Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks

Murali Kodialam
Bell Labs, Lucent Technologies
Holmdel, NJ 07733. USA
muralik@lucent.com

Thyaga Nandagopal
Bell Labs, Lucent Technologies
Holmdel, NJ 07733. USA
thyaga@lucent.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. In this paper, we propose a network model that captures the key practical aspects of such systems and characterize the constraints binding their behavior. We provide necessary conditions to verify the feasibility of rate vectors in these networks, and use them to derive upper bounds on the capacity in terms of achievable throughput, using a fast primal-dual algorithm. We then develop two link channel assignment schemes, one static and the other dynamic, in order to derive lower bounds on the achievable throughput. We demonstrate through simulations that the dynamic link channel assignment scheme performs close to optimal on the average, while the static link channel assignment algorithm also performs very well. The methods proposed in this paper can be a valuable tool for network designers in planning network deployment and for optimizing different performance objectives.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communication; C.2.8 [**Mobile Computing**]: Algorithm Design and Analysis; G.1.6 [**Optimization**]: Linear Programming

General Terms

Algorithms, Design, Theory

Keywords

Algorithms, Capacity, Graph Coloring, Optimization, Routing, Scheduling, Wireless Networks

MobiCom'05, August 28–September 2, 2005, Cologne, Germany.

1. INTRODUCTION

The emergence of broadband wireless networks has been spurred by the development of standards such as IEEE 802.11 a/b/g [1] and 802.16 [2]. These networks are being deployed as a solution to extending the reach of the last-mile access to the internet, using a multi-hop configuration. Many such networks are already in use, ranging from prototype testbeds [3] to complete commercial solutions [4]. 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.

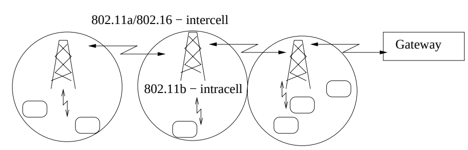

Figure 1: Network Example

1.1 Wireless Mesh Networks

The main characteristics of the fixed broadband wireless networks considered in this paper can be summarized as follows.

1. There are multiple wireless channels of operation and these channels are orthogonal to each other.
2. Nodes are not mobile and have multiple radio transceivers, which allow them to communicate, interference-free, simultaneously with more than one neighbor at the same time using different channels.
3. 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 $A \neq B$.
4. The number of orthogonal wireless channels could be limited in number, which implies that more than one node in a given region could contend for the same channel at the same time, thereby resulting in interference and collisions.
5. The radios (NICs) at each node are capable of fast switching between channels, with a switching overhead.
6. Channels can be assigned for communication between neighbors in a static or dynamic fashion. In a static link channel assignment, every link between a pair of neighboring nodes is bound to a particular channel and this binding does

not vary over time. In dynamic link channel assignment, the bindings between a link and the operating channel for that link can vary dynamically with time. While dynamic assignment involves negotiation, it also provides more flexibility to combat interference.

We identify such fixed multi-channel multi-hop wireless networks with multiple radios per node as *MC-MR* networks.

One of the main goals in the design of fixed wireless broadband networks is capacity planning. Within this realm, given a set of end-to-end demands, there are multiple design goals for which a network can be optimized, for example, maximizing a function of the rates where the function can be chosen to be an user utility function or a network price function, ensuring some notion of rate fairness, or minimizing end-to-end delays. For every design goal, one needs to be able to *design routes, assign channels and schedule packets* to meet these goals.

There are numerous studies [7]-[22] that consider routing, channel assignment and scheduling for such networks. Some study the problem of finding efficient routes to maximize throughput [9, 10, 13, 14, 16], while some consider only channel assignment and scheduling [15, 21, 22] and others consider both routing and scheduling [7, 8, 17, 18, 19, 20]. All of these previous studies consider only a subset of the problem presented above, with most of the papers only addressing the question of how to improve throughput compared to some other algorithm. In our view, the more relevant question is how to characterize the capacity region of the network for a given optimization objective. In this paper, we address this capacity planning question, and develop algorithms 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. To the best of our knowledge, we are not aware of any other work that considers the computation of capacity in a multi-radio multi-channel wireless mesh network.

1.2 Our Contributions

Our contributions are as follows.

1. We develop a network model that characterizes the multi-hop and multi-radio features in a fixed broadband wireless network with limited number of orthogonal channels and with multiple radios at each node. Our model provides both *necessary and sufficient conditions* for a feasible channel assignment and schedule in the network, using the protocol model of interference [5]. Our model also has the flexibility of specifying neighbors and interferers for each node in an arbitrary fashion to suit the actual system needs. We propose extensions that allow us to model multiple heterogeneous wireless standards with inherent rate diversity inter-operating with each other as part of a unified multi-radio multi-channel mesh network.

2. Due to the hardness of the joint routing and scheduling problem, we provide a relaxation of the model above to a linear program that gives the *necessary conditions* for a valid feasible solution. In addition to computing feasibility of a solution for a given objective function, our model can also point out potential bottlenecks that cause performance to degrade, which is a key requirement for network capacity planning.

3. Among the many possible capacity planning objectives, in this paper, we consider the fundamental problem of feasibility of a given end-to-end demand vector, and design a fast primal-dual algorithm for a fully polynomial time approximation solution (FPTAS) which provides an upper bound to the feasibility problem. Our algorithm is extensible to the optimization of similar rate-based objective functions that are useful for characterizing the capacity region.

4. We propose two link channel assignment algorithms: *Balanced Static Channel Assignment* (BSCA), and *Packing Dynamic Channel Assignment* (PDCA) that allocate channels to links in order to meet any given demand vector which satisfies the constraints imposed by our network model. These algorithms are applicable to any objective function used for capacity-planning. The BSCA algorithm performs static link channel assignment followed by greedy coloring for conflict resolution, while the PDCA algorithm performs link channel assignment and scheduling simultaneously. The PDCA algorithm gives an achievable lower bound on the capacity and can be used to compare the performance of any channel assignment algorithm, and evaluate how it performs with respect to the capacity upper bound obtained using the techniques proposed in this paper.

5. We show using extensive simulations on a range of graphs that the aggregate (joint routing, link channel assignment and scheduling) performance is very close to the optimum in practice for the PDCA algorithm, even though 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. We also quantify the performance of the BSCA algorithm and identify avenues for further performance improvements.

The solutions proposed in this paper for general graphs are non-trivial because the computation of the feasible scheduling space is NP-hard, even if one looks at a subset of restrictive classes of graphs such as disk graphs or $(r, s)-$ civilized graphs[21, 33]. In addition, the link channel assignment problem is also NP-hard [8]. Even if we assume that there is no interference in the network, one cannot compute the optimum in polynomial time. We believe, therefore, that our contributions will be of value to the research community and network planners. From this perspective, our work extends the work done in [6] where the placement of nodes is optimized in a multi-hop wireless access network.

The rest of the paper is organized as follows. In Section 2, we present a review of relevant related work in this area. Section 3 describes the basic network model, notations and design considerations used in this paper. We describe the constraints characterizing the *MC-MR* network model in Section 4, and describe the optimization functions and obtaining the upper bounds in Section 5. We present the link channel assignment and scheduling algorithms in Section 6 to derive the lower bounds. Section 7 describes our evaluation using simulations, followed by a discussion of issues that influence our network model in Section 8. We conclude the paper in Section 9.

