMA/CA protocol, which assumes bi-directionality of links for correct operation.

[1] If a feasible rate vector is recomputed every T_f time slots, then this assumption can be relaxed to saying that $c_i(e)$ is constant over the time period T_f.

3) Link Channel Assignment: At the beginning of each time slot every node has to make two decisions:

- Which node (if any) it is going to communicate with.
- The channels on which this communication is going to take place.

Both these decisions are negotiated between neighboring nodes and then transmission takes place. Notice that a node can simultaneously talk to its neighbor on multiple orthogonal channels provided both nodes have sufficient number of radios, with each radio at either endpoint tuned to a particular channel. The decision of which channels to communicate on, can either be done on a per time slot basis or can be fixed for a longer period of time. We assume that the channels over which a node communicates with its neighbor is decided on a per-time-slot basis, since this provides maximum flexibility which in turn maximizes capacity.

4) Design Considerations: **Link-Channel Restriction**: A node v can be active simultaneously on $\kappa(v)$ channels at the same time, and any subset m of these channels can be used to talk to one neighbor, assuming that the neighbor has at least m radios. We can restrict each data link e to use no more than a certain number of channels $\varrho(e)$. In this paper, for the evaluation, we do not place such restrictions, i.e., any number of channels can be used between two nodes subject to radio and channel constraints. In other words, $\varrho(u \leftrightarrow v) = \min(\kappa(u), \kappa(v), \mathcal{C})$.

Interference Links: We show in the next section how to model the constraints based on data links and interference links. Beyond that, we will then assume that there are no interference links in the graph to maintain clarity of presentation. Note that this is only to keep the presentation focused and allow the reader to grasp key concepts. Our algorithms do not need to be modified when interference links are present.

Multi-path Routing: Choosing only one route between a source and destination does not exploit the inherent multi-path diversity present in mesh networks for maximizing throughput. Therefore, we use multi-path routing for routing end-to-end flows. While this leads to questions regarding packet re-ordering and loss recovery, such issues are beyond the scope of this paper.

5) Approach: When measuring the asymptotic capacity of a random network, the per-node throughput is defined as the highest value of end-to-end flow that can be supported by every source-destination pair in the network, where each node picks its destination at random. We will solve a similar problem in this paper, where for a given instance of the network, each node randomly choses its destination, and attempts to send one unit of flow across the network. Our goal is then to determine, by using an optimization formulation, how much this unit of flow can be scaled in the given MC-MR network instance, in order to obtain the network capacity.

We proceed in three steps: (a) we first determine the constraints placed by the nodes, channels, interference model, and the network parameters on the feasibility of a flow, (b) we use link flow feasibility constraints as necessary conditions and solve the optimization problem using the stated objective, and (c) we perform joint link channel assignment and scheduling to obtain a feasible schedule from the solution in step (b). The last step gives a feasible lower bound on the optimum per-node throughput, and hence, the capacity, while the second step provides an upper bound on the optimum.

III. Constraint Model and Optimization Framework

We now present a mathematical constraint model and the optimization framework for the *MC-MR* fixed wireless mesh network described in Section II-B. This is an overview of the work in [1].

A. Channel, Node and Interference Constraints

Assume that we are given the *link flow set*, $\mathbf{f} = \{f_i(e)\}$, where $f_i(e)$ is the desired flow on channel $i \in OC = \{1, 2, ..., \mathcal{C}\}$ over link $e \in E$. The objective now is to determine necessary and sufficient conditions for this link flow vector to be achievable in the network in terms of a valid schedule.

In order to achieve this link flow we first define a $0 - 1$ scheduling variable

$$y_i^t(e) = \begin{cases} 1 & \text{If link } e \text{ is active on channel } i \text{ in time slot } t \\ 0 & \text{Otherwise} \end{cases}$$

Note that $y_i^t(e)$ is set to one if there is a transmission on channel i over link e in time slot t. Note that $y_i^t(e) = 0, \forall i \in OC, \forall t$ when $e \in E^{\mathcal{I}}$. In other words, interference links do not carry data.

Link-Channel Constraint: By definition, the maximum number of channels that can active on link e at any time slot t is $\varrho(e)$. Thus, we have

$$\sum_{i \in OC} y_i^t(e) \leq \varrho(e), \ \forall e \in E, \forall t \tag{1}$$

Node-Radio Constraint: A node can use at most $\kappa(v)$ radios in a given time slot for transmission or reception or both. This leads to the following constraint.

$$\sum_{e \in E(v)} \sum_{i \in OC} y_i^t(e) \leq \kappa(v), \qquad \forall v \in V \quad \forall t \tag{2}$$

Interference Constraint: Let us initially assume that the antennas are omni-directional and that $E^{\mathcal{I}} = \phi$, i.e., there are no interference links in the set. The extensions needed to cover interference links are described in [1].

