

Joint Routing and Scheduling in Multi-hop Wireless Networks with Directional Antennas*

Partha Dutta
IBM Research India
parthdut@in.ibm.com

Vivek Mhatre
Motorola Inc.
vivekmhatre@motorola.com

Debmalya Panigrahi
CSAIL, MIT
debmalya@mit.edu

Rajeev Rastogi
Yahoo Research India
rrastogi@yahoo-inc.com

Abstract—Long-distance multi-hop wireless networks have been used in recent years to provide connectivity to rural areas. The salient features of such networks include TDMA channel access, nodes with multiple radios, and point-to-point long-distance wireless links established using high-gain directional antennas mounted on high towers. It has been demonstrated previously that in such network architectures, nodes can transmit concurrently on multiple radios, as well as receive concurrently on multiple radios. However, concurrent transmission on one radio, and reception on another radio causes interference. Under this scheduling constraint, given a set of source-destination demand rates, we consider the problem of satisfying the maximum fraction of each demand (also called the *maximum concurrent flow problem*). We give a novel joint routing and scheduling scheme for this problem, based on linear programming and graph coloring. We analyze our algorithm theoretically and prove that at least 50% of a satisfiable set of demands is satisfied by our algorithm for most practical networks (with maximum node degree at most 5).

I. INTRODUCTION

Standardization of the air interface for IEEE 802.11 and 802.16 has resulted in rapid production and deployment of WiFi and WiMAX based mesh networks. Such multi-hop mesh networks have several applications including wireless backhauling for 3G cellular networks, and low cost networking for rural connectivity [1], [2], [3], [4]. In such networks, one typically makes a distinction between *long-distance links* and much shorter *local access links* [5]. In this paper, we are concerned only about long-distance links which set up the backbone of the network. These links are set up by high-gain directional antennas mounted on top of tall towers to achieve Fresnel clearance above obstructions [6], [7]. Typically, a subset of network nodes (called *gateways*) have direct connectivity to the Internet. The remaining nodes (called *mesh nodes*) use multi-hopping to access the Internet through the gateway nodes.

The focus of this paper is the design of *joint* routing and scheduling algorithms for throughput maximization in such multi-hop wireless mesh networks.

We assume that nodes are equipped with multiple radios and directional antennas. Directional antennas are helpful in mitigating interference and improving spatial reuse [8], [9]. A node communicates with its neighboring nodes using line-of-sight point-to-point wireless links. Each antenna is

connected to a single radio at the node. For such a setting, it has been experimentally demonstrated on commercial off-the-shelf wireless equipment [8], [10] that either (i) concurrent transmission on all the radios of a node, or (ii) concurrent reception on all the radios of a node is possible. This is because the directional antennas effectively isolate the signals of all the incoming links. However, if one radio is transmitting during a given time slot, and another radio on the same node is receiving concurrently, then due to radiation leakage and the proximity of the two antennas, the transmitted signal can drown the received signal. Hence transmission on one radio, and concurrent reception on another radio of the same node is not permitted. We refer to this operation of concurrent transmission or concurrent reception on all the radios of a node as SynOP (synchronous operation). For the above network model, we consider the problem of joint routing and scheduling to best meet a given set of source-destination demands.