2. RELATED WORK

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 [23], 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. The theoretical capacity of single-channel IEEE 802.11 networks has also been studied in [24] for a centralized base-station configuration for a single channel, assuming nodes can talk to at most one other node at any time. There have been various other studies regarding asymptotic capacity of multi-hop wireless networks [25, 26, 27]. A parallel work of Kyasanur and Vaidya [34] studies the asymptotic capacity of MC-MR networks, and we use those results to justify some of the capacity trends that we observed in our experiments.

Recent results in [7, 8, 9, 10] provide load-balanced routing metrics that are also used to perform load-sensitive channel assignment in *MC-MR* (multi-channel multi-radio) mesh networks. In [7, 8], distributed and centralized heuristic algorithms, respectively, are proposed for channel assignment and load-balanced routing. These papers do not consider scheduling in the presence of conflicting links assigned to the same channel, or do not discuss the optimality of their solution. While [9, 10] propose solutions to the same problem for mobile ad hoc networks, they do not consider either scheduling or the optimality of the solution. However, [9, 10] do not require knowledge of the demand vector. In our paper, we assume that the demand vector is known since optimizing rate-based objectives requires some knowledge of the network demands. The channel assignment problem considered in [7, 9, 10] deals with assigning channels to the radio interfaces, as opposed to links. We think that assigning channels to links is a better approach (similar to [8]) since radio interfaces have to switch channels in either case, however, link channel assignment modeling is more flexible.

There has been recent work on routing in multi-hop *MC-MR* mesh networks [13, 14, 16]. These studies consider only routing strategies for increasing throughput, and are based on heuristics. Link channel assignment has been studied in the context of link layer and MAC protocols for multi-channel wireless networks [15, 11, 12]. While a recent result suggests that current hardware is not yet capable of fast channel switching [28] on the time-scale of a packet, we think that this is an engineering barrier that will be overcome in the near future. Our network model provides performance bounds for all static and dynamic link channel assignment strategies.

Note that all the results in [7]-[16] seek to *improve, not optimize*, the throughput performance in the network. 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. In addition, knowledge of the upper bounds helps us to design better algorithms by cross-optimization of routing, channel assignment and scheduling. This is our goal in this paper.

The key studies that are related to the network model proposed in this paper are the papers by Jain et. al. [17], and Kodialam et.al. [18, 19]. In [17], the problem of optimal routing in the presence of interference is considered, where the interference is modeled as a conflict graph. A similar problem is also considered in [18] for optimizing routing and scheduling. Both papers model the protocol model of interference, which is based on the CSMA/CA protocol, adopted by IEEE 802.11 standards. Interference is caused when nodes in the neighborhood of a sender or receiver are active. From the perspective of network modeling, the key difference between these two papers is the complexity of modeling interference. In [17], characterizing the interference caused by a set of k links requires $O(|E|^k)$ time, while [18] requires only $O(|E|)$ time to list all the interference constraints in the network. The latter does not discuss the multi-channel multiple-radio case, while the approach proposed in [17] has super-linear complexity for specifying constraints in the *MC-MR* model considered in this paper. We, therefore, adapt the method of specifying interference constraints from [18] for use in the *MC-MR* model in this paper.

In [19], the authors propose a joint routing and scheduling scheme for a multi-channel network with one radio interface per node assuming that there are sufficient number of orthogonal channels in order to avoid interference. They study the routing, and scheduling in such a scenario. This work is extended in [20] to include full duplex channels and multiple radio interfaces. The channel assignment problem is non-existent since there is no interference at all.

In this paper, we specify a simple approach that allows us to distinguish between the communication range and carrier-sensing range, a feature that is absent in all earlier studies. Our network model thus allows us to arbitrarily specify data-link neighbors and purely interfering neighbors, and does not depend on strict mathematical graph models such as disk graphs, (r, s)-civilized graphs and planar graphs, as proposed for scheduling in [21, 22, 33]. This flexibility allows us to accommodate specialized network graphs that result due to directional antennas [29], physical obstacles, etc. In addition, we also propose extensions that enable our approach to be adopted where the radios could belong to many different standards, allowing the characterization of heterogeneous wireless mesh networks.

A parallel work in [33] 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 this paper, we address the capacity problem for general graphs, and also provide extensions that allow us to accommodate variable interference patterns that can arise in a given network topology.

In the presence of multi-radio wireless networks, one of the many ways of improving throughput is to perform striping across the multiple interfaces. The work by Hsieh et.al. [30] studies this problem, and contains an excellent survey of related work. The model proposed in our paper can be used to place bounds on the performance on any such striping protocol in a heterogeneous multi-hop multi-channel multi-radio wireless network, and as such it will be helpful to quantify their performance.

3. 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.

3.1 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[1].

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 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 802.11b standard[2], $\mathcal{C} = 3$. 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 [3]. 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 1 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.

3.2 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 [1]. 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.

3.3 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 channel on which this communication is going to take place.

Both these decisions are negotiated between neighboring nodes and then transmission takes place. The decision of which channel to communicate on, can either be done on a per time slot basis or can be fixed for the entire lifetime of the network[4]. If the channel on which communication takes place between neighbors is decided at the beginning of each time slot then we call this *Dynamic Link Channel Assignment*. If the mapping of links to channels is fixed for the entire lifetime then we call this *Static Link Channel Assignment*.

In both cases, if the number of radios (NICs) at a node is less than the number of neighbors (and the number of orthogonal channels available), then the node can be expected to switch its radios to various channels in order to make use of the multiple channels available [9]. Naturally, one would expect the throughput of static channel allocation to be lower than dynamic channel allocation due to the fact that static channel allocation is a special case of dynamic channel allocation. Under static channel assignment, however, the only decision a node has to make is to determine which neighbor (if any) it is going to communicate with in a given time slot, since the channel with which to communicate to the neighbor is already fixed and therefore need not be negotiated. We have developed algorithms for both these versions of the link channel assignment problems and we give results on the performance of both these algorithms with variable number of channels and radios.

3.4 Design Considerations

Link-Channel Restriction: A node 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 $\varrho(e)$.

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.

Channel Allocation: The network characterization developed in Section 4 is applicable to both static channel allocation and dynamic channel allocation methods. As noted in Section 2, existing constraints on hardware [28] may preclude fast dynamic switching or impose non-negligible throughput penalty. Since our goal is to characterize and seek the maximum throughput possible, we assume that

[1]The readers are referred to [18] for how unidirectional links can be modeled using a similar approach

[2]IEEE 802.11b has 12 channels, but there are only 3 channels that can operate simultaneously without mutual interfering with each other.

[3]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.

[4]or fixed for a long time until the demands change.

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 1: Index of symbols used in the paper

channel switching can be performed with a negligible overhead without affecting capacity.

Multi-path Routing: Choosing only one route between a source and destination does not exploit the inherent multipath 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.

3.5 Approach

We solve an cross-optimization problem with a given objective for an instance of a *MC-MR* multi-hop network 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 link channel assignment (static and dynamic) along with scheduling based on greedy coloring to resolve potential conflicts, and obtain a feasible schedule from the solution in step (b). The last step gives a feasible lower bound on the optimum solution, while the second step provides an upper bound.