We consider the IEEE 802.11 based RTS-CTS-DATA-ACK model to identify pairs of nodes that can simultaneously transmit. In this model, neighbors of both an intended transmitter and receiver have to refrain from both transmission and reception. Interference will occur only among users sharing the same channel, say channel i. Consider a node v and its neighborhood $N(v)$ in G. Let one of the links e in $E(v)$ be active on channel i and let u be the other endpoint of the link. For e to be active on channel i, all other links incident on v, $E(v) \setminus e$, have to be idle and, in addition, each neighbor of v must remain idle, on channel i. The same argument applies to u. For silencing $N(v)$, due to the non-overlapping neighborhoods of nodes in $N(v)$, we have to include them in separate constraints, as follows.

$$\sum_{e \in E(v) \cup E(v')} y_i^t(e) \leq 1, \quad \forall \ v' \in N(v), \forall i \in OC, \ \forall t$$

This can be seen in Figure 3(a) for link uv and nodes $v_1, v_2 \in N(v)$.

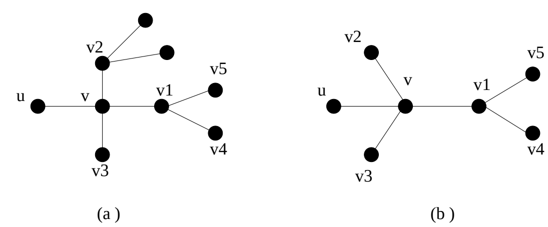

Fig. 3. Interference Constraints

Interestingly, these constraints are the same whenever any link incident on v has to be active, since it prevents other links incident on $\{v\} \cup N(v)$ from being active at the same time on the same channel. Therefore, we can rewrite these constraints and state the interference constraint as follows, in terms of a link e.

$$\sum_{e' \in E(t(e)) \cup E(h(e))} y_i^t(e') \leq 1, \qquad \forall i \in OC, \forall e \in E, \forall t \tag{3}$$

Note that there are only $|E|$ interference constraints in all, under this approach, with the number of variables per constraint never exceeding the product of the number of orthogonal channels and twice the maximum degree of the graph.

If we include interference links, we have the following interference constraints for the general graph [1].

$$\sum_{e' \in E(t(e)) \cup E(h(e))} y_i^t(e') \leq 1, \qquad \forall i \in OC, \forall e \in E \cup E^{\mathcal{I}}, \forall t \tag{4}$$

Table II lists these three constraints that characterize the *MC-MR* wireless network G. Each of these constraints characterize the channel, node and interference constraints respectively, and thus, equations (1), (2), and (4) are both necessary and sufficient conditions to check for the feasibility of a link schedule in the *MC-MR* network G.

ILP: 0–1 Variables (Necessary & Sufficient Conditions)	LP: Continuous Variables (Necessary Conditions)
$\sum_{i\in OC} y_i^t(e) \le \varrho(e),\ \forall e \in E, \forall t$	$\sum_{i\in OC} g_i(e) \le \varrho(e),\ \forall e \in E$
$\sum_{e\in E(v)} \sum_{i\in OC} y_i^t(e) \le \kappa(v),\ \forall v \in V\ \forall t$	$\sum_{e\in E(v)} \sum_{i\in OC} g_i(e) \le \kappa(v),\ \forall v \in V$
$\sum_{e'\in E(t(e))\cup E(h(e))} y_i^t(e') \le 1,\ \forall i \in OC, \forall e \in E \cup E^{\mathcal{I}}, \forall t$	$\sum_{e'\in E(t(e))\cup E(h(e))} g_i(e') \le 1,\ \forall i \in OC, \forall e \in E \cup E^{\mathcal{I}}$

TABLE II
NETWORK CHARACTERIZATION: CONSTRAINTS

While the necessary and sufficient conditions are required to check for the feasibility of link channel assignments (see Section IV), the variables in these equations are binary variables, and as such, they are inconvenient to use in any optimization problem, as they lead to Integer Linear Programming problems (ILPs) which are much harder to solve than linear programs defined on continuous variables. Moreover, the variables are time-indexed which makes the problem size very large. We therefore seek a relaxation of the integral constraints to continuous variables in terms of link flows.

Over a period of time $[0,T]$, the fraction of time link e is active on channel i is given by $(\sum_{t\le T} y_i^t(e))/T$. Therefore, the mean flow on channel i over link e is given by

$$f_i(e) = \frac{c_i(e) \sum_{t\le T} y_i^t(e)}{T}. \tag{5}$$

Let $g_i(e) = f_i(e)/c_i(e)$ be the mean utilization of channel i over link e over the period $[0,T]$, as defined earlier in Section II-B.

We can average the variables in Equations (1), (2) and (4) to obtain the constraints in terms of link utilizations. Thus, we state the following lemma.

