Characterizing Achievable Rates in Multi-hop Wireless Mesh Networks with Orthogonal Channels

Murali Kodialam, Thyaga Nandagopal
Bell Laboratories, Lucent Technologies
Holmdel, NJ 07733, USA
{muralik, thyaga}@bell-labs.com

Abstract

This paper considers the problem of determining the achievable rates in multi-hop wireless mesh networks with orthogonal channels. We classify wireless networks with orthogonal channels into two types, half duplex and full duplex, and consider the problem of jointly routing the flows and scheduling transmissions to achieve a given rate vector. We develop tight necessary and sufficient conditions for the achievability of the rate vector. We develop efficient and easy to implement Fully Polynomial Time Approximation Schemes for solving the routing problem. The scheduling problem is a solved as a graph edge-coloring problem. We show that this approach guarantees that the solution obtained is within 50% of the optimal solution in the worst case (within 67% of the optimal solution in a common special case) and, in practice, is close to 90 % of the optimal solution on the average. The approach that we use is quite flexible and can be extended to handle more sophisticated interference conditions, and routing with diversity requirements.

1 Introduction

Wireless multi-hop networks have attracted lot of attention in recent years as the next evolutionary step for wireless data networks. These networks have evolved into two distinct classes: fixed mesh networks and mobile ad hoc networks. Nodes in the former are fixed in one place and are typically endowed with more power resources than nodes in a mobile ad hoc network. However, in both these network classes, wireless nodes possess limited communication capabilities. One of these limitations is the number of neighbors that these nodes can communicate with simultaneously. This is primarily determined by the channel model: a single shared channel or a set of orthogonal communication channels. From a medium access protocol perspective, in the former channel model, interference from ongoing neighboring transmissions on the same channel can prevent successful transmissions between two nodes, while in the latter, neighboring transmissions are possible using different (orthogonal) channels.

In this paper, we consider the fundamental problem of characterizing the rates that are achievable in a multi-hop wireless network that uses orthogonal channels for communication between nodes. This problem has been studied earlier in a limited context in [1], where each node is allowed to be in communication with at most one other node at any time instant. The paper develops fast and efficient polynomial-time approximation algorithms for the joint routing and scheduling problem. We generalize this model in this paper to allow for a node to engage in multiple simultaneous communications with its neighbors. This extension is non-trivial due to the various constraints placed by system design and other physical layer considerations. Our main objective is, as in [1], to develop efficient algorithms that perform end-to-end flow routing and link scheduling in any instance of a multi-hop wireless network with orthogonal communication channels. The availability of multiple orthogonal channels in existing and proposed wireless standards [2, 3] and the use of multiple radios on a single wireless device [4] have increased the importance of the kind of studies undertaken in this paper.

Developments in CDMA and MIMO [5] systems have enabled wireless nodes to engage in simultaneous communication on multiple channels at the same time. Theoretically, nodes can simultaneously transmit and receive using a Full Duplex transceiver. However, system design constraints might restrict nodes to Half Duplex operation. In this paper, we consider each of these cases separately and analyze the achievable rates in such networks. An interesting aspect of the framework developed in this paper is that it also allows us to characterize networks with a mix of Full Duplex and Half Duplex nodes.

The seminal paper of Gupta and Kumar [6] shows that the throughput per node in a multi-hop wireless networks with n nodes scales as $O(\frac{1}{\sqrt{n}})$ bit-meters/second. This asymptotic throughput bound holds under fairly general conditions for networks using a single shared channel. In this paper, however, we are interested in networks with orthogonal channels and derive the bounds on the actual performance for a given node configuration. While the asymptotic results in [6] apply for orthogonal channels, we investigate the following question: In a given multi-hop wireless network with specified node configurations, communication constraints, and wireless link speeds, is a given rate vector between multiple source-destination pairs jointly achievable ?

This problem is analogous to the multi-commodity flow problem [7], and is non-trivial due to the fact that this problem involves jointly solving a routing and scheduling problem. We characterize the achievable scheduling space first, and then solve the routing problem over the achievable scheduling space.

Our contributions are as follows.

- We characterize the achievable scheduling space under various communication models for multi-hop wireless networks with orthogonal channels.
- We address the problem of determining if a given set of source-destination rates are achievable or not and, if achievable, we derive efficient, simple to implement algorithms to compute the end-to-end routes and the per-link flows.
- We provide efficient polynomial-time graph edge-coloring algorithms for computing schedules for any given set of achievable source-destination rates.

We summarize some of the related work in Section 2. In Section 3, we describe the network and channel models. We characterize the necessary and sufficient conditions for achieving link flows in Section 4. We describe the multiple source-destination flow problem in Section 5. Simulation results are presented in Section 6. We conclude the paper in Section 7.

2 Related Work

In [1], the problem of achieving a given rate vector in a multi-hop wireless network with orthogonal channels is studied with the constraint that a node can communicate with at most one other node at any time instant. We use a linear programming formulation to characterize the schedulable space of such a network, and solve the multi-commodity flow problem over this space. Finding the link schedules is reduced to a multi-graph edge coloring problem, which is NP-complete. However, by using well-known efficient coloring algorithms, we guarantee 67 % of the optimal solution to compute efficient link schedules in $O(mn)$ time, where n is the number of nodes and m is the number of links in the network. In this paper, we extend the work in [1] in two directions.

- We generalize the approach to the Full Duplex systems where we derive routing and scheduling algorithms for achieving a rate vector.
- We consider the case where a node has multiple receive channel elements that enables it to receive transmissions from multiple neighbors simultaneously.

In an earlier work, Hajek and Sasaki [8] consider the link flow achievability problem in the case of a spread spectrum system, similar to the model in [1]. The authors showed that the problem can be solved in polynomial time using fractional coloring [8]. They formulated the problem as a linear optimization problem on the fractional matching polyhedron and used the ellipsoid algorithm [9] to solve the problem. While the solution can be found optimally, the trade-off is in the running time of the algorithm, which is roughly $O(mn^4)$, using a separation oracle as a subroutine in the ellipsoid algorithm. As the authors point out in their paper, the algorithm developed is not practical. Several authors including Post, Kershenbaum and Sarachick [11], Wieselthier, Barnhart, and Ephremides [14] have considered heuristic procedures for this problem.

The authors in [8] also consider the problem of achieving a given set of source-destination rates. This problem, in theory, can be formulated as a multi-commodity flow problem on the fractional matching polyhedron. It can be solved in $O(n^7)$ polynomial time. The approach would not be practical to solve even small sized problems. Further, their approach cannot be extended to interference limited systems.

Jain et. al. [12] have recently proposed a model that uses a linear programming approach to characterize networks with interference. They use a conflict graph to model constraints on simultaneous transmissions. They focus on the routing component alone, and assume that the existence of an ideal scheduling mechanism. Moreover, they do not propose any approximation algorithm with guaranteed performance to solve the routing problem. In a parallel work [13], we studied the same problem, and have characterized the necessary conditions on the achievable scheduling space. We have also proposed approximation algorithms that solve both the end-to-end flow routing problem and the link scheduling problem near optimally.

Our approach in this paper is similar to the one is [1, 13]. We first solve the link flow achievability problem by characterizing the necessary and sufficient conditions for the achievability of a link flow vector, for both Full Duplex and Half Duplex systems. This approach naturally extends to the problem where we have to determine the maximum data rate that can be sent between two nodes in the network. In the case of determining the maximum rate, the problem has a routing component and a scheduling component. The necessary conditions from the link scheduling problem give rise to constraints on the routing problem that are needed for schedulability. This is a natural generalization of the maximum flow problem in capacitated networks. We also consider the problem of determining if a given rate vector is achievable.

The approach that we take to solve the routing component of the problem is to formulate it as a linear programming problem with an exponential number of variables. We then use a primal-dual approach to develop a Fully Polynomial Time Approximation Scheme (FPTAS). The idea in FPTAS is to obtain an ϵ optimal solution to the problem. An ϵ optimal solution to the (maximizing) problems that we consider in this paper is a solution to the problem that has a value at least $(1-\epsilon)$ times the optimal solution. An FPTAS is a family of algorithms that finds an ϵ-optimal solution in time that is a polynomial function of the problem parameters and $\frac{1}{\epsilon}$. The problem parameters in our case are the number of nodes in the network n, the number of links in the graph m, and the number of source-destination pairs. The idea of solving linear programming problems approximately with FPTAS originated with the work of Shahrokhi and Matula [15]. There were a series of papers improving and extending these results. This paper follows the algorithm and analysis in Garg and Könemann [16] and Karakostas [18]. For the ease of notation, throughout this paper we use $\mathcal{O}(f)$ to represent $f \log^{O(1)} m$, i.e., we hide polylog terms in $\mathcal{O}$.