4. CHANNEL, NODE AND INTERFERENCE CONSTRAINTS

We now present a mathematical constraint model for the *MC-MR* fixed wireless mesh network described in Section 3.

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

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

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.

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

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. We briefly outline the derivation of the interference constraint using the approach in [18], which uses these assumptions. We then extend it to cover interference links.[5]

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. Note that in the ideal case, only transmitters in the neighborhood of the receiver have to remain silent. While that case can also be modeled using the approach described here (see [18] for further details), we use only the former model in order to keep the presentation simple.

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.

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

This can be seen in Figure 2(a) for link uv and nodes $v_1, v_2 \in N(v)$.

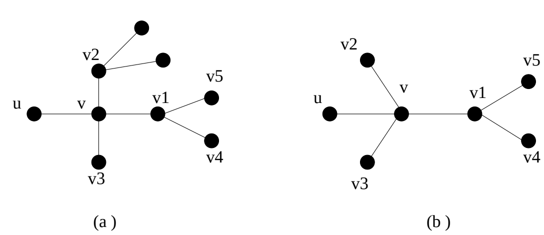

Figure 2: Interference Constraints

[5]Directional antenna interference can be modeled using the extension described later in Section 8.

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 2: Network Characterization: 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 as follows.

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

This is depicted in Figure 2(b) for link $v \leftrightarrow v1$.

Thus, we formally state the interference constraint in terms of an link e.

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

Note that there are only $|E|$ interference constraints in all, under this approach, with the number of variables per constraint never exceeding twice the maximum degree of the graph. The reduction in number of constraints significantly reduces the complexity of modeling interference and increases the convergence speed of any linear optimization problem using these constraints, as we show in subsequent sections.

Interference Links: When there is an interference link $e \in E^{\mathcal{I}}$ between two nodes v_1 and v_2 on channel i, then whenever one of the data links incident on either of the nodes is active, then the other node has to be silent for that slot on channel i. It is easy to see that if we write the previous *interference constraint* for an interference link e, then only the data links incident on either endpoints can be included. Thus, we have the following interference constraints for the general graph, where this distinction is made clear.

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

Table 2 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.

While the necessary and sufficient conditions are required to check for the feasibility of link channel assignments (see Section 6), 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

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

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 3.

We sum Equation (1) over all $t \le T$, interchange the order of summation, and divide by T to get

$$\sum_{i\in OC} \frac{\sum_{t\le T} y_i^t(e)}{T} \le \varrho(e), \quad \forall e \in E$$

Now, using Equation (5), we have the relaxed condition in terms of utilizations and link flows,

$$\sum_{i\in OC} g_i(e) = \sum_{i\in OC} \frac{f_i(e)}{c_i(e)} \le \varrho(e), \quad \forall e \in E$$

We can perform the same operations for the other two necessary and sufficient conditions in Equations (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* $\mathbf{f}$ *does not satisfy the following inequalities*

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

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

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

then the link flow set $\mathbf{f}$ *is not schedulable.*

4.1 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, as illustrated by the following example.

Consider the network shown in Figure 3, operating on a single channel with one radio per node. In this network, $K_{2m,m'}$ represents a complete bipartite graph with $2m$ vertices in one partition and m' vertices in another partition, with $m' < 2m$. There are a total of $2m(m'+1)+1$ links in the network, all with capacity 1 each. Taking into account all the necessary conditions, a maximum flow of $1/(m+m'+1)$

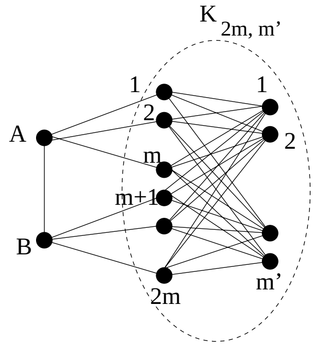

Figure 3: Unbounded Sufficiency Gap

over each link is feasible, resulting in an aggregate link utilization of

$$\frac{2m(m'+1)+1}{m+m'+1} = \frac{2(m'+1)+\frac{1}{m}}{(1+\frac{1+m'}{m})} = O(m')$$

However, only one link can be active at any time in this network due to interference. To see this, consider any link e. The end-points of e are either both A and B or one of $2m$ nodes in the middle. If $e = AB$, then none of the $2m$ nodes in the middle can be transmitting or receiving data, thereby silencing the rest of the network. If one of these $2m$ nodes is talking to either A or B, then all of the m' nodes on the right have to be silent and this in turn implies that the remaining $(2m-1)$ nodes in the middle have to be silent. The same argument applies if one of nodes in the middle is talking one of the nodes on the right. Thus the total optimal link utilization is 1. Thus, the difference between the solution that merely satisfies the necessary conditions and the optimum solution is $O(m')$, which can be unbounded as m increases.

There are other such configurations, for example, complete subgraphs (cliques), odd cycles, etc. A complete characterization of these forbidden subgraphs is not possible.[6] However, the key point is to note that *the ILP constraints in Table 2 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.

4.2 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 $\sum_{(e,i)\in S_j} y_i^t(e) \leq \beta(S_j)$, $\forall j$, while the necessary conditions (using continuous variables) have the general form $\sum_{(e,i)\in S_j} g_i(e) \leq \beta(S_j)$, $\forall j$, 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 $\mathbf{f}$ represent a link flow set $\{f_i(e)\}$, where $f_i(e)$ represents the flow on channel i over link e. If $\mathbf{f}$ does not satisfy the following inequalities*

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

then the link flow set $\mathbf{f}$ 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.

5. OPTIMIZATION: MULTI-COMMODITY FLOWS AND UPPER BOUNDS

We define a standard multi-commodity flow problem on the *MC-MR* network: a set of sources, $\mathbf{s}$, want to send data to a set of destinations $\mathbf{d}$, with a end-to-end rate demand vector $\mathbf{r}$. 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. First, we translate the constraints identified earlier from link-flow variables to end-to-end rate variables.

5.1 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[7] 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*

[6] A common characteristic among all forbidden subgraphs is the existence of cycles. It is possible, therefore, that if the graph G is a tree, then the relaxed necessary conditions might also be sufficient. This requires that there be no interference links in G, with no path redundancy. Since this is uninteresting for practical purposes, we do not pursue it further.

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

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.

Figure 4: Optimization Problem

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.

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

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 $f_i(e) = \sum_{q \leq Q} x_i^q(e)$.

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

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

Let $I(P, j)$ denote the set of (link, channel) pairs that are on path P and are also in the set S_j, i.e,

$$I(P, j) = \{(e, i) : (e, i) \in P\} \cap \{(e, i) : (e, i) \in S_j\}.$$

Since $f_i(e) = g_i(e)c_i(e)$, from Equation (9), the amount of flow permitted by set S_j on path P is given by

$$F(P, j) = \beta(S_j) \left(\sum_{e \in I(P,j)} \frac{1}{c_i(e)} \right)^{-1}.$$

The flow that can be sent on path P denoted by $F(P) = \min_{j \in \mathcal{L}} F(P, j)$. We use $F()$ to represent flow on paths and $f()$ to represent flow on links. The necessary conditions for a rate vector r to be achievable is given by

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

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

As we mentioned at the beginning of this section, there are many objectives of interest for the network capacity planner that can be solved using an optimization framework. Each objective is associated with some additional constraints that are specific to the chosen objective. For example, maximizing the sum of end-to-end rates might be associated with a minimum rate constraint for each source-destination pair.