Lemma 1: For the multi-channel multi-radio multi-hop wireless network under consideration, if a given link flow set **f** does not satisfy the following inequalities

$$\sum_{i\in OC} g_i(e) \quad \le \quad \varrho(e),\ \forall e \in E \tag{6}$$

$$\sum_{e\in E(v)} \sum_{i\in OC} g_i(e) \quad \le \quad \kappa(v),\ \forall v \in V \tag{7}$$

$$\sum_{e'\in E(t(e))\cup E(h(e))} g_i(e') \quad \le \quad 1,\ \forall i \in OC, \forall e \in E \cup E^{\mathcal{I}} \tag{8}$$

then the link flow set **f** is not schedulable.

1) Optimality Gap: Due to the relaxation of the integer variables by averaging, the above inequalities are only necessary conditions and are no longer sufficient conditions. A simple example is a *4-cycle* $A \leftrightarrow B \leftrightarrow C \leftrightarrow D \leftrightarrow A$. Let the capacity on each link be 1. A link flow of $1/3$ on each link satisfies the necessary conditions, for a total link utilization of $4/3$, but it is not sufficient as only one link can be active at any time, resulting in a total link utilization of 1. In fact, the gap between the necessary conditions and the optimal can be unbounded[1].

However, the key point is to note that *the ILP constraints in Table II are both necessary and sufficient for any network graph*. The gap arises only due to relaxation of the integral constraints. Any solution satisfying the necessary conditions will be an upper bound of the optimal solution. Therefore, by using these necessary conditions as optimization constraints, we can derive upper bounds for the performance and then compute a feasible lower bound using channel assignment and scheduling, based on the solution from the optimization problem. This approach will mitigate the impact of the optimality gap.

2) Generalizing the Constraint Sets: Note that all of the constraints described earlier have a common structure. We have $\mathcal{L}$ sets composed of *(link, channel)* pairs, $S_1, S_2, \ldots, S_{\mathcal{L}}$ that are defined on links e and colors $i \in OC$. The necessary and sufficient conditions (using binary variables) take the general form $\sum_{(e,i)\in S_j} y_i^t(e) \le \beta(S_j),\ \forall j$, while the necessary conditions (using continuous variables) have the general form $\sum_{(e,i)\in S_j} g_i(e) \le \beta(S_j),\ \forall j$, where $\beta(S_j)$ is the RHS constant associated with the constraint identified by that set. For example, $\beta(S_j) = 1$ for the sets identified by the interference constraints, and $\beta(S_j) = \kappa(v)$ for the sets identified by the node-radio constraints.

We now restate Lemma 1 in a generic form as follows.

Lemma 2: Let $S_1, S_2, \ldots, S_{\mathcal{L}}$ represent sets of (link, channel) pairs identified by the constraints in Lemma 1, with $\beta(S_j)$ being the associated constant for each set S_j. Let **f** represent a link flow set $\{f_i(e)\}$, where $f_i(e)$ represents the flow on channel i over link e. If **f** does not satisfy the following inequalities

$$\frac{1}{\beta(S_j)} \sum_{(e,i)\in S_j} g_i(e) \le 1,\ \forall j \in \{1, 2, ..., \mathcal{L}\} \tag{9}$$

then the link flow set **f** is not schedulable.

With this set of constraints that characterize a *MC-MR* wireless mesh network in hand, we now proceed to address the problem of optimization of throughput criteria subject to these constraints.

B. Optimization: Multi-Commodity Flows and Upper Bounds

We define a standard multi-commodity flow problem on the *MC-MR* network: a set of sources, **s**, want to send data to a set of destinations **d**, with a end-to-end rate demand vector **r**. There can be multiple objectives of interest to the network planner: (a) achievability of the demand vector, (b) maximizing aggregate network throughput or the sum of end-to-end rates subject to minimum rate requirements, (c) maximizing the minimum end-to-end rate, (d) maximizing the aggregate utility function of the end-to-end rates, or (e) imposing certain fairness criteria on the rates in the network. All of these problems can be solved using our framework described here. We are, however, more interested in the feasibility problem, which allows us to identify the capacity of a random network. First, we translate the constraints identified earlier from link-flow variables to end-to-end rate variables.

INPUT: A directed graph $G = (V, E, E^{\mathcal{I}})$ with a link speed $c_i(e)$ for channel $i \in OC$, data link $e \in E$ and Q node pairs $(s(q), d(q))$ and a desired rate $r(q)$ associated with each node pair q.
OUTPUT: A set of routes, link channel assignments and associated schedule that achieves the given rates or declare the problem infeasible.

Fig. 4. Optimization Problem

1) End-to-End Flow Constraints for Routing Multiple Source Destination Pairs: We assume that the traffic demand for different source-destination pairs is given in the form of a rate vector $\mathbf{r}$. We assume that the rate vector[2] has $Q < n(n-1)$ components. Each source-destination pair between which there is a request will be termed a commodity. We use q to index the commodities. Let $s(q)$ represent the source node for commodity q and $d(q)$ the destination node for commodity q. Let $r(q)$ represent the flow that has to be routed from $s(q)$ to $d(q)$. The problem that we have to solve is shown in Figure 4.