3 Model and Assumptions

We consider a multi-hop wireless network with n nodes. The nodes communicate with each other via wireless links. Each node in the network can communicate directly with a subset of the other nodes in a network. If node u can transmit *directly* to node v, we represent this fact by a directed edge (link), $u \rightarrow v$, from node u to node v. We assume that there are m links in the network. We represent the nodes in the network and possible communication with a directed graph $G = (V, E)$ where V represents the set of nodes in the network and E the set of directed edges (links) in the network. We do not assume that links are bi-directional.

We assume that the system operates in a synchronous time-slotted mode. In the most general model, one can assume that a node can transmit to or receive from multiple nodes in any given time slot. In most multi-hop networks, the operating point of the system is typically in the linear portion of the power rate curve. This implies that the achieved rate scales linearly with the power. Instead of allowing a node to transmit to multiple nodes in a given time slot by splitting the power at the node, it is possible to achieve the same mean rate by

- Allowing transmission to at most one node at peak power in any given time slot.
- Allocating slots proportional to desired link rates.

Therefore, in any time slot, we assume that a *node will transmit to only one user at peak power.* Given a link $e \in E$, we use $t(e)$ to represent the transmission end of the link and $r(e)$ to be the receiving end of the link e. We say that a link e is *active* when there is a transmission from $t(e)$ to $r(e)$. We assume that link e can transmit data at $c(e)$ bits/second. Therefore we implicitly assume stationary channel conditions and transmission at maximum power at each node. An example of a network along with the link parameters are shown in Figure 1.

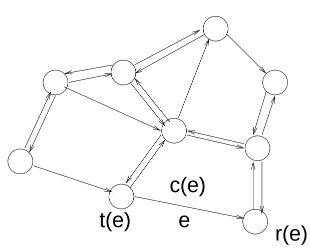

Figure 1: Network Example

We assume that node v can receive transmissions from at most $\Omega(v)$ nodes in any time slot, where $\Omega(v) \geq 1, \forall v \in V$. This is determined by the number of receive channel elements at the node. We consider two different communication modes in this paper.

- **Half Duplex Mode:** If a node does not have a duplexer, then the node can either transmit or receive in each time slot but not both in each time slot.
- **Full Duplex Mode:** If the node is operating in Full Duplex mode, then the node can transmit and receive in the same time slot.

The length of each time slot is τ seconds. Therefore, if a link is active for one time slot, then $\tau \cdot c(e)$ bits will be transmitted from $t(e)$ to $r(e)$ in one time slot. In the computation of the schedule for nodes, we assume that the schedule is periodic and has T time slots in each period. We label the time slots $1, 2, \ldots, T$. If the system is asynchronous, then the results in this paper, will provide upper bounds on the performance of the system.

Given a node $v \in V$, we use $N_{in}(v)$ to denote the set of links that terminate at node v. In other words,

$$N_{in}(v) = \{e \in E : r(e) = v\}.$$

Similarly, for a given node $v \in V$, we use $N_{out}(v)$ to denote the set of edges that emanate from v, i.e,

$$N_{out}(v) = \{e \in E : t(e) = v\}.$$

We use $N(v)$ to denote $N_{in}(v) \cup N_{out}(v)$. A list of symbols used in this paper appears in the appendix in Table 3.

The network model that we consider here occurs with networks with multiple orthogonal channels available for transmission with nodes having multiple channel elements for receiving simultaneously from multiple nodes. As stated earlier, we assume that in any given time slot a given node transmits to at most one other node in the network. An example of a system modeled by this mathematical abstraction is a frequency division system where potentially colliding transmissions are assigned non-overlapping frequency bands. We assume that there is a higher layer that does frequency planning to ensure that there is no interference. This model can also be used as an approximation for combined TDMA/CDMA based multi-hop networks [10] where nearby transmissions are assigned orthogonal codes to communicate between each other.

In the rest of this paper, we assume that all link speeds, flows and rates are rational numbers. Before we consider the maximum data rate that can be transmitted from a given source node to destination node, we consider the simpler problem of the achievability of a given set of link flows.

4 Achieving Link Flows

We now consider the problem of determining if a set of link flows are achievable. Instead of attempting to solve this problem directly as in [8], we outline simple necessary and sufficient conditions for the achievability of link flows. The objective is to derive a set of simple conditions that can be used to formulate and solve the end-to-end flow requirement problems. Solving the link scheduling problem also serves to illustrate the effect of scheduling on the routing problem. Some of the results in this section are also implicit in [8].

Assume that we are given a m-vector $\mathbf{f}$ where $f(e)$ is the desired flow on link $e \in E$. The objective is to determine necessary and sufficient conditions for this link flow vector to be achievable. Note that the flow is specified as a link flow and not as an end-to-end flow. This connection between end-to-end flows and individual link flows will be made in the next section. In order to achieve this link flow we first define a $0-1$ scheduling variable

$$y_e^t = \begin{cases} 1 & \text{If link } e \text{ is active in time slot } t \\ 0 & \text{Otherwise} \end{cases}$$

Note that y_e^t is set to one if there is a transmission on link e in time period t.

Since no node can be transmitting to more than one node in a given time slot, we have the inequality,

$$\sum_{e \in N_{out}(v)} y_e^t \leq 1, \quad \forall v \in V, \quad \forall t \leq T \tag{1}$$

Similarly, since no node can receive from more than $\Omega(v)$ neighbors in a given time slot, we have,

$$\frac{1}{\Omega(v)} \sum_{e \in N_{in}(v)} y_e^t \leq 1, \quad \forall v \in V, \quad \forall t \leq T \tag{2}$$

Note that the fraction of time a link e is active is given by

$$\frac{\sum_{t \leq T} y_e^t}{T}$$

Therefore, the mean flow on link e is given by

$$f(e) = \frac{c(e) \sum_{t \leq T} y_e^t}{T}$$

4.1 Necessary Conditions

We now derive the necessary conditions that a link flow vector has to satisfy, in order to be schedulable.

4.1.1 Half Duplex Systems

For a Half Duplex system, we need to introduce two additional variables. Let $I_x^t(v)$ and $I_r^t(v)$ be indicator functions that take the value of 1 if v is transmitting and receiving, respectively, at time slot t, and 0 otherwise. Therefore, when a node v is Half Duplex, we have the following communication constraints.

$$\begin{aligned} I_x^t(v) + I_r^t(v) &\leq 1, & \forall v \in V, \ \forall t \leq T \\ \sum_{e \in N_{out}(v)} y_e^t &\leq I_x^t(v), & \forall v \in V, \ \forall t \leq T \\ \sum_{e \in N_{in}(v)} y_e^t &\leq \Omega(v) I_r^t(v), & \forall v \in V, \ \forall t \leq T \end{aligned} \tag{3}$$

We can combine the communication constraints in Equation (3) as,

$$\sum_{e \in N_{out}(v)} y_e^t + \frac{1}{\Omega(v)} \sum_{e \in N_{in}(v)} y_e^t \leq 1, \quad \forall v \in V, \ \forall t \leq T$$

Summing this equation over all T, interchanging the order of summation, and dividing by T, we get the necessary conditions for the achievability of a link flow vector, which we state formally below.

Lemma 1 *A given link flow vector* $\mathbf{f}$ *is schedulable only if*

$$\sum_{e \in N_{out}(v)} \frac{f(e)}{c(e)} + \frac{1}{\Omega(v)} \sum_{e \in N_{in}(v)} \frac{f(e)}{c(e)} \leq 1, \quad \forall v \in V.$$

For the Half Duplex system, the above condition is only a necessary condition. This is illustrated by the following example, where $\Omega(v) = 1, \forall v \in V$. The value of $c(e)$ for each link is 1 unit and the value of $f(e)$ for each link is 0.5 units as shown in the Figure 2. Note that this flow vector

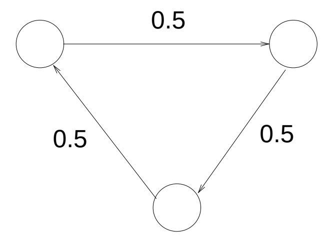

Figure 2: Three Node Example

satisfies Lemma 1 at all the nodes in the network. However, this flow vector is not achievable. This is so because in any given time slot at most one of the three links can be active. This gives a throughput of at most $\frac{1}{3}$ on each link in the network. This gap between the necessary condition for achievability and actual achievability of the flow vector results from averaging the scheduling constraints over time which relaxes the $0-1$ constraints on the y_e^t variables.