Our characterization allows us to plug-in these constraints, in addition to necessary rate constraints identified above, into the optimization problem, and obtain an upper bound on the performance for that objective.

5.2 The Feasibility of Demands

As an example of our flexibility, we consider 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 upper bounds, and in the next section, we describe the procedure to obtain lower bounds 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.

5.2.1 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.

$$\max \lambda$$

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

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 $\lambda^* \geq L^*\tau$. The sufficiency gap created by 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 will now formulate the dual to this problem, and compute a fully polynomial time approximation algorithm (FPTAS) using a primal-dual algorithm. The FPTAS algorithm and its analysis follows the work of Garg and Könemann [31].

The dual to the LP defined in Section 5.2.1 assigns a weight $\alpha(j)$ to each set S_j in the network, and the dual variable $z(q)$ for the rate scaling constraint (the second set of constraints) in the LP.

$$\min \sum_j \alpha(j)$$

$$\begin{aligned}
\sum_{(e,i)\in P} \frac{\sum_{j:(e,i)\in S_j} \alpha(j)/\beta(S_j)}{c_i(e)} &\geq z(q),\ \forall P \in \mathcal{P}_q,\ \forall q \\
\sum_{q=1}^{Q} r(q)z(q) &\geq 1 \\
\alpha(j) &\geq 0,\ \forall j \in \{1, ..., \mathcal{L}\}
\end{aligned}$$

$w(j) = \delta$, $\forall j \in \{1, 2, ..., \mathcal{L}\}$ and $b = 0$

While $\sum_j w(j) < 1$
For $q = 1, 2, \ldots, Q$
  $r = r(q)$
  While $r > 0$
    Set weights of $\frac{\sum_{j:(e,i)\in S_j} \alpha(j)/\beta(S_j)}{c_i(e)}$
    on each pair (e, i) and compute P^*,
    the shortest path from $s(q)$ to $d(q)$.
    Let $u = \min_j f(P^*, j)$.
    $\delta = \min\{r, u\}$ $r \leftarrow r - \delta$.
    $f_i(e) \leftarrow f_i(e) + \delta$ and
    $w(j) \leftarrow w(j)\left(1 + \frac{\epsilon\delta}{f(P^*,j)}\right)\ \forall j$.
  end While
end For
$b \leftarrow b + 1$
end While
Compute $\rho = \max_j \sum_{(e,i)\in S_j} \frac{f_i(e)}{c_i(e)}$.
Output $\lambda^* = \frac{b}{\rho}$

Table 3: Primal Dual algorithm

The primal dual algorithm to solve the sizing problem starts by assigning a weight of δ to all sets S_j. The algorithm proceeds in phases. In each phase, for each commodity q, we route $r(q)$ units of flow from $s(q)$ to $d(q)$. A phase ends when commodity Q is routed. The $r(q)$ units of flow from $s(q)$ to $d(q)$ for commodity q is sent via multiple iterations. In each iteration, the path P^* from $s(q)$ to $d(q)$ is determined. Let $F(P^*)$ represent the capacity of this path. We can send a flow of at most $F(P^*)$ units this iteration. Since $r(q)$ units of flow have to be sent for commodity q in each phase, the actual amount of flow sent is the lesser of $F(P^*)$ and the remaining amount of flow to make up $r(q)$ in this phase. Once the flow is sent, the weights of the nodes that carry the flow is increased. The algorithm is shown in Table 3. Therefore, the algorithm then alternates between sending flow along shortest path pairs and adjusting the length of the links along which flow has been sent until the optimal solution is reached. Therefore the primal dual algorithm solves a sequence of shortest path problems. It can be shown that by choosing ϵ and δ appropriately, we can get as close to the optimal solution as desired. The running time increases with the accuracy needed. The next Theorem states the running time and the correctness of the algorithm shown in Table 3. The proof is omitted here due to lack of space, but is similar to the one in [31].

THEOREM 4. *The algorithm in Table 3 computes a* $(1-\epsilon)^{-3}$ *optimal solution to the flow scaling problem in time polynomial in* $Q, \mathcal{L}, n$ *and* $\frac{1}{\epsilon}$*, where* Q *is the number of commodities,* $\mathcal{L}$ *is the number of constraining sets, and* n *is the number of nodes.*

6. LINK CHANNEL ASSIGNMENT AND SCHEDULING

The linear program described above 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 and we use variations of the greedy approach to solve the problem. There are two versions of this problem that we solve in this paper. The first one is the Static Link Channel Assignment Problem where link channel assignments are made to the links at the beginning and will remain fixed over all time slots. In the second version of the problem, the assignment of channels to links is done every T_d slots ($T_d \geq 1$). This is the Dynamic Link Channel Assignment problem. The dynamic link channel assignment results for the case of $T_d = 1$ will provide a basis for comparison of the performance of all link channel assignment algorithms, since it provides the maximum flexibility in link channel assignment and scheduling. For the sake of simplicity, the following discussion assumes that only one channel can be allotted to a given link, i.e., a (node,neighbor) pair. In other words, $\varrho(e) = 1, \forall e \in E$. Our algorithms can be easily generalized for $\varrho(e) > 1$.

6.1 Static Link Channel Assignment

We first use an (almost) greedy approach in solving the static channel assignment problem. Once this link channel assignment is done, time-slots are assigned to the each channel using a coloring algorithm. This assignment of time slots to channels is done almost independently for each channel. The channels interact during assignment of time-slots only to resolve the *node-radio constraint* (Equation (2)) at each node in the network.

Recall that network model constraints are modeled by the sets $S_1, S_2, \ldots, S_{\mathcal{L}}$. These sets are specified by (link, channel) pairs. If the link channel assignment results in a large number of channels being assigned to one set, then a large number of time slots will be needed to resolve the conflicts in this set. This will result in an inefficient schedule. Therefore the main idea behind the *Balanced Static Channel Assignment (BSCA)* algorithm is to ensure that none of the constraint sets are loaded by any channel. Therefore, we track the total load that is assigned to each constraint set. At the end of solving the linear programming problem, let $\hat{x}_i^q(e)$ represent the flow on link e corresponding to commodity q and channel i. We first determine

$$f_i(e) = \sum_{q \leq Q} \hat{x}_i^q(e).$$

This represents the total flow (load) on channel i corresponding to link e. Given a particular link e, let $T(e,i)$ denote the constraint sets that contain the pair (e,i). Let l_S denote the total flow that has been assigned to constraint set S. This is initialized to zero for all sets. At a generic step in the algorithm, assume that we want to assign a channel to link e. (We will soon elaborate on how this link is picked.) We first determine, for each channel $i \in OC$, $m(e,i) = \max_{S \in T(e,i)} l_S$, representing the maximum load on any constraint set that contains the (link, channel) pair (e,i). We then assign link e to channel j that attains the minimum value of $m(e,i)$ over all channels, i.e, we assign e to channel j where $j = \text{Arg} \min_i m(e,i)$. Once this is done the flow on all the sets in $T(e,j)$ are incremented by $\sum_i f_i(e)$ since all the flow on link e gets assigned to channel j. As stated earlier, the main intuition behind the scheme is to not allow any interference set to be overloaded with any channel. This is the reason for picking a min-max allocation, since it ensures that the load on the constraint sets is too distributed as much as possible among the given channels.