Recall that multiple paths can exist between $s(q)$ and $d(q)$ for each commodity q. It is easy to show the following result.

Theorem 3: Given a graph $G = (V, E, E^{\mathcal{I}})$, with link speed $c_i(e)$ associated with data link $e \in E$ and channel $i \in OC$, Q source destination pairs $(s(q), d(q))$ for $q = 1, 2, \ldots Q$ with a desired flow rate $r(q)$ between $s(q)$ and $d(q)$, let $x_i^q(e)$ be the flow on channel i over data link e that belongs to the end-to-end flow q. A necessary condition for rate vector r to be achievable is the existence of link flows $x_i^q(e), \forall i, q, e$ that satisfies the following constraints.

$$\begin{aligned}
\sum_{e:t(e)=s(q)} \sum_{i \in OC} x_i^q(e) &= r(q), \forall q \\
\sum_{e \in E_{in}(v)} \sum_{i \in OC} x_i^q(e) &= \sum_{e \in E_{out}(v)} \sum_{i \in OC} x_i^q(e), \\
&\qquad \forall v \neq s(q), d(q)\ \forall q \\
\frac{1}{\beta(S_j)} \sum_{(e,i) \in S_j} \frac{\sum_{q \le Q} x_i^q(e)}{c_i(e)} &\le 1 \ \ \forall q, j.
\end{aligned}$$

The first constraint ensures the end-to-end rate is met. The second constraint maintains flow balance at intermediate nodes in the network for each end-to-end flow. The third constraint is a restatement of Equation (9) in terms of $x_i^q(e)$, since $f_i(e) = \sum_{q \le Q} x_i^q(e)$.

An alternate formulation of the above conditions can be given in an arc-path formulation. Let $\mathcal{P}_q$ represent the set of (link, channel) pairs for the source-destination pair q. Consider a path $P \in \mathcal{P}_q$. Let $x(P)$ be the amount of flow sent on that path. This path leads from $s(q)$ to $d(q)$. From the demand requirements, note that

$$\sum_{P \in \mathcal{P}_q} x(P) = r(q) \quad \forall q.$$

The total amount of flow on channel i over link e, represented by $f_i(e)$ is given by

$$f_i(e) = \sum_q \sum_{P \in \mathcal{P}_q : (e,i) \in P} x(P).$$

Thus, the necessary conditions for a rate vector r to be achievable is given by

$$\begin{aligned}
\sum_{P \in \mathcal{P}_q} x(P) &= r(q), \forall q \\
\sum_{(e,i) \in S_j} \frac{\sum_q \sum_{P \in \mathcal{P}_q : P \ni (e,i)} x(P)}{\beta(S_j) c_i(e)} &\le 1, \ \forall j \in \{1, 2, ..., \mathcal{L}\}
\end{aligned}$$

Given a rate vector $\mathbf{r}$, the strategy then is to solve for the x variables that satisfies the necessary conditions.

2) The Feasibility of Demands: We now consider the capacity determination problem. This can be formulated as a fundamental optimization problem: *feasibility*. We want to know if a given rate-demand vector can be achieved in the network. We will use the optimization framework to derive an upper bound, and in the next section, we describe the procedure to obtain a lower bound for this problem.

Instead of solving the feasibility problem directly, we write it in the form of a concurrent flow problem. In the concurrent flow problem, the desired rate vector is scaled and the objective is to determine the maximum scaling factor that still satisfies the necessary conditions. Note that there can be an exponential number of paths between two given nodes in the network, resulting in an exponential number of variables in the path-arc formulation. Our formulation allows us to avoid this problem, however, using a primal-dual approach based on shortest path routing.

3) Solving the Concurrent Flow Problem: We first write the feasibility problem as a concurrent flow problem and then use a primal-dual algorithm to solve the linear programming problem.

$$\max \lambda$$

$$\begin{aligned}
\sum_{(e,i) \in S_j} \frac{\sum_q \sum_{P \in \mathcal{P}_q : P \ni (e,i)} x(P)}{\beta(S_j) c_i(e)} &\le 1, \ \forall j \in \{1, ..., \mathcal{L}\} \\
\sum_{P \in \mathcal{P}_q} x(P) &= \lambda . r(q), \ \forall q \\
x(P) &\ge 0, \ \forall P \in \mathcal{P}_q, \ \forall q
\end{aligned}$$