4.1.2 Full Duplex Systems

In the case of a Full Duplex system, a given node v can transmit to at most one node in any given time slot. Therefore, summing Equation (1) over all t,

$$\sum_{t \leq T} \sum_{e \in N_{out}(v)} y_e^t \leq T, \quad \forall v \in V$$

Interchanging the order of summation, and dividing by T, we get

$$\sum_{e \in N_{out}(v)} \frac{f(e)}{c(e)} = \sum_{e \in N_{out}(v)} \frac{\sum_{t \leq T} y_e^t}{T} \leq 1, \quad \forall v \in V.$$

Similarly, since each node can receive from at most $\Omega(v)$ channels in any time slot,

$$\sum_{e \in N_{in}(v)} \frac{f(e)}{c(e)} = \sum_{e \in N_{in}(v)} \frac{\sum_{t \leq T} y_e^t}{T} \leq \Omega(v), \quad \forall v \in V.$$

This implies the following result.

Lemma 2 *A given link flow vector* $\mathbf{f}$ *is schedulable only if*

$$\sum_{e \in N_{out}(v)} \frac{f(e)}{c(e)} \leq 1, \quad \forall v \in V$$

and,

$$\frac{1}{\Omega(v)} \sum_{e \in N_{in}(v)} \frac{f(e)}{c(e)} \leq 1, \quad \forall v \in V$$

Even though we average the $0-1$ constraints over time, these conditions also turn out to be sufficient conditions, as we prove later in this section.

4.2 Sufficient Conditions

We derive sufficient conditions for the achievability of a link flow vector based on graph edge coloring. In order to do so, we need a few more definitions. In the rest of this paper, we use the term *link* exclusively to denote an edge of the network graph $G = (V, E)$.

Definition 1 *A multi-graph is a graph where there may be multiple edges between the same pair of nodes. An alternate representation of a multi-graph is to have an integral weight $w(e)$ on link e in the network G, where $w(e)$ represents the number of edges between $t(e)$ and $r(e)$ in the multi-graph.*

We now define a multi-graph on the network G to help us obtain an achievable schedule for a given link flow vector $\mathbf{f}$. With τ seconds/slot, we can send at most $\tau \cdot c(e)$ bits/slot on link e. Hence, in order to achieve a link flow of $f(e)$ bits/second, link e needs to be scheduled for $f(e)/\tau \cdot c(e)$ slots/second. Therefore, choose a slot time τ such that

$$w(e) = \frac{f(e)}{c(e)} \frac{1}{\tau} \text{ slots/second} \tag{4}$$

is integral. Such a τ exists since all values of $f(e)$ and $c(e)$ were assumed to be rational. There exist many such τ that satisfy the above equation. We choose the largest such τ. The $w(e)$ is the slot rate required to satisfy a flow of $f(e)$ bits/second on a link e with capacity $c(e)$.

Definition 2 *Given the network $G = (V, E)$, the flow vector $\mathbf{f}$ and link speed vector $\mathbf{c}$, let the link weights $w(e)$ be defined as $w(e) = \frac{f(e)}{\tau \cdot c(e)}, \forall e \in E(G)$. The* scheduling multi-graph *$G_S(\mathbf{f}, \tau)$, corresponding to G, has the same node set V, with a link $e \in E(G)$ represented by $w(e)$ edges in $G_S(\mathbf{f}, \tau)$ between the same endpoints.*

Definition 3 *The* maximum degree *of the scheduling multi-graph is denoted by*

$$\Delta = \max_{v \in V} \sum_{e \in N(v)} w(e)$$

The maximum out-degree *of a node is denoted by*

$$\Delta^+ = \max_{v \in V} \sum_{e \in N_{out}(v)} w(e)$$

The maximum weighted in-degree *of a node is denoted by*

$$\Delta_l^- = \max_{v \in V} \frac{1}{\Omega(v)} \sum_{e \in N_{in}(v)} w(e)$$

We use the sub-script l in the notation for the max weighted in-degree in order to distinguish it from the maximum in-degree Δ^- which does not have the term $1/\Omega(v)$. We are now ready to derive the sufficiency conditions for the two models.

4.2.1 Half Duplex Systems

We define a suitable coloring on the scheduling multi-graph that will translate to a valid schedule in a Half Duplex network graph, G.

Definition 4 *A proper ψ-coloring of the scheduling multi-graph such that the following conditions apply for all $v \in V$:*

1. *No two edges in the outgoing edge set $N_{out}(v)$ have the same color, and*
2. *Any color i is present on at most $\Omega(v)$ incoming edges in $N_{in}(v)$.*
3. *No color that appears on an edge in $N_{in}(v)$ appears on any edge in $N_{out}(v)$, i.e. the color list of the outgoing edges has nothing in common with the color list of the incoming edges.*

The ψ-chromatic index, L_ψ, of the scheduling multi-graph G_S is defined as the minimum number of colors needed to ψ-color the edges of the multi-graph.

In the case where $\Omega(v) = 1$ for all $v \in V$, the ψ-chromatic index is just the regular chromatic index and we represent this value by L.

The following Theorem is due to Shannon [20].

Theorem 3 *Let $G_S(V, E)$ represent a multi-graph with maximum degree Δ. The chromatic index L of this multi-graph satisfies the following relation:*

$$\Delta \leq L \leq \frac{3}{2}\Delta.$$

The lower bound on L is obvious. Shannon [20] gives a constructive proof of the upper bound.

Lemma 4 *Any proper ψ-coloring of the scheduling multi-graph $G_S(\mathbf{f}, \tau)$ corresponds to a valid schedule for the network graph G representing a* Half Duplex *system. A flow $\mathbf{f}$ is achievable iff the slot-rate is at least L_ψ slots per second, i.e., iff $L_\psi \tau \leq 1$.*

Proof:
Let us consider a scheduling frame with slot of length τ seconds each. Each edge e in the scheduling multi-graph $G_S(\mathbf{f}, \tau)$ *corresponds* to one time slot that has to be allocated to the link that represents e, in the network graph G. If a set of edges, W, in $G_S(\mathbf{f}, \tau)$ have color i, then correspondingly, in time slot i, make all the links in G, represented by the edges in W, active. Repeat this process for colors $i = 1, 2, \ldots L_\psi$. No two edges in $N_{out}(v)$ have the same color implies that only node v can transmit on only one link at any time. Similarly, since at most $\Omega(v)$ edges in $N_{in}(v)$ have the same color implies that node v can receive on at most $\Omega(v)$ links simultaneously. The coloring does not allow for the same color to appear on an incoming edge and outgoing edge. This is the same as saying that a node cannot receive and transmit in the same time slot. Thus, a proper ψ-coloring with R colors translates to a valid Half Duplex schedule in R slots.

With R colors, we have colored all edges of the multi-graph, which implies that in a schedule of length R slots, each link e in the network graph G is active for $w(e)$ time slots, thereby sending $f(e)$ bits. The length of the schedule has to be at least L_ψ slots, since any proper ψ-coloring needs at least L_ψ colors.

Therefore, to send a flow of $f(e)$ bits per second, we need a slot-rate of $R \geq L_\psi$ slots per second. Since the slot length is τ seconds per slot, the available slot rate is $1/\tau$ slots per second. Thus, $L_\psi \leq 1/\tau$ for the flow $\mathbf{f}$ to be achievable.

Since a proper edge coloring of the scheduling multi-graph cannot exist using less than L_ψ colors, the corresponding mapping ensures that a valid schedule that satisfies the demand with a slot rate of less than L_ψ slots per second cannot be found. Therefore, if $L_\psi > 1/\tau$, then $\mathbf{f}$ is not achievable. □

Corollary 5 *Let L_ψ represent the ψ-chromatic index of the scheduling multi-graph $G_S(\mathbf{f}, \tau)$. Given that a slot rate of L_ψ slots/second is needed to achieve the flow $\mathbf{f}$, then, with the available slot rate of $1/\tau$ slots per second,*

$$\frac{f(e)}{L_\psi \tau}, \quad \forall e \in E$$

is an achievable flow in a Half Duplex network G.

Proof:
If we have L_ψ slots/second, then we are guaranteed to achieve a flow of $\mathbf{f}$ bits/second. However, if there are only $1/\tau$ slots/second, then, using these slots, we can only send

$$\frac{\mathbf{f}}{L_\psi} \frac{1}{\tau} \text{bits/second}$$

□

Note that once the ψ-chromatic index L_ψ and the corresponding coloring is known for $G_S(\mathbf{f}, \tau)$, we do not have to recompute the chromatic index to find the schedule for a flow of $\mathbf{f}' = \mathbf{f}/L\tau$. By changing the length of the time slot to $\tau' = 1/L_\psi$, it is easy to see that $G_S(\mathbf{f}', \tau') = G_S(\mathbf{f}, \tau)$. Thus, $L'_\psi = L_\psi$ and $L'_\psi \tau' = 1$. Therefore, the coloring and hence, the schedules are still the same as before, except that the time slot length has changed.