We experimented with three different methods of choosing link e.

- Pick a link at random.
- Pick links in a predetermined order.
- Pick a link whose $\min_i m(e,i)$ is the lowest.

From experimenting we determined that the third option outperforms the first two and all the results reflect this implementation. The Balanced Static Assignment algorithm is described in Figure 5.

$l_S = 0$, $\forall$ constraint sets S; $A = \emptyset$
$p(e) = \sum_i f_i(e), \forall e \in E$

While $\sum_{e' \in E} p(e') > 0$
  For $e \in E \setminus A$
    $m(e,i) = \max_{S \ni (e,i)} l_S, \quad \forall i \in OC$
    $w(e) = \min_i m(e,i)$
    $b(e) = \arg\min_i m(e,i)$
  end For
  $l = \min_{e \notin A} w(e)$
  Assign l to channel $b(l)$
  $l_S = l_S + p(l), \forall S \ni (l, b(l))$
  $f_{b(l)}(l) = p(e)$; $f_i(l) = 0, \forall i \neq b(l)$
  $A = A \cup l$; $p(l) = 0$
end While

Figure 5: Balanced Static Channel Assignment

Greedy Scheduling: At the end of static link channel assignment, we have a set of flows assigned to links that have been assigned to a particular channel. We now scale all the flows so that they are integral. We multiply all flows by a large number M and ignore the fractional part. We now assign time slots to each channel separately, while taking care that the *node-radio constraint* is met, ensuring that the number of active links incident on a node in a given time slot is always at most the number of radios at the node. This is done via a greedy coloring algorithm applied to each channel separately.

Let a link e be assigned to channel i with a scaled flow $f'_i(e)$. Let $T_1(e,i)$ be the collection of constraint sets that contain the pair (e,i). The greedy coloring algorithm is as follows.

1. Consider the link e with the highest residual flow $f'_i(e)$.
2. Assign the smallest color (time-slot), k, that does not occur more than $\beta(S_j)$ times in the set $S_j \in T_1(e,i), \forall j$.
3. Add k to all sets in $T_1(e,i)$.
4. Reduce the scaled flow by $c_i(e)\tau$, where τ is the length of a time slot.
5. Repeat until all flows have been scheduled.

If NS denotes the maximum number of time slots taken by any channel, then the demand met by this system is given by

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

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

6.2 Dynamic Link Channel Assignment

In static link channel assignment, we assume that each link is assigned to a channel and cannot switch channels during different time slots. In dynamic link channel assignment, we assume 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 $T_d = 1$, dynamic link channel assignment gives the highest flexibility to maximize the achievable performance for any link channel assignment scheme.

Since the problem of determining the optimal link channel assignment is NP-hard [8], we use a packing based heuristic to approximate the optimal solution. Let $f_i(e) = \sum_{q \leq Q} \hat{x}_i^q(e)$ represent the desired flow on link e corresponding to channel i. Once we get the solution from the linear programming problem, we scale all the link flows to make them integral. If the all the data is rational, then the solution of the LP is rational and therefore can be scaled to integers. Since we are using a primal-dual scheme, the answers will be fractional and therefore we scale the flows by a suitably large value M so that the eliminated fractional portion is negligible. (We typically scale by 100 in the experiments.) Once we get these flows, the objective is to assign these flows to channels in as few time slots as possible.

The main idea behind the *Packing Dynamic Channel Assignment (PDCA)* algorithm is to pack the flows in a greedy manner in each time period. 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 $f^M(e) = \sum_i f_i(e)$. 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 period T_d, 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. If there are still unassigned flows, we move to the time-period and repeat the link 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 period of length T_d, 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 PDCA algorithm shown in Figure 6, 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. As in the static channel allocation, the demand met by the system is computed as

```
NS = 0; A = E; d(e) = f^M(e)  ∀e ∈ E

While A ≠ ∅
    NS ← NS + T_d
    Assume links are numbered in decreasing
    order of d(e)
    z_S = 0  ∀ constraint sets S.
    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
            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
```

Figure 6: Packing Dynamic Channel Assignment Algorithm

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

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

7. PERFORMANCE EVALUATION

In this section, we present results of our performance evaluation based on 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 (dynamic) PDCA algorithm for $T_d = 1$ with respect to the capacity of the network
3. The performance of the simpler (static) BSCA algorithm along with greedy scheduling with respect to the dynamic link channel assignment algorithm.

We evaluated our algorithms for various random topologies, grids, full meshes, and other types of topologies. We present the results for random topologies and grids here. As mentioned earlier in Section 6, we set $\varrho(e) = 1, \forall e \in E$. We also only considered networks with no separate interference links, i.e., $E^{\mathcal{I}} = \emptyset$. In all the experiments, 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.

7.1 Grid Topology

We consider a 5×6 grid topology with 30 nodes. Each node has at most 4 neighbors in the grid. We divide the grid into four quadrants and assign one node in each quadrant to be a sink node for flows. We then assign for each node in the grid the closest sink node as the destination for its flow. All flows have a demand of 1 unit.

We vary the number of flows in the grid from 5 to 25. We measure three parameters for each set of flows: (a) the capacity upper bound, (b) the BSCA (static assignment) lower bound, and (c) the PDCA (dynamic assignment) lower bound. All these values are in terms of the scaling factor λ by which the demands have to scaled in order to make the scaled demand vector feasible in the network. We present the results in Figures 7, 8, 9. Each data point is the average of the performance attained by varying the number of flows $(5, 10, 15, 20, 25)$ in the grid.

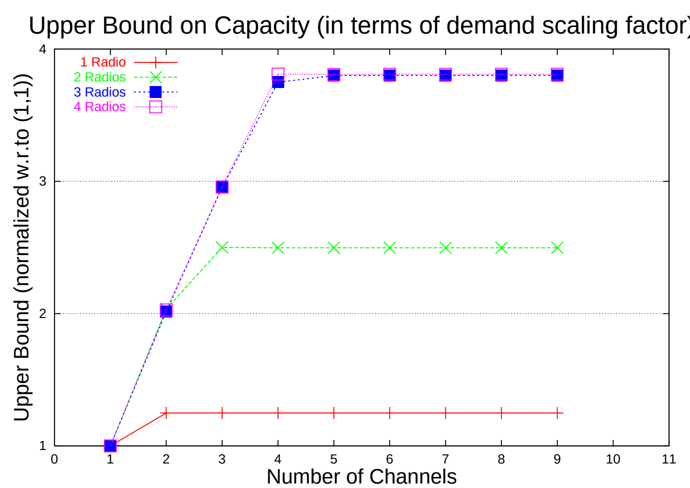

Figure 7: Grid: Upper Bound on Capacity
The y-axis is normalized w.r.t. the upper bound for the case of 1 Radio per node and 1 Channel.