Let λ^* be the optimal solution to the linear programming problem above. λ^* represents the maximum scaling factor by which flows can be scaled up, and still satisfy the necessary constraints. Therefore, if $\lambda^* < 1$, then the rate vector is not feasible, and its value (and that of the constrained variables) will denote how much slack capacity needs to be added to the network to make the demand vector feasible. The largest rate vector that still satisfies the necessary constraints is $\lambda^* \mathbf{r}$, given by the optimal path flow vector $\mathbf{x}^*$. Given a slot-length of τ seconds, we seek to know if we can schedule $\mathbf{r}$ bits in a schedule of length at most $1/\tau$. Thus, if L^* is the smallest length of a schedule that can schedule $x^* = \lambda^* \mathbf{r}$ bits, then the schedule corresponds to a flow vector $\frac{\lambda^* \mathbf{r}}{L^* \tau}$. For this achievable flow to be at least $\mathbf{r}$, we need $\lambda^* \ge L^* \tau$. The sufficiency gap created by

[2]Depending on the optimization objective, $\mathbf{r}$ can also be set to $\mathbf{0}$.

relaxing the integer constraints (given by Equations (1), (2) and (4)) is denoted by the interval $[1, L^*\tau)$. It indicates the space where we cannot guarantee the existence of a schedule.

We solve this problem using a primal-dual approach, and this procedure and the algorithm is described in [1]. This algorithm gives a Fully Polynomial Time Approximation Solution, where we can control the accuracy of the desired approximation as needed.

IV. Link Channel Assignment and Scheduling

We had earlier assumed that every link has the ability to switch channels once every T_d time-slots ($T_d \geq 1$). The channels still have to respect the constraints imposed by Equations (1), (2) and (4). This implies that apart from the coordination of the nodes at the end of the links, there has to be co-ordination across different links to perform link channel assignment at the beginning of each time period. Though this may be difficult in practice, when $T_d = 1$, such a dynamic link channel assignment gives the highest flexibility to maximize the achievable performance for any link channel assignment scheme. Since our goal is to measure the capacity, we will assume $T_d = 1$ for the rest of this paper. However, the algorithm proposed here can be adapted to a different value of T_d without much difficulty.

The linear program described in the previous section gives an upper bound on the achievable rates in the mesh network. We use this LP solution to assign channels to the links and also schedule the time slots in which each link and channel is active. Both these problems are NP-hard [7]. As a first cut approach, we use a variation of the greedy approach to solve the problem. Unlike the algorithms proposed in [1], the greedy algorithm proposed here is much simpler, due to the fact that we do not have any restrictions on how many channels can be simultaneously assigned to a particular link. Our solution approximates the optimal solution by using a greedy packing-based heuristic.

Let $f_i(e) = \sum_{q \leq Q} \hat{x}_i^q(e)$ represent the desired flow on link e corresponding to channel i. Since we do not necessarily have to assign the link flows to channels that are given by the LP solution, we first aggregate all the flows on the different channels on a given link into a single scaled flow on the link denoted by $f^M(e) = M \sum_i f_i(e)$. The value of M was set to 100 in our experiments, in order to make the fractional part negligible. As we pack the flows in each time slot, we denote the amount of unassigned flow on link e by $d(e)$.

At the beginning of each time slot, we sort the links in descending order of the unassigned flows. We assign the first link e to the channel $j = \arg\max_i c_i(e)$ and decrement the value of $d(e)$ by $c_j(e)T_d$. We then check if the next link in the ordered list can be assigned to some channel in this time-period. If this can be done, then we pick such a channel with the highest capacity and assign the link to that channel. If this link cannot be assigned to any channel, then we move on to the next link. and repeat this until we reach the end of the list. *We repeat this procedure again until no channels have been assigned to any link in an iteration*. If there are still unassigned flows, we move to the time-slot and repeat the assignment process. The objective of course, is to assign all flows in as few time slots as possible.

The way we check if a particular channel is feasible for a data link in a time-period is by having a binary variable z_S associated with each constraint set S. At the beginning of each time slot, z_S is set to zero. We set z_S to one whenever a data link e is assigned channel i and $(e, i) \in S$. We also ensure that the *node-radio constraint* is respected at each node by verifying that the number of channels allocated to a given node in any time slot is not greater than the number of radios at that node.

In the description of the greedy algorithm shown in Figure 5, we use $T(e, i)$ to denote the collection of constraint sets that are associated with the (link, channel) pair (e, i) and NS to count the number of time slots. The demand met by the system is computed as

$$\frac{M}{NS}\lambda^* \mathbf{r},$$

where $\lambda^* \mathbf{r}$ is the LP optimal solution, and M is the multiplicative factor used to convert the link flows to integral values.

```
NS = 0; A = E; d(e) = f^M(e)  ∀e ∈ E
While A ≠ ∅
    NS ← NS + 1; iter = 1
    Sort links in decreasing order of d(e).
    Assume links are numbered in decreasing
    order of d(e)
    z_S = 0   ∀ constraint sets S.
    While iter = 1
        iter = 0
        For e = 1, 2, ... E
            If ∃i such that z_S = 0  ∀S ∈ T(e, i)
                j = arg max_i c_i(e)
                Assign e to channel i
                iter = 1
                d(e) ← d(e) − T_d c_j(e)
                If  d(e) = 0 then  A = A\e
                Set z_S = 1, ∀S ∈ T(e, j)
            End If
        end For
    end While
end While
```