Theorem 6 *Let L_ψ be the ψ-chromatic index of a scheduling multi-graph, $G_S(\mathbf{f}, \tau)$, representing a Half Duplex network, G. Let L be the chromatic index of $G_S(\mathbf{f}, \tau)$. We have,*

$$L_\psi \leq L \leq \frac{3}{2}\Delta$$

Proof:
Any proper edge-coloring of the scheduling multi-graph is also a proper ψ-coloring of the multi-graph. Therefore, an edge-coloring using L colors is a proper ψ-coloring. Thus, $L_\psi \leq L$. From Shannon's Theorem [20], we know that $L \leq \frac{3}{2}\Delta$. This concludes the proof. □

Determining the chromatic index L of a graph is NP-hard. However,

- There are fast algorithms to determine a 1.1– approximation to the chromatic index [21].
- A simple greedy algorithm [1] exists for a 2-approximate solution to the problem in time $O(\frac{m}{\tau})$. This will be covered in detail in Section 4.3.
- An algorithm to construct the $\frac{3}{2}\Delta$ solution can be computed in time $O(\frac{nm}{\tau})$. A constructive proof was provided by Shannon [20].

It is easy to implement the greedy edge coloring algorithm in a distributed manner. Though the algorithm guarantees only a 2-approximation in the worst case, it usually does much better in practice.

We now state sufficient conditions for the achievability of a link flow vector $\mathbf{f}$ in a Half Duplex network.

Theorem 7 *For a Half Duplex network represented by the network graph G, an m-link flow vector is achievable if*

$$\sum_{e \in N(v)} \frac{f(e)}{c(e)} \leq \frac{2}{3}, \quad \forall v \in V$$

Proof:
From the conditions in the statement of the theorem, and the definition of $w(e)$, note that

$$\sum_{e \in N(v)} w(e) \leq \frac{2}{3}\frac{1}{\tau}$$

Therefore, $\Delta \leq \frac{2}{3}\frac{1}{\tau}$ and, using Theorem 6,

$$L_\psi \leq L \leq \frac{3}{2}\Delta \leq \frac{1}{\tau}$$

This implies that $L_\psi \tau \leq 1$ and, therefore, by Lemma 4, $f(e)$ is achievable. □

Therefore when a link flow vector satisfies the sufficiency conditions, it is schedulable. Note that there is a gap between the necessary and sufficiency conditions.

Clearly, if there exits a $v \in V$ such that

$$\sum_{e \in N_{out}(v)} \frac{f(e)}{c(e)} + \frac{1}{\Omega(v)} \sum_{e \in N_{in}(v)} \frac{f(e)}{c(e)} \geq 1$$

then it not schedulable. If a link flow vector satisfies

$$\sum_{e \in N(v)} \frac{f(e)}{c(e)} \leq \frac{2}{3}, \qquad \forall v \in V$$

then it is schedulable. If $\mathbf{f}$ does not satisfy both necessary and the sufficient conditions, then it is not clear whether it is schedulable or not. We attempt to close this gap (in practice) using the following strategy: As long as the link flow vector $\mathbf{f}$ satisfies the necessary conditions, we construct the scheduling multi-graph and determine its chromatic index. Let τ denote the length of the time-slot and L'_ψ denote the approximation to the ψ-chromatic index of the resulting scheduling multi-graph. If $L'_\psi \tau \leq 1$ then the given link flow vector $\mathbf{f}$ is achievable. If $\mathbf{f}$ satisfies the sufficiency conditions, then clearly it satisfies $L'_\psi \tau \leq 1$ and is therefore achievable. Using this strategy will result in determining the achievability of link flow vectors that fall in the gap between the necessary and sufficiency conditions. We state the algorithm formally below:

- If the vector $\mathbf{f}$ does not satisfy the necessary conditions then output f is not achievable.
- Determine τ and $w(e)$ as shown in equation (4).
- Construct the scheduling graph and determine an approximation L'_ψ to its ψ-chromatic index.
- If $L'_\psi \tau \leq 1$, then f is achievable.

Experimental testing indicates that the above algorithm performs extremely well in practice. The effectiveness of this approach will be illustrated in the case of determining the maximum achievable rate between a given pair of nodes. Also note that the link rates are not just achievable in the long run but are achieved in time at most 1 second.

4.2.2 Full Duplex Systems

We obtain the graph coloring characterization for Full Duplex systems, in the same way as for Half Duplex systems.

Definition 5 *A proper ϕ-coloring of the scheduling multi-graph colors the edges of the multi-graph such that the following conditions apply for all nodes $v \in V$:*

1. *No two edges in the outgoing edge set $N_{out}(v)$ have the same color, and*
2. *Any color i is present on at most $\Omega(v)$ incoming edges in $N_{in}(v)$.*

The ϕ-chromatic index, L_ϕ, of the scheduling multi-graph G_S is defined as the minimum number of colors needed in any proper ϕ-coloring of the scheduling multi-graph.

Note that in the definition, there is no requirement tying up the colors of the incoming and outgoing links at any node.

Note that ψ-coloring is more restrictive than ϕ-coloring. The proofs of the following theorem and corollary follow along the lines of the proofs for Half Duplex systems.

Lemma 8 *Any proper ϕ-coloring of the scheduling multi-graph $G_S(\mathbf{f}, \tau)$ corresponds to a valid schedule for the network graph G representing a* Full Duplex *system. A flow $\mathbf{f}$ is achievable iff $L_\phi \tau \leq 1$.*

Corollary 9 *Let L_ϕ represent the ϕ-chromatic index of the scheduling multi-graph $G_S(\mathbf{f}, \tau)$. Given that a slot rate of L_ϕ slots/second is needed to achieve the flow $\mathbf{f}$, then, with the available slot rate of $1/\tau$ slots per second,*

$$\frac{f(e)}{L_\phi \tau}, \quad \forall e \in E$$

is an achievable flow in a Full Duplex network G.

The ϕ-chromatic index of a scheduling multi-graph can be determined using the following lemma. It uses the results of Hakimi and Kariv [19].

Lemma 10 *The ϕ-chromatic index, L_ϕ, of a scheduling multi-graph $G_S(\mathbf{f}, \tau)$ is given by*

$$L_\phi = \max(\Delta^+, \Delta_l^-) \tag{5}$$

Proof:
We derive a bipartite graph $H(V_T, V_R)$ from the scheduling multi-graph G_S. Let a node $v \in V(G_S)$ be represented by two nodes, $v_t \in V_T$ and $v_r \in V_R$ in H. For each edge e going from node u to node v in G_S, define an edge from the corresponding vertices u_t to v_r in H.

From the above definition, we can see that there are no edges between vertices in V_T, or between vertices in V_R. Hence, H is bipartite with partite sets V_T and V_R. Now, for each vertex v in H, define

$$\kappa(v) = \begin{cases} 1 & \text{If } v \in V_T \\ \Omega(v) & \text{If } v \in V_R \end{cases}$$

We define a proper κ-coloring of H to be a coloring for which, at each vertex $v \in V(H)$, the number of edges having a color i is at most $\kappa(v)$.

It is easy to see that a proper ϕ-coloring of G_S implies a proper κ-coloring of H, and vice-versa. Therefore, the minimum number of colors L_κ needed for a proper κ-coloring of H is equal to L_ϕ. If $d(v)$ denotes the vertex degree in H, then it is clear that $\max_{v \in V_T} \frac{d(v)}{\kappa(v)} = \Delta^+$ and $\max_{v \in V_R} \frac{d(v)}{\kappa(v)} = \Delta_l^-$.

From the results in Hakimi and Kariv [19], the minimum number of colors L_κ needed for a proper κ-coloring of a bipartite graph H is given by

$$\begin{aligned} L_\kappa &= \max_{v \in (V_T \cup V_R)} \lceil \frac{d(v)}{\kappa(v)} \rceil \\ &= \max(\Delta^+, \Delta_l^+) \end{aligned}$$

Since $L_\kappa = L_\phi$, we have our result. □

Using this, we can show that the sufficient condition for the achievability of a link flow vector in a Full Duplex network is the same as the necessary conditions in Lemma 2.