Since the maximum degree of a node in the grid is only 4, we vary the number of radios from 1 to 4, and the number

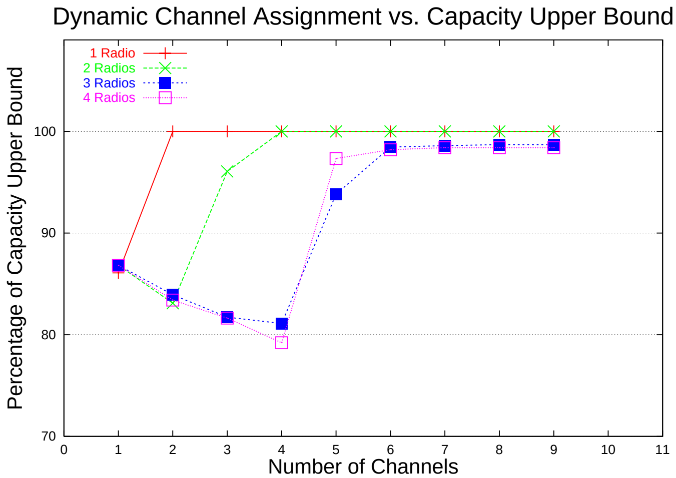

Figure 8: Grid: Dynamic Link Channel Assignment

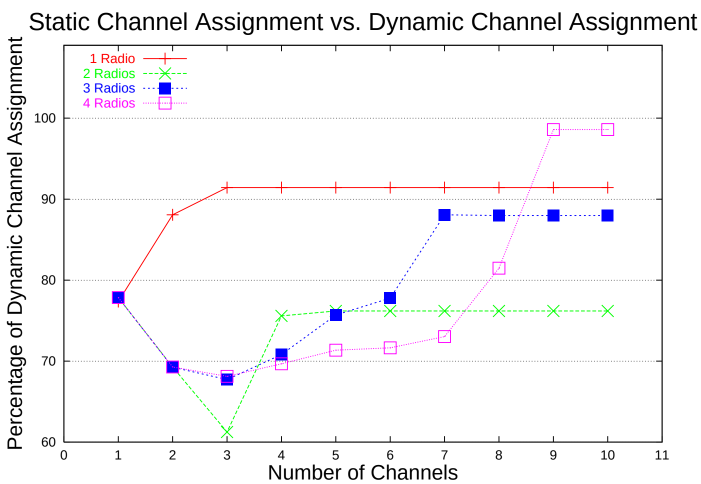

Figure 9: Grid: Static Link Channel Assignment

of channels from 1 to 10. From Figure 7, it can be seen that capacity curve is convex as expected. However, an interesting observation emerges from this data. Given κ number of radios per node, the capacity increases almost linearly up to $\kappa + 1$ channels. This is counter-intuitive since one would expect that the near-linear capacity increase will stop after κ channels. The results seem to indicate that this trend continues for one more additional channel. We do not observe this pattern when we have 4 radios since the maximum degree of nodes is only 4, and we can assign only one channel for each link to a neighbor ($\varrho(e) = 1$).

In Figure 8, we compare the performance of the PDCA algorithm for $T_d = 1$, with respect to the upper bound. Recall that the performance of the channel assignment algorithm is given by $\frac{M}{NS}\lambda^*\mathbf{r}$, where $\lambda^*\mathbf{r}$ is the upper bound from the LP. Thus, we plot $\frac{M}{NS}$ as a percentage in this plot. Once again, the values are averaged over multiple flow sets. It is evident that when channels are assigned dynamically every time slot, the PDCA algorithm performs within 80% of the capacity upper bound. The algorithm in fact achieves close to the capacity when the number of channels exceeds the number of radios by 2. This is a result of the flexibility in packing afforded by the greater number of channels. Even if the performance of the PDCA gets relatively worse when the number of channels initially increases, it must be noted that the degree of decrease is always less than 10%.

In Figure 9, we evaluate how the static BSCA algorithm performs with respect to the PDCA algorithm. It can be seen that BSCA algorithm does not really match PDCA even when if we increase the number of channels. However this is to be expected since once we bind the channels, the greedy coloring-based scheduling algorithm will account for most of the sub-optimal behavior. Note that when there is only one channel in the network, the greedy coloring is the cause for the performance penalty. While this penalty goes down when more channels are available, it still remains a significant component. In spite of this, the BSCA algorithm performs within 60% of the PDCA algorithm.

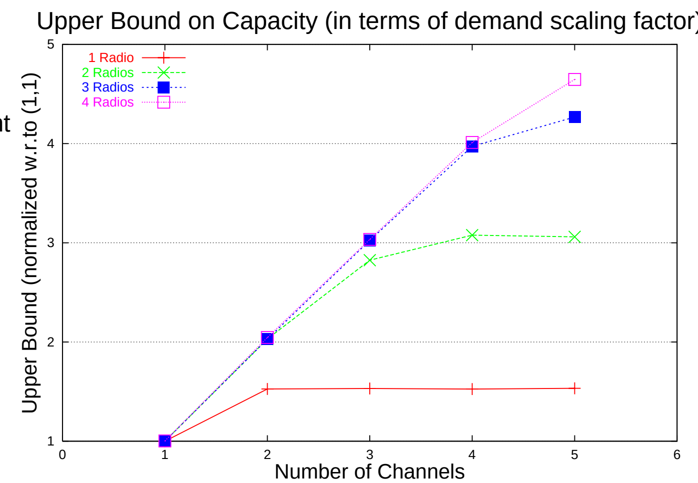

Figure 10: Random Graphs: Upper Bound on Capacity

The y-axis is normalized w.r.t. the upper bound for the case of 1 Radio per node and 1 Channel.