Fig. 5. Greedy Channel Scheduling Algorithm

V. Performance Evaluation

In this section, we aim to quantitatively evaluate the performance of our greedy channel scheduling algorithm against the upper bounds on capacity derived using the optimization model described in Section III. The evaluation is done by means of simulations. We highlight three key aspects in this section:

1) The ability of the necessary conditions based on continuous flow variables to model the capacity of the network in practical circumstances
2) The performance of the greedy channel scheduling algorithm with respect to the capacity of the network
3) The variation in achievable channel capacity as the number of radios and channels vary.

Our simulation platform is a custom-written simulator that uses a primal-dual FPTAS LP solver, along with a link scheduler that assigns channels and slots to links to meet a certain demand. This platform was previously used to report the results in [1], [16], [15], [17]. The simulator's performance is limited by the amount of memory of the machine on which it is running. Our experiments were conducted on a dual-processor Pentium-III Linux server with 2GB of RAM. This allows us to simulate mesh network topologies where the product of the number of links and the number of orthogonal channels does not exceed 10000. We hope to optimize the memory management of the simulator further to run larger instances in the future.

A. *Test Methodology*

Our goal here is to measure the achievable capacity for a large network. Our optimization framework allows for a multi-commodity flow problem formulation. Therefore, we let each node pick a randomly chosen node as its destination, and set a demand of 1 unit of flow, end-to-end, for all source-destination pairs. When we solve the feasibility/concurrent flow problem on this instance, then we obtain a scaling factor λ^*, which implies that all end-to-end flows can send at most λ^* units of flow. Thus, the total network capacity is upper bounded as $n.\lambda^*$ [5], [2]. We then run the greedy channel scheduling algorithm to obtain an achievable schedule and compute a lower bound on the capacity from the schedule. A good schedule will have a very small gap between the upper and lower bounds on capacity.

In order to replicate the same capacity assumptions outlined in [2], we set the number of radios per node to be the same across all nodes, i.e.,$\kappa(v) = \kappa, \forall v \in V$, and set the links to be of unit capacity for all channels, $c_i(e) = 1, \forall i \in OC, e \in E$. Note that all channels are orthogonal. We do not model wireless channel errors at this time as our goal is estimate the maximum capacity of the network.

Our experiments were conducted for large topologies, and the results presented in this section are an average of five random topologies, each with 100 nodes. The number of links in each of these topologies varies from 350 to 900. There are no partitions in the network graph, and all links are bi-directional. Note that multi-path routing is enabled in all these instances. We vary the number of channels from 1 to 8.

We measure two parameters for each topology: (a) the capacity upper bound and (b) the greedy channel scheduling lower bound. All these values are in terms of the scaling factor λ.

B. *Random Networks*

We summarize the results of our evaluation in Figures 6, 7 and 8. The first set of results show the performance of the greedy channel scheduling algorithm, as illustrated in Figure 6. The key point to note is that for all combinations of channels and number of radio interfaces per node, the greedy channel scheduling algorithm performs within 80% of the upper bound, and is typically within 90% of the upper bound in most instances. While the worst-cases may not have been encountered in the topologies considered here, we have tested this algorithm on many other smaller topologies, and have found that the results shown here are typical of those cases.

In addition to proving the efficacy of the greedy channel scheduling algorithm, the figure can also be used to infer that *the upper bounds is a valid approximation of the network capacity*. Hence, we will now use the upper bound numbers to explain the capacity of the network.