Theorem 11 *A given link flow vector $\mathbf{f}$ is schedulable (achievable) if and only if*

$$\sum_{e \in N_{out}(v)} \frac{f(e)}{c(e)} \leq 1 \qquad \forall v \in V$$

and,

$$\frac{1}{\Omega(v)} \sum_{e \in N_{in}(v)} \frac{f(e)}{c(e)} \leq 1 \qquad \forall v \in V$$

Proof:
We only need to prove the sufficiency. From the conditions in the statement of the theorem, and the definition of $w(e)$, note that

$$\sum_{e \in N_{out}(v)} w(e) \leq \frac{1}{\tau}$$

$$\frac{1}{\Omega(v)} \sum_{e \in N_{in}(v)} w(e) \leq \frac{1}{\tau}$$

Therefore, $\Delta^+ \leq \frac{1}{\tau}$ and $\Delta_l^- \leq \frac{1}{\tau}$. Using Equation (5), we have

$$L_\phi = \max(\Delta^+, \Delta_l^-) \leq \frac{1}{\tau}$$

This implies that $L_\phi \tau \leq 1$ and by Lemma 8, $f(e)$ is achievable. □

4.3 Greedy Coloring Algorithm

While there exists an optimal polynomial time algorithm for edge-coloring the scheduling multi-graph for a Full Duplex system [19], due to lack of space, we describe a unified greedy algorithm for both the Half Duplex and the Full Duplex systems described in this section. We can use greedy edge coloring to obtain a 2-approximate solution to the optimal. Greedy coloring can be implemented in a distributed manner in a multi-hop wireless network, while finding the optimum coloring requires centralized coordinator to compute the schedules. In addition, greedy coloring can be performed in $O(m)$ time, while the optimum coloring algorithm takes much longer. We used the greedy coloring algorithm for obtaining the schedules in our simulations. The greedy coloring algorithm is described in Figure 3.

- INPUT: The scheduling multi-graph $G_S(\mathbf{f}, \tau)$ representing either a Full Duplex or a Half Duplex network.
- OUTPUT: A valid edge coloring satisfying the communication constraints (Full Duplex or Half Duplex) imposed by the system represented by G_S.
- PROCEDURE:
  1. Pick a link e in the scheduling multi-graph. Let its end-points be u and v.
  2. Pick the smallest color i that can be assigned to e such that the communication constraints are not violated at u or v. Assign color i to e.
  3. Repeat steps (1) and (2) until all edges of G_S are colored.

Figure 3: Greedy Coloring Algorithm

We can easily prove the greedy solution is at most twice the optimal schedule length.

Lemma 12 *The greedy coloring algorithm described in Figure 3 is 2-optimal.*

Proof:
Let $k = \max(\Delta^+, \Delta_l^-)$ for the scheduling multi-graph of the Full Duplex system. We know that $L_\phi = k$. Let the greedy coloring algorithm consider an edge e in the scheduling multi-graph, with nodes u and v as its end-points. In the worst-case, we can have colors $1, ..., k-1$ fully used at node u, and colors $k, ..., 2k-1$ fully used at node v, thereby forcing edge e to have color $2k$. Thus, any edge of the scheduling multi-graph can be colored using at most $2k$ colors.

For the scheduling multi-graph corresponding to a Half Duplex system, let

$$k = \max_{v \in V}\Big(\sum_{e \in E_{out}(v)} w(e) + \lceil \frac{1}{\Omega(v)} \sum_{e \in E_{in}(v)} w(e) \rceil \Big)$$

It is easy to see that $L_\psi \geq k$. Using the same argument as above, the greedy algorithm can color all edges of the scheduling multi-graph using at most $2k$ colors. Thus, $k \leq L_\psi \leq 2k$. This concludes the proof of the lemma. □

Note that [19] provides an algorithm for optimally coloring the scheduling multi-graph of a Full Duplex network. Thus, for Full Duplex systems, there exist algorithms to find the optimum. However, for Half Duplex systems, the greedy coloring given above is the algorithm with the best known performance bound (of 50%), thus far. For the Half Duplex network with $\Omega(v) = 1, \forall v \in V$, Theorems 6 and 7 provide an algorithm that achieves 67% of the optimal solution.

5 Achievable Rates for Multiple Source Destination Pairs

Now consider the problem of characterizing the achievable rates in the case of multiple source destination pairs.[1] 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 has $K < n(n-1)$ components. Each source-destination pair between which there is a request will be termed a commodity. We use k to index the commodities. Let $s(k)$ represent the source node for commodity k and $d(k)$ the destination node for commodity k. Let $r(k)$ represent the flow that has to be routed from $s(k)$ to $d(k)$. The problem that we have to solve is the following:

- INPUT: A directed graph $G = (V, E)$ with a link speed $c(e)$ for $e \in E$ and K node pairs $(s(k), d(k))$ and associated with each node pair k is a desired rate $r(k)$.
- OUTPUT: Either declare a set of routes and associated schedule that achieves the given rates, or declare the problem as not achievable.

The strategy that we use to determine the set of feasible routes and the associated schedules for the above problem is as follows:

- We use the results developed in the last section to formulate necessary and sufficient conditions for a rate to be achievable.
- We get an upper bound on the achievable rates by solving a linear optimization problem over the *necessary conditions*.
- We use the scheduling multi-graph to get an achievable solution which is a lower bound on feasible rate vector.

[1] We also present a detailed solution for the single source-destination pair (max-flow) problem in [17]. It is a simpler version of the problem considered here.

- We show that this lower bound gets to within 50% of the optimal solution in the worst case and, in practice typically gets to between 80-90% of the optimal solution.

The linear optimization problem that is solved in the second step is the key to determining bounds on the achievable rate vector. We first give a straightforward formulation for this problem with flow variables. We also give an alternate path-arc formulation that is amenable to the development of primal-dual algorithms for the solution of fully polynomial time approximation schemes (FPTAS). The reason for preferring the FPTAS to solving the linear program directly are the following:

- The FPTAS are very simple to implement.
- It is possible to trade-off the accuracy needed with the speed of solution. From our experiments, we observed that solving the linear programming problem approximately is enough to solve the routing-scheduling problem almost optimally.
- There is no need for a linear programming solver to solve the problem. (This is especially important if we have to implement the algorithm at the individual nodes).

We define the following quantities in order to keep notation simple. Let $f_k(e)$ be the flow on link e associated with commodity k. Let

$$J_{out}(v) = \sum_{e \in N_{out}(v)} \frac{\sum_k f_k(e)}{c(e)}$$

and

$$J_{in}(v) = \frac{1}{\Omega(v)} \sum_{e \in N_{in}(v)} \frac{\sum_k f_k(e)}{c(e)}.$$

5.1 Half Duplex Case

As in the case of the single source-destination flow problem, it is easy to show the following result.

Theorem 13 *Given a graph $G = (V, E)$, with link speed $c(e)$ associated with link $e \in E$, K source destination pairs $(s(k), d(k))$ for $k = 1, 2, \ldots K$ with a desired flow rate $r(k)$ between $s(k)$ and $d(k)$. The rate vector* $\mathbf{r}$ *is achievable only if there exists a flow $f_k(e)$ such that*

$$\begin{aligned}
\sum_{e:t(e)=s(k)} f_k(e) &= r(k), \quad \forall k \\
\sum_{e \in N_{in}(v)} f_k(e) &= \sum_{e \in N_{out}(v)} f_k(e), \quad \forall v \neq s, d, \quad \forall k \\
J_{in}(v) + J_{out}(v) &\leq 1, \quad \forall v \in V \\
f_k(e) &\geq 0 \quad \forall e \quad \forall k.
\end{aligned}$$

Proof:
The first constraint ensures that a data rate of $r(k)$ units is sent out of the source node for commodity k. The second set of equalities ensure flow balance at the nodes in the network for each commodity. The last set of inequalities are the necessary conditions for a link flow vector to be achievable. □

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

$$\sum_{P \in \mathcal{P}_k} x(P) = r(k), \quad \forall k.$$

The total amount of flow on link e, represented by $f(e)$ is given by

$$f(e) = \sum_k \sum_{P \in \mathcal{P}_k : e \in P} x(P)$$

We define

$$Q_{out}(v, x) = \sum_{e \in N_{out}(v)} \frac{\sum_k \sum_{P \in \mathcal{P}_k : P \ni e} x(P)}{c(e)}$$

$$Q_{in}(v, x) = \sum_{e \in N_{in}(v)} \frac{\sum_k \sum_{P \in \mathcal{P}_k : P \ni e} x(P)}{\Omega(v) c(e)}.$$

Then the necessary conditions for a rate vector $\mathbf{r}$ to be achievable is the existence of path flows $x(P)$ such that

$$\begin{aligned}
\sum_{P \in \mathcal{P}_k} x(P) &= r(k), \quad \forall k \\
Q_{out}(v, x) + Q_{in}(v, x) &\leq 1, \quad \forall v \in V. \\
x(P) &\geq 0, \quad \forall P \in \mathcal{P}_k, \ \forall k.
\end{aligned}$$

Given a rate vector $\mathbf{r}$, the strategy then is to solve for the x variables that satisfies the necessary conditions. If such a vector does not exist, then the given rate vector is not achievable. If it satisfies the necessary condition, then we determine the length of the time slot τ, form the schedule multi-graph and determine its ψ-chromatic index L_ψ. Using the same techniques used above, it is not difficult to show that

$$\frac{r(k)}{L_\psi \tau}$$

is achievable. As in the previous cases, we show that in practice the performance of the algorithm is extremely good in practice.