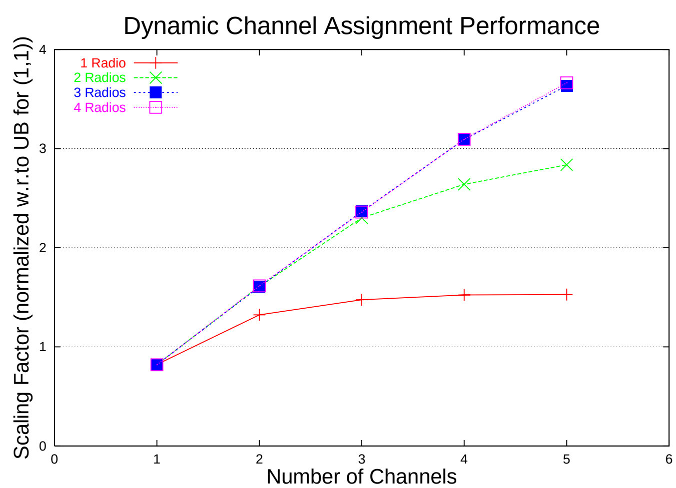

Figure 11: Random Graphs: PDCA

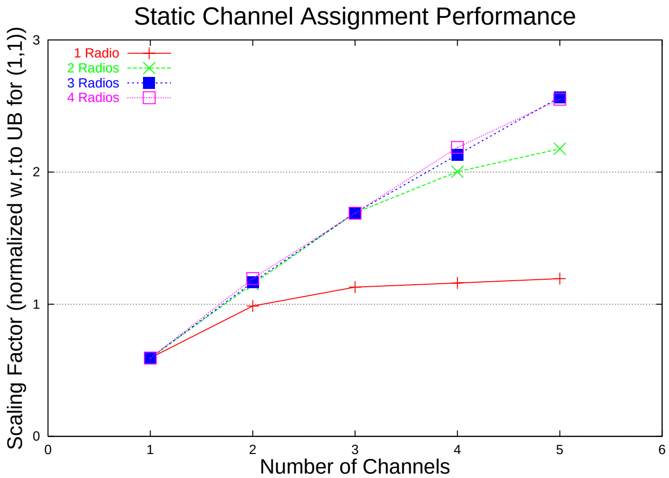

Figure 12: Random Graphs: BSCA

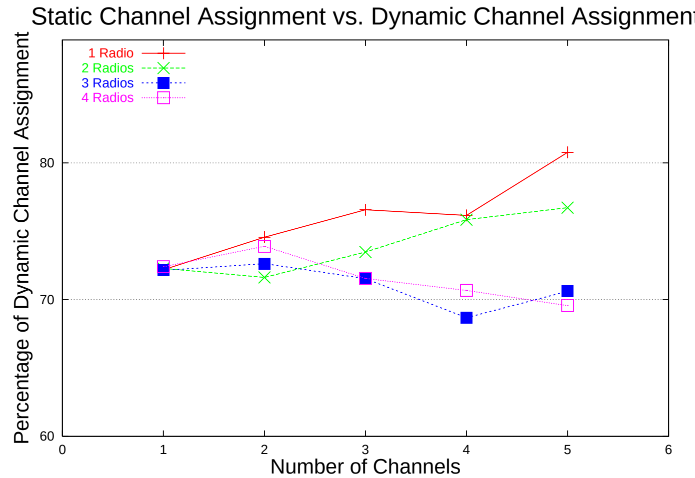

Figure 14: Random Graphs: BSCA vs. PDCA

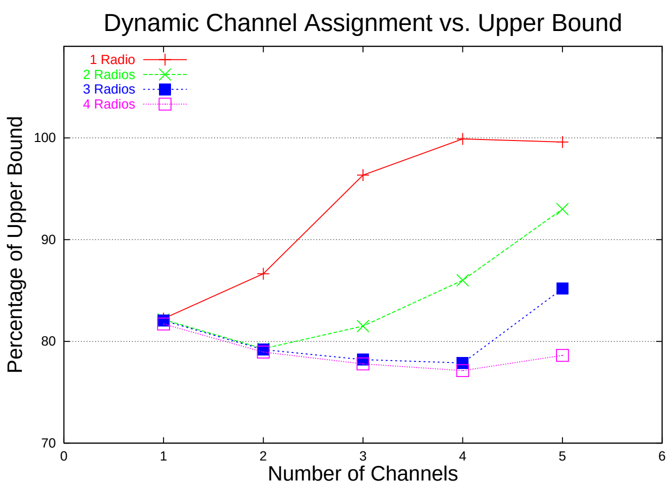

Figure 13: Random Graphs: PDCA vs. Capacity

7.2 Random Graph Topology

We also tested the PDCA and BSCA algorithms on randomly generated connected graph topologies. The sources and destinations are randomly chosen in this scenario as opposed to the grid described earlier. We varied the number of nodes, links, and flows in the graphs and measured correlation between various graph parameters such as maximum and average degree, maximum and average path length of flows, and connectivity of the graph to the solutions generated by the linear program, PDCA and BSCA algorithms. We tested ten random graph topologies. The number of nodes in random graphs varied from 15 to 50, with the average node degree varying from 3 to 9. The maximum degree of the graph varied from 7 to 20, while the connectivity of the graphs varied from 1 to 4. The path length of the flows range from 1 to 10. Our observations suggest that the relative performance of the algorithms with respect to the linear program was not dependent on any combination of the above graph parameters.

The results from these experiments are presented in Figures 10, 11, 12. The y-axis in all these figures show the performance (feasible demand scaling factor) in normalized with respect to that of the capacity upper bound for (link, channel) $= (1, 1)$. Each data point is averaged over the 10 random graph topologies.

Figure 10 shows that the capacity increases almost linearly with κ radios for up to $\kappa + 1$ channels. This suggests that in order to make the best use of available $\mathcal{C}$ orthogonal channels, then we may need at least $\mathcal{C} - 1$ radios per node. The gains in capacity diminish quickly beyond this point. Based on a parallel work by Kyasanur [34], we observe that in this model, one can expect a linear increase in capacity as long as the ratio of $\mathcal{C}/\kappa$ is $O(\log n)$. Our results tend to agree with this observation in both the grid and random graph topologies, even though large-scale simulation results are needed to confirm this.

Figures 11 and 12 show the performance of PDCA and BSCA for the random graphs. The results show that while the trend in capacity improvement is similar to that of the upper bounds, the PDCA algorithm needs more channels to achieve the same gain as the upper bound, and BSCA requires even more channels than PDCA. Interestingly, in both algorithms, the performance with 3 or 4 radios is similar for up to 5 channels. The performance in this case will differ when more channels are included.

In Figure 13, we compare the PDCA algorithm with the capacity upper bound, and find that it achieves 75% of the upper bound on the average. The worst case performance that we observed with PDCA in all tested topologies was 55% of the capacity upper bound. Another interesting fact is that when we have less channels than the number of radios per node, then the relative performance of the link channel assignment algorithms decreases with increasing number of channels. The relative performance improves steadily beyond a certain point until it saturates at a latter point. This pattern is observed even in the results for the grid topology of the previous example, and for both PDCA and BSCA. We think that this reflects the inefficiencies in our algorithms, and it is possible to devise better algorithms that can bridge this performance gap. We show similar results for the performance of BSCA versus that of PDCA in Figure 14. The BSCA algorithm performs, in the worst case, within 50% of the PDCA algorithm, based on our experimental observations.

It is important to note that the small gap between the capacity upper bound and the PDCA performance implies not only that PDCA performs close to capacity, but also that the linear program models the achievable capacity with good accuracy. Thus, the relaxation of the integer constraints in Equations (1), (2) and (4) to flow constraints in Lemma 1 does not penalize our model by a lot for practical instances of wireless mesh networks.

8. DISCUSSION

We now discuss some directions into which our network modeling can be extended, present some issues arising in *MC-MR* networks and modeling their behavior, and identify areas for future work.

8.1 Extensions to the Network Model

In a practical wireless network scenario, there might be directional antennas at some nodes, and the network might use heterogeneous radios as part of the multiple-orthogonal channel regime. For example, one might have 802.11b and 802.11a radios co-existing in the network, for a total of $3 + 12 = 15$ orthogonal channels. However, if a node has two 802.11b radios and one 802.11a radio, then the two 802.11b transceivers have to share the three channels only, while the one 802.11a radio has the flexibility of choosing any one of the 12 channels. These restrictions have to be modeled in any network characterization.