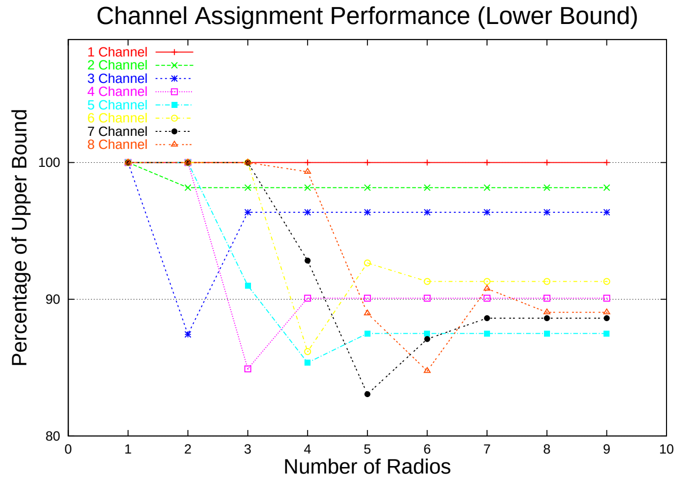

Fig. 6. Greedy Algorithm Performance

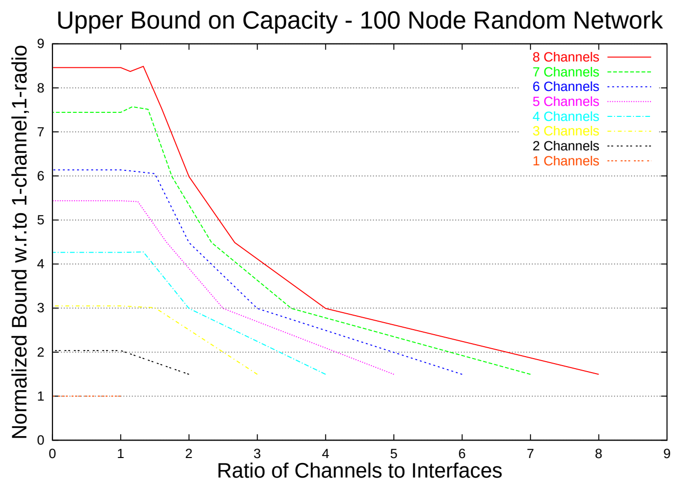

Fig. 7. Variation of capacity with $\mathcal{C}/\kappa$
The y-axis is normalized w.r.t. the upper bound for the case of 1 Radio per node and 1 Channel.

First, we revisit Figure 2 in Section I. The key observation in [2] was that network capacity of an *MC-MR* network depends primarily on the ratio of the number of channels to the number of interfaces per node, and not on the individual values themselves. Our results, shown in Figures 7 and 8 confirm this observation. Consider Figure 7, where we show the network capacity $n.\lambda^*$, normalized with respect to the capacity of the network with 1 channel and 1 radio interface per node. Each data point is averaged over 5 random graph topologies with 100 nodes each, as described earlier. It confirms that network capacity starts declining as soon as the ratio $\mathcal{C}/\kappa$ exceeds a certain threshold, as predicted in [2].

In Figure 8, we show the network capacity for each channel, again normalized by the network capacity for the (1 channel, 1 radio) case. These results re-affirm the validity of the results in [2]. It also shows that the $\Theta(W\sqrt{\frac{n}{\log n}})$ function is tightly bounded[3], since the per-channel network capacity does not vary significantly at all.

[3]When $f() = \Theta(g())$ is tightly bounded, it implies that $c_1 g() \leq f() \leq c_2 g()$, with $\frac{c_2}{c_1} \approx 1$.

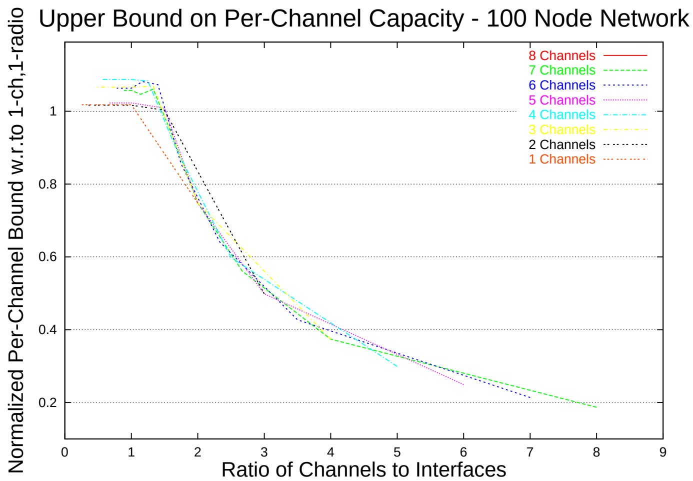

Fig. 8. Per-Channel Capacity variation with $\mathcal{C}/\kappa$

However, the most significant inference from these results is the fact that the threshold for $\mathcal{C}/\kappa$ beyond which network capacity begins to decrease is very small, in fact, close to 1.5. Figure 2 proposes that the network capacity decrease threshold is $O(\log n)$. Note that for a 100- node network, $\log_1 0100 = 2$, and $\ln 100 = 4.61$. This suggests that the constants associated with the $O()$ notation are small enough to deny any potential benefits that one might expect from this capacity trend for a reasonably large *MC-MR* network. Clearly, this threshold will increase for a larger network, but the increase will be meaningful only when the network size increases by a couple of orders of magnitude. Thus, the claim made in [2] that a single radio per node might be potentially good enough to maximize network capacity may be valid only for networks with either millions of nodes, or networks with only one channel of operation. A single radio could also perform similar to a multiple radio-per-node network when the traffic load is light, though this does not mean capacity will also be the same.