5.2 Solving the Linear Programming Problem

In order to solve the linear programming problem, we first write the achievability problem as a concurrent flow problem and then use a primal-dual algorithm to solve this problem.

$$\max \lambda$$

$$\begin{aligned}
Q_{out}(v, x) + Q_{in}(v, x) &\leq 1, \quad \forall v \in V. \\
\sum_{P \in \mathcal{P}_k} x(P) &= \lambda\, r(k), \quad \forall k = 1, 2, ..., K \\
x(P) &\geq 0, \quad \forall P \in \mathcal{P}_k, \quad \forall k
\end{aligned}$$

In the concurrent flow problem, the objective is to determine the maximum scaling factor λ^* that if all the desired traffic rates are scaled up by this factor, then it will still fit in the network. Therefore, if the objective function λ^* is less than than one then the vector is not achievable. If $\lambda^* \geq 1$, then we have to schedule the flow to determine if the flow is schedulable.

The largest flow vector that still satisfies the necessary constraints is $\lambda^* \mathbf{r}$ which is given by the optimal link flow vector $\mathbf{x}^*$, found from the solution to the LP above. If we have a Full Duplex system, then Theorem 11 tells us that the flow $\lambda^* \mathbf{r}$ is schedulable, and a schedule can be found using the algorithms described in Section 4.

For a Half Duplex system, we apply Corollary 5 to the flow $\lambda^* \mathbf{r}$ and get a schedule for a flow vector $\frac{\lambda^* \mathbf{r}}{L_\psi^* \tau}$, where L_ψ^* is the ψ-chromatic index of the scheduling multi-graph $G_S(\mathbf{x}^*, \tau)$. For this schedulable flow to be at least $\mathbf{r}$, we need $\lambda^* \geq L_\psi^* \tau$.

We summarize this in the following theorem.

Theorem 14 *1. If $\lambda^* < 1$, then $\mathbf{r}$ is not schedulable.*

2. For a Full Duplex system, if $\lambda^* \geq 1$, then the flow $\lambda^* \mathbf{r}$ is schedulable.

3. For a Half Duplex system, if $\lambda^* \geq L_\psi^* \tau$, then there exists a schedule for the flow $\frac{\lambda^* \mathbf{r}}{L_\psi^* \tau}$, and hence for $\mathbf{r}$. If $1 \leq \lambda^* \leq \frac{\lambda^* \mathbf{r}}{L_\psi^* \tau}$, then it is not known whether there exists a valid schedule for $\mathbf{r}$.

The dual to this problem assigns a weight $\eta(v)$ to each node v in the network, and a variable $z(k)$ for each commodity (source-destination pair) $k = 1, 2, ..., K$.

$$\min \sum_{v \in V} \eta(v)$$

$$\begin{aligned}
\sum_{e \in P} \frac{1}{c(e)} \left[\eta(t(e)) + \frac{\eta(r(e))}{\Omega(r(e))} \right] &\geq z(k), \quad \forall P \in \mathcal{P}_k, \ \forall k \\
\sum_{k=1}^{K} r(k) z(k) &\geq 1 \\
\eta(v) &\geq 0, \quad \forall e \in E
\end{aligned}$$

The primal dual algorithm to solve the concurrent flow problem starts by assigning a precomputed weight of δ to all nodes v. The algorithm proceeds in phases.[2] In each phase, for each commodity k, we route $r(k)$ units of flow from $s(k)$ to $d(k)$. A phase ends when commodity K is routed. The $r(k)$ units of flow from $s(k)$ to $d(k)$ for commodity k is sent via multiple iterations. In each iteration, the shortest path P^* from $s(k)$ to $d(k)$ is determined. Let $f(P^*)$ represent the maximum flow that can be sent on this path. We can send a flow of at most $f(P^*)$ units this iteration. Since $r(k)$ units of flow have to be sent for commodity k in each phase, the actual amount of flow sent

[2]The algorithm is explained in greater detail in [1, 17].

```
DETERMINE_FEASIBILITY

η(v) = δ   ∀v ∈ V and c = 0
While  Σ_{v∈V} η(v) < 1
    For  k = 1, 2, ..., K
        r = r(k)
        While  r > 0
            Set l(e) = 1/c(e) [η(t(e)) + η(r(e))/Ω(r(e))]
            Compute shortest path length from s(k)
                to t(k)
            Let P* = arg min_{P∈P_k} η(P) be
                the optimal path.
            Let u = min_{v∈P*} f(v, P*).
            δ = min{r, u};     r ← r − δ
            f(e) ← f(e) + δ, ∀e ∈ P*
            η(v) ← η(v) (1 + θ(v, P*)δ),   ∀v ∈ P*
        end While
    end For
    c ← c + 1
end While
Compute ρ = max_{v∈V} Σ_{e∈N(v)} f(e)/c(e)
Output λ* = c/ρ
```

Figure 4: Primal Dual Algorithm: Half Duplex Case

is the lesser of $f(P^*)$ and the remaining amount of flow to make up $r(k)$ in this phase. Once the flow is sent, the weights of the nodes that carry the flow is increased. The algorithm is shown in Figure 4. 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. A detailed analysis of this algorithm can be found in [1].

By organizing the computation by source, one can send flows to multiple destinations at the same time as in Karakostas [18] and the running time of the algorithm has only a logarithmic dependency on the number of source destination pair. The proof of the following theorem is similar to that in [1].

Theorem 15 *The DETERMINE_FEASIBILITY algorithm computes a $(1-\epsilon)^{-3}$ optimal solution to the rate achievability problem in time $\mathcal{O}(\epsilon^{-2} m^2)$.*

5.3 Full Duplex Case

In the Full Duplex case, the following theorem that parallels Theorem 13 for the Half Duplex case forms the basis for the multiple source destination feasibility problem.

Theorem 16 *Given a graph $G = (V, E)$, with link speed $c(e)$ associated with link $e \in E$, K source destination pairs $(s(k), d(k))$ for $k = 1, 2, \ldots K$ with a desired flow rate $r(k)$*

between $s(k)$ and $d(k)$. The rate vector $\mathbf{r}$ is achievable only if there exists $f_k(e)$ such that

$$\begin{aligned}
\sum_{e:t(e)=s(k)} f_k(e) &= r(k), \quad \forall k \\
\sum_{e\in N_{in}(v)} f_k(e) &= \sum_{e\in N_{out}(v)} f_k(e), \quad \forall v \neq s, d, \quad \forall k \\
J_{out}(v) &\leq 1 \quad \forall v \in V \\
J_{in}(v) &\leq 1 \quad \forall v \in V \\
f_k(e) &\geq 0 \quad \forall e \in E, \forall k.
\end{aligned}$$

The path-arc formulation and the resulting algorithm follow the same pattern as in the Half Duplex case and is omitted.

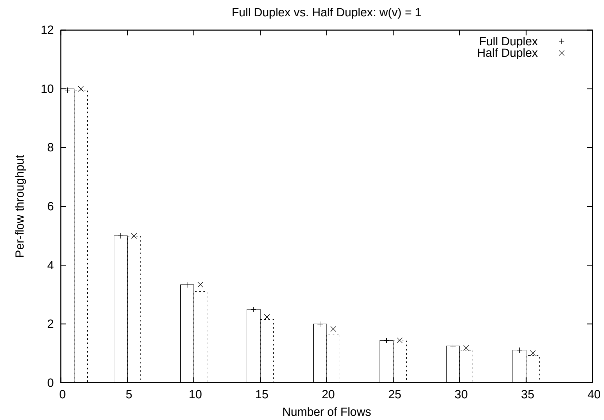

Figure 5: Grid - Full Duplex '+' vs. Half Duplex 'x': $\Omega(v) = 1$

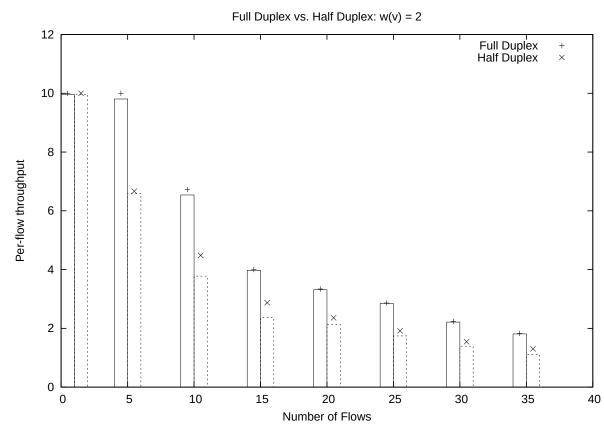

Figure 6: Grid - Full Duplex '+' vs. Half Duplex 'x': $\Omega(v) = 2$

6 Simulation Results

In this section, we present some results on the performance of the routing-scheduling algorithms for the Full Duplex and Half Duplex systems. The routing problem is solved using the primal-dual scheme with $\epsilon = 0.05 - 0.1$. The algorithm executes within a couple of seconds for all the problems considered. To find the schedules, we used the 2-approximate greedy algorithm to solve the coloring problem.

In all the examples discussed below, each source node has a demand of 1. The results of in the simulation give the maximum scaling factor λ^*, by which a flow can be scaled so that it still meets the constraints of the LP. It is important to note that all flows are scaled uniformly by this factor. Thus, even though a single flow might be able to send more along its paths, we are interested only in the minimum common flow that every source can send.

In each example, we provide the upper bound, λ^*, that is given by solving the linear programming problem with the necessary conditions as well as the achievable solution given by the coloring algorithm.

In all of the figures, the upper bound is shown by a $+$ for the Full Duplex case, and by a x in the Half Duplex case. The achieved lower bounds (through coloring) are shown as a histogram with the full duplex results on the left side and the half duplex results on the right side for a given set of flows. In all cases we assumed that each time slot is 0.01 time units.

6.1 Example 1: Grid

In the first example, we consider a 7×7 grid, with 49 nodes and 84 bi-directional links. The destinations are randomly chosen from one of the four corners of the grid. and the sources are picked at random from the rest of the grid. A node has at most 4 neighbors in a grid, while the destinations have only 2 neighbors each. Therefore, we investigate the per-flow throughput for $\Omega(v) = 1, 2, 3$, in order to compare the Full Duplex and the Half Duplex systems. The head-to-head comparison is shown in Figures 5,6, and 7, for the cases $\Omega(v) = 1, 2,$ and 3 respectively. The x-axis shows the number of flows in the simulation, and the y-axis is the per-flow throughput, in terms of the scaling factor λ^*.

In each graph, we show the per-flow throughput when the number of flows are $1, 5, 10, 15, 20, 25, 30,$ and 35. The

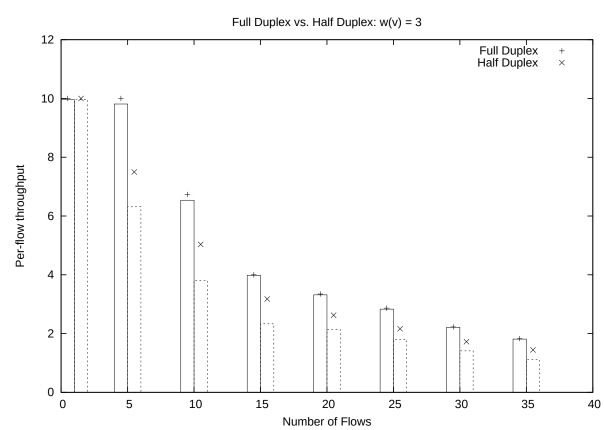

Figure 7: Grid - Full Duplex '+' vs. Half Duplex 'x': $\Omega(v) = 3$

No. of flows	$\Omega = 1$		$\Omega = 2$		$\Omega = 3$	
	UB	SF	UB	SF	UB	SF
1	10	99.6	10	99.6	10	99.6
5	5	100	10	98.1	10	98.1
10	3.333	99.8	6.724	97.3	6.732	97.1
15	2.5	99.8	4	99.4	4	99.5
20	2.0	99.8	3.3411	99.2	3.3458	99.2
25	1.4389	99.8	2.864	99.3	2.8662	98.8
30	1.2502	99.8	2.2358	99.2	2.2243	99.5
35	1.1112	99.8	1.8286	99.2	1.8288	99.2

Table 1: Full Duplex System: Effect of $\Omega(v)$

No. of flows	$\Omega = 1$		$\Omega = 2$		$\Omega = 3$	
	UB	SF	UB	SF	UB	SF
1	10	99.6	10	99.6	10	99.6
5	5	99.8	6.667	99	7.5	84.2
10	3.334	93.1	4.484	84.3	5.034	75.8
15	2.233	96.4	2.874	82.2	3.177	73.4
20	1.829	90.5	2.359	90.6	2.622	81.4
25	1.44	98.8	1.918	90.9	2.164	83.2
30	1.184	93.7	1.5445	90.4	1.723	82.1
35	1.008	92	1.304	85.1	1.442	77

Table 2: Half Duplex System: Effect of $\Omega(v)$

rectangle corresponds to the achieved flow from the greedy coloring, while the point above the bar represents the linear programming bound. Our first observation is that greedy coloring performs very close to optimal for the Full Duplex system, achieving 95% and above performance in all cases. In fact, the number of colors used by the greedy algorithm is only one more than the optimal. For the Half Duplex case, the greedy algorithm performs within 70% of the upper bound in all cases.

When there is only one receiver unit ($\Omega(v) = 1$), there is not much of a difference in performance between the Half Duplex and the Full Duplex cases, for a small number of flows. This is because there exist at least two independent paths from the source to the destination, and in such a scenario, the Half Duplex throughput will be equal to the single path Full Duplex throughput. Note that with 35 flows, the network is very loaded, as the per-flow throughput drops below 1 for the Half Duplex case with $\Omega(v) = 1$.

For $\Omega(v) = 2, 3$, the difference between these two systems becomes very pronounced. As before, when the number of flows is very small, there is no difference, but when the network gets loaded, the intermediate nodes in the grid can handle more load with the additional resources, represented by the $\Omega(v)$, and make use of the Full Duplex system to achieve higher per-flow throughput.

We now look at the performance of each individual system as we increase the number of receivers (radios), $\Omega(v)$. We tabulate the results shown in the figures in Tables 1 and 2. For each $\Omega(v) = 1, 2, 3$, we list the upper bound from the linear program (UB) and the schedulable flow obtained from the greedy coloring algorithm (SF), as a percentage of the upper bound.

It can be seen from the tables that per-flow throughput increases as Ω increases. The gain improves as the load (number of flows) increases. In addition, the improvement is more when we migrate from $\Omega = 1$ to $\Omega = 2$, than when $\Omega = 3$. This can interpreted as the dependency of Ω on the number of neighbors that use a particular node to relay information. The more neighbors that a node relays information for, the more performance gain achieved by using a larger $\Omega(v)$. However, since a node can transmit to only one node at a time, there is not much use in receiving from a lot of neighbors at once, since the transmit link will become a bottleneck.

6.2 Example 2: Random Graphs

We tested many random topologies with varying number of nodes. We present one such example in this paper. The results shown here are representative of the results with random graphs in general. In this example, 15 nodes were randomly distributed in a 1000 meter $\times$1000 meter square. All nodes within a distance of 200 meters from a given node were assumed to have direct communication with the node. The link speed was normalized to 1 units. As before, each node has unit rate to send to every node in the network. There were a total of 28 links in the graph, which is shown in Figure 8.

The number of flows were varied from 2-10 in steps of 2 and the achievable per-flow throughput is plotted in Figures 9,10, and 11. The number of flows are shown on the x-axis, while the per-flow throughput is shown in the y-axis. For $\Omega = 1$, the difference between Half Duplex and Full Duplex systems is more pronounced here because for most sources, almost all of the flow is routed on a single path, where simultaneous transmission and reception yields significant benefits. For $\Omega(v) = 2$, greedy coloring on the Full Duplex system performed slightly worse, with the results at least 92% of the optimal. The coloring performs near optimally for $\Omega(v) = 3$, however. For the Half Duplex system, the coloring always is within 70% of the upper bound.

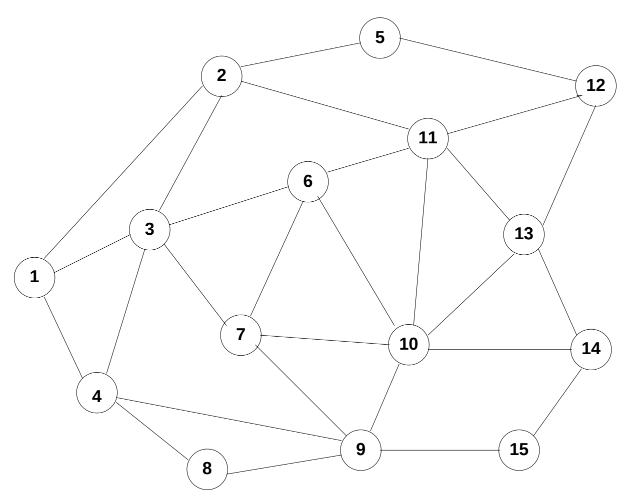

Figure 8: 15 Node Example

The simulations confirm that a Full Duplex system easily outperforms the Half Duplex system even for $\Omega = 1$. In addition, there are clear throughput gains when we have

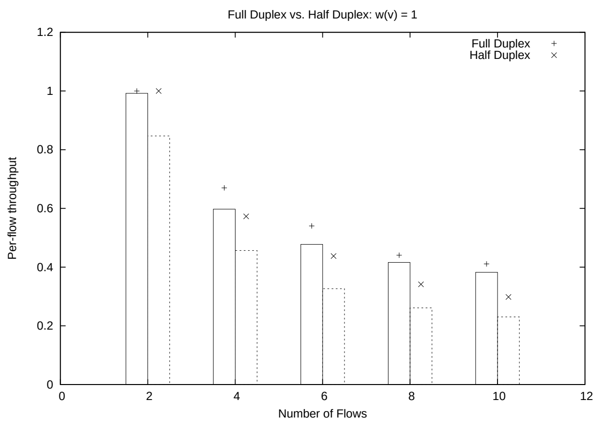

Figure 9: Random graph - Full Duplex '+' vs. Half Duplex 'x': $\Omega(v) = 1$

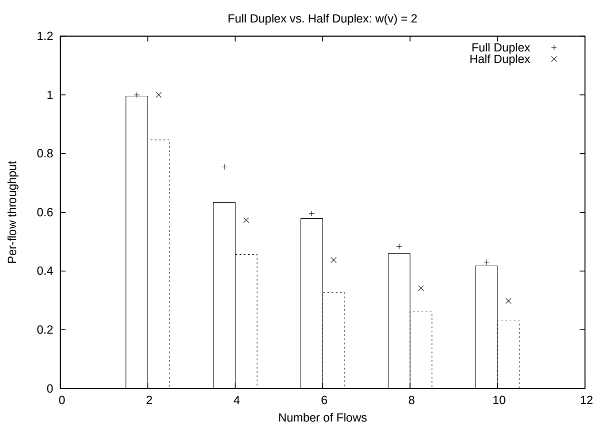

Figure 10: Random Graph - Full Duplex '+' vs. Half Duplex 'x': $\Omega(v) = 2$

the capability of receiving from more than one neighbor, even for a Half Duplex system. These are considerations that a system designer can use during the design of a multi-hop wireless network.

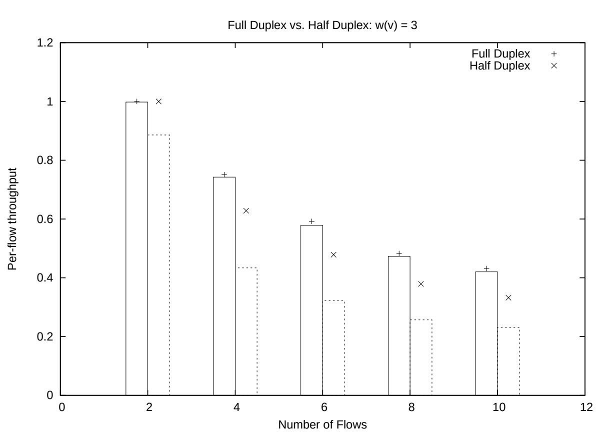

Figure 11: Random Graph - Full Duplex '+' vs. Half Duplex 'x': $\Omega(v) = 3$

7 Conclusions

We studied algorithms for routing flows and scheduling transmissions in multi-hop wireless mesh networks. We considered networks with only primary interference, where the only constraint on nodes is that each node can be transmitting to at most one node at any time and/or receiving from at most $\Omega(v)$ neighbors at a time. This is applicable in practice due to the availability of multiple orthogonal channels in such networks that permit neighboring links to be allotted non-interfering channels. We then look at two main types of networks: Full Duplex and Half Duplex. The approach that we use is to develop tight necessary conditions and solve this as a linear programming problem. We developed FPTAS which are very simple to implement. The scheduling problem is solved as a coloring problem. We characterize the necessary and sufficient conditions, and provide efficient coloring algorithms that guarantee 50 % of the optimal solution. In all the cases that we tested, our approach is within 15-20 % of the optimal solution. Our approach can be used by system designers for capacity analysis and node placement in multi-hop wireless networks.

8 Acknowledgements

We thank our editor Sergio Palazzo and the various anonymous referees for their valuable suggestions to help improve the quality of this paper.

References

[1] Kodialam, M., and Nandagopal, T., "Characterizing Achievable Rates in Wireless Multi-hop Networks: The Joint Routing and Scheduling Problem", *ACM Mobicom'03*, September 2003.

[2] IEEE Wireless LAN Working Group, "http://groups.ieee.org/802/11", October 2004.

[3] IEEE Broadband Wireless Access Working Group, "http://groups.ieee.org/802/16", October 2004.

[4] Padhye, J., Draves, R., and Zill, B., "Routing in multi-radio multi-hop wireless mesh networks", *ACM Mobicom*, September 2004.

[5] Paulraj, A., Gore, D., Nabar, R., and Bolcskei, H., "An Overviewof MIMO Communications – A Key to Gigabit Wireless", *Proc. of the IEEE*, 92(2), pp. 198-218, February 2004.

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

[7] Ahuja, R.K., Magnanti, T.L., Orlin, J.B., "*Network Flows: Theory, Algorithms, and Applications*", Prentice Hall, 1993.

[8] Hajek, B., and Sasaki, G., "Link Scheduling in Polynomial Time", *IEEE Transactions on Information Theory*, 34(5), pp. 910-917, 1988.

V	set of vertices, $n = \mid V \mid$	E	set of links, $m = \mid E \mid$
G	network graph	G_S	scheduling multi-graph
$t(e), e \in E$	transmitting node of link e	$r(e), e \in E$	receiving node of link e
$c(e)$	capacity of link e	$\Omega(v)$	max number of parallel receptions at node v
$N_{in}(v)$	set of incoming links at node v	$N_{out}(v)$	set of outgoing links at node v
$N(v)$	$N_{in}(v) \cup N_{out}(v)$, set of links incident at v	τ	length of a time slot
$f_k(e)$	flow rate on link e for commodity k	$f(e)$	$\sum_k f_k(e)$, total flow rate on link (e)
L_ψ	ψ-chromatic index: for Half-Duplex network	L_ϕ	ϕ-chromatic index: for Full Duplex network
ϵ	desired level of accuracy for FPTAS	$x(P)$	flow on a path P in the network, G
$f(v, P)$	$\theta(v, P)^{-1}$, max flow through node v on path P		

Table 3: Index of symbols used in the paper

[9] Grotschel, M., Lovasz, L., and Schrijver, A., "The Ellipsoid Method and its Consequences in Combinatorial Optimization", *Combinatorica*, 1(2), pp. 169-197, 1981.

[10] Baker, D.J., Wieselthier, J.E., and Ephremides, A., "A Distributed Algorithm for Scheduling the Activation of Links in a Self-Organizing Mobile Radio Networks", *IEEE Int. Conference Communications*, 1982, pp. 2F6.1-2F6.5.

[11] Post, M.J, Kershenbaum, A.S. and Sarachik, P.E., " Scheduling Multi-hop CDMA Networks in the Presence of Secondary Conflicts", *Algorithmica*, 1989, pp. 365-393.

[12] Jain, K., Padhye, J., Padmanabhan, V., and Qiu, L., "Impact on Interference on Multi-hop Wireless Network Performance", *ACM Mobicom'03*, September 2003.

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

[14] Wieselthier, J.E., Barnhart, C.M., and Ephremides, A., "A Neural Network Approach to Routing Without Interference in Multi-hop Networks", *IEEE Transactions on Communications*, 1994.

[15] Shahrokhi, F., and Matula, D., "The Maximum Concurrent Flow Problem", *Journal of the ACM*, 37, pp. 318-334, 1990.

[16] 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.

[17] Kodialam, M., and Nandagopal, T., "Charactering Achievable Rates in Multi-hop Wireless Mesh Networks with Orthogonal Channels", *Bell Laboratories Technical Report*, October 2004.

[18] Karakostas, G., "Faster Approximation Schemes for Fractional Multi-commodity Flow Problems", *13th ACM/SIAM Symposium on Discrete Algorithms*, pp. 166-173, 2002.

[19] Hakimi, S.L., and Kariv, O., "A Generalization of Edge-Coloring in Graphs", *Journal of Graph Theory*, 10, pp. 139-154, 1986.

[20] Shannon, C.E., "A Theorem on Coloring the Lines of a Network", *J. of Math. Physics*, 28, pp. 148-151, 1949.

[21] Nishizeki, T., and Kashiwagi, K., "On the 1.1 Edge-Coloring of Multi-graphs", *SIAM Journal of Discrete Math.*, 3(3), pp. 391-410, 1990.

A Index of Symbols

Table 3 gives a list of frequently used symbols and their meaning.