Heterogeneous Radios: Let there be $\mathcal{M}$ different radio systems in the network. In our model, the interference constraints remain unchanged, as they are specified for each individual channel. We have to rewrite the *link-channel* and *node-radio* constraints given by Equations (1) and (2) respectively. For radio system $j = 1, 2, ..., \mathcal{M}$, let $OC_j, \mathcal{C}_j, \varrho_j(e), k_j(v)$ be the channel set, number of channels, maximum number of channels over link e, and number of radios at node v, respectively for that particular system j. Then we can substitute these radio-system indexed variables in place in the *link-channel* and *node-radio* constraints, and specify them for every radio system in the network to obtain the constraints for a heterogeneous radio network. The relaxation to continuous variables follows similarly using radio-system indexed parameters. However, the link channel allocation algorithms will have to be redesigned to split link flows between various radio systems.

Directional Antennas: Directional antennas are used to reduce interference between links, apart from other uses. The impact of including directional antennas is felt in the *interference constraints*. The constraint described in Equation (4) represents a group of mutually interfering links. With directional antennas leading to directional data and interference links, this is no longer true. Instead of considering all links incident on two neighbors as mutual interferers, we now have to treat a subset of them as mutual interferers. To deal with this, we combine the approach proposed here with the modeling used in [17]. We can take the set of links incident on either endpoint of a given link e as a subgraph, and derive the *conflict graph* [17] for this restricted subgraph. We then compute the maximal cliques, and can write the interference constraints for each link represented in the clique following the procedure in [17]. Since this is done on a smaller subgraph, this approach is more scalable than the original approach in [17].

The key observation here is that the proposed model is extremely flexible, and accommodates a range of network constraints.

8.2 Issues

The authors in [14] report that there are issues with full-duplex operation on multiple orthogonal channels with multiple radios on the same node when the radios are homogeneous. This is due to the power effects in the internal electronics of the node, and is not due to channel effects. Since this is an issue with existing implementations, we think that it can be fixed in the next generation of chips, as it is the only way that one can take advantage of multiple radios and channels.

In place of the protocol model of interference followed in this paper, one could also consider the physical model of interference [5], which considers the cumulative interference from all nodes in the network at a given pair of communicating nodes. However, obtaining a deterministic constraint model for the entire network under this scenario is hard, and deterministic scheduling under this interference model is hard to approximate [32]. A different approach is required to address the problem of optimal routing and scheduling in the physical interference model.

Scheduling algorithms for multi-hop multi-radio networks can be expected to be more complex than existing work for single radio shared channel networks, due to the complex interactions between the nodes, channels and the interference regions, as highlighted by our network model. We have proposed a greedy approach to scheduling in this paper, which has been shown to work well in existing single-radio networks. However, the performance of this algorithm are not as good for multi-radio multi-channel networks. We pose this challenge of developing strong scheduling algorithms for this case as an open problem, which requires careful investigation.

Channel assignment algorithms become much more complex with the different number of parameters, $\varrho(e), \kappa(v), \mathcal{C}$. When $\varrho(e) = 1$, we provided link channel assignment algorithms that perform very well on the average. However, for any $\varrho(e)$, or when we have different radio systems coexisting in the network, one of the questions is that given a desired flow between links, is it optimal to split the flow across multiple channels, which can cause interference at all neighbors but for a small amount of time on each channel, or is it optimal to send all the flow on one channel, which allows more interference-free channels on neighboring links. This is a fertile area for further research.

The proposed algorithms in this paper are centralized algorithms and will be useful as a benchmarking tools to quantify the performance of routing, link channel assignment and scheduling algorithms. It is not clear if it is possible to jointly optimize routing, link channel assignment and scheduling in a distributed manner. Distributed algorithms in a wireless network must address many new issues such as synchronization, state dissemination, and coordination. This is a wide open area for research.

8.3 Ongoing and Future Work

In our investigation of network capacity here, we have assumed that the channel switching overhead for a radio is negligible. However, until the development of high-performance radio interfaces that make this assumption a reality, switching penalty is here to stay and will affect the throughput. We plan to investigate models to characterize channel switching delay among radio interfaces and look into link channel assignment and interface-channel switching algorithms to minimize switching overhead without sacrificing throughput.

Our simulations have assumed equal capacity across channels for all links. It will be interesting to see how the behavior seen in the simulations will vary when we introduce multi-rate channels across links. We plan to investigate this as part of future work. We are currently evaluating the effect of interference links on the capacity of the network and the performance of the link channel assignment algorithms.

9. CONCLUSION

We presented a network characterization that captures the constraints associated with multi-channel multi-radio *(MC-MR)* multi-hop wireless mesh networks. We showed that our model is extensible and can cover a wide variety of cases that reflect a range of practical constraints. We then presented an algorithm that computes the optimal routes for a given objective of meeting a set of demands in the network using a set of necessary conditions as constraints. The approach used here can be applied to optimize many other objective functions. We also propose two link channel assignment algorithms, one static and the other dynamic, which allow us to schedule flows on the links in the network. We demonstrated through simulations that our proposed routing, link channel assignment and scheduling algorithms are able to characterize network capacity and achieve a performance that is close to optimal.

In conclusion, we hope that the methods proposed 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.

10. REFERENCES

[1] IEEE Std. 802.11-1999, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 1999.

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

[3] Microsoft Research, "Mesh Networking", *http://research.microsoft.com/mesh/*.

[4] Mesh Dynamics Inc., *http://www.meshdynamics.com*.

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

[6] Qiu, L., Chandra, R., Jain, K., and Mahdian, M., "Optimizing the Placement of Integration Points in Multi-hop Wireless Networks", *IEEE ICNP*, 2004.

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

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

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

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

[11] So, J., and Vaidya, N., "Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver", In *ACM Mobihoc*, 2004.

[12] Tzamaloukas, A., and Garcia-Luna-Aceves, J.J., "A receiver initiated collision-avoidance protocol for multi-channel networks", In *IEEE Infocom*, 2001.

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

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

[15] 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.

[16] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A Multi- Radio Unification Protocol for IEEE 802.11 Wireless Networks", *IEEE Broadnets*, 2004.

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

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

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

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

[21] 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.

[22] 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.

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

[24] Cali, F., Conti, M., and Gregori, E., "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit", *IEEE/ACM Transactions on Networking*, 8(6), pp. 785-799, 2000.

[25] 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.

[26] 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.

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

[28] Chandra, R., Bahl, V., and Bahl, P., "MultiNet: Connecting to multiple IEEE 802.11 networks using a single wireless card", *IEEE Infocom*, 2004.

[29] Choudhury, R., and Vaidya, N., "Impact of Directional Antennas on Ad Hoc Routing", In *IEEE Personal Wireless Communication Conference*, Sept. 2003.

[30] Hsieh, H.-Y., Kim, K.-H., Zhu, Y., and Sivakumar, R., "A Receiver-Centric Transport Protocol for Mobile Hosts with Heterogeneous Wireless Interfaces", In *ACM Mobicom*, 2003.

[31] Garg, N., and Könemann, J., "Faster and Simpler Algorithms for Multi-commodity Flow and other Fractional Packing Problems", *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pp.300-309, 1998.

[32] S. Borbash and A. Ephremides, "Wireless Link Scheduling with Power Control", *IEEE WiOpt '04*, March 2004.

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

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