VI. Conclusion

In this paper, we described a network optimization framework that captures the constraints associated with multi-channel multi-radio *(MC-MR)* multi-hop wireless mesh networks. We presented a simple greedy channel assignment algorithm that uses this network characterization to obtain near-optimal performance based on the upper bounds on capacity derived via the optimization framework. In addition, by means of extensive simulations on large-scale networks, we realized a practical channel assignment and schedule to compute achievable network capacity, and verified a recent theoretical capacity model proposed in [2]. Our evaluation also highlighted the fact that in order to obtain maximum network capacity for a given instance of a medium-to-large sized *MC-MR* network with $\mathcal{C}$ orthogonal channels, nodes should have around the same number ($\mathcal{C}$) of radio interfaces per node.

In conclusion, we hope that the results identified in this paper can be a valuable tool for network designers in planning network deployment and in controlling performance objectives in the emerging field of wireless mesh networks.

References

[1] M. Kodialam and T. Nandagopal, “Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks”, *ACM Mobicom*, August 2005.

[2] P. Kyasanur and N. Vaidya, “Capacity of Multi-Channel Wireless Networks: Impact of Number of Channels and Interfaces”, *ACM Mobicom*, August 2005.

[3] IEEE Std. 802.11-1999, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifi cations”, 1999.

[4] IEEE Std. 802.16-2004, “Part 16: Air Interface for Fixed Broadband Wireless Access Systems”, July 2004.

[5] Gupta, P., and Kumar, P.R., “The Capacity of Wireless Networks”, *IEEE Transactions on Information Theory*, 46(2), pp. 388-404, 2000.

[6] A. Raniwala and T. Chiueh, “Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network”, *IEEE Infocom*, March 2005.

[7] A. Raniwala, K. Gopalan and T. Chiueh, “Centralized Algorithms for Multi-channel Wireless Mesh Networks”, *ACM Mobile Computing and Communications Review*, April 2004.

[8] P. Kyasanur and N. Vaidya, “Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks”, *IEEE WCNC*, 2005.

[9] P. Kyasanur and N. Vaidya, “Routing in Multi-Channel Multi-Interface Ad Hoc Wireless Networks”, *Technical Report*, December 2004.

[10] De Couto, D., Aguayo, D., Bicket, J., and Morris, R., “High throughput path metric for multi-hop wireless routing”, *ACM Mobicom*, 2003.

[11] Draves, R., Padhye, J., and Zill, B., “Routing in Multi-radio, Multi-hop Wireless Mesh Networks”, *ACM Mobicom*, Sept. 2004.

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

[13] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, “A Multi-Radio Unifi cation Protocol for IEEE 802.11 Wireless Networks”, *IEEE Broadnets*, 2004.

[14] Jain, K., Padhye, J., Padmanabhan, V., and Qiu, L., “Impact of Interference on Multi-hop Wireless Network Performance”, *ACM Mobicom*, September 2003.

[15] Kodialam, M., and Nandagopal, T., “The Effect of Interference on the Capacity of Multi-hop Wireless Networks”, *IEEE Symposium on Information Theory*, June 2004.

[16] Kodialam, M., and Nandagopal, T., “Characterizing Achievable Rates in Multi-hop Wireless Networks : The Joint Routing and Scheduling Problem”, *ACM Mobicom*, September 2003.

[17] Kodialam, M., and Nandagopal, T., “Characterizing Achievable Rates in Multi-hop Wireless Mesh Networks with Orthogonal Channels”, *IEEE/ACM Transactions of Networking*, September 2005.

[18] Krumke, S., Marathe, M., and Ravi, S.S., “Models and Approximation Algorithms for Channel Assignment in Radio Networks”, *ACM Wireless Networks*, vol. 7(6), pp. 575-584, 2001.

[19] Kumar, A., Marathe, M., Parthasarathy, S., and Srinivasan, A., “End-to-end packet-scheduling in wireless ad-hoc networks”, In *ACM SODA*, pp. 1021-1030, 2004.

[20] Gastpar, M., and Vetterli, M., “On the Capacity of Wireless Networks: The Relay Case”, *IEEE Infocom*, pp. 1577-1586, 2002.

[21] Grossglauser, M., and Tse, D.N.C., “Mobility Increases the Capacity of Ad Hoc Wireless Networks”, *IEEE/ACM Transactions on Networking*, 10(4), pp. 477-486, 2002.

[22] Li, J., Blake, C., De Couto, S.J., Lee, H., and Morris, R., “Capacity of Ad Hoc Wireless Networks”, *ACM Mobicom*, pp. 61-69, 2001.

[23] Kail, E., Németh, G., and Turányi, Z., “Throughput of Ideally Routed Wireless Ad Hoc Networks”, *ACM MobiHoc*, pp. 271-274, 2001.

[24] Alicherry, M., Bhatia, R., and Li, L., “Joint Channel Assignment and Routing for Throughput Optimization in Multi-radio Wireless Mesh Networks”, *ACM Mobicom*, 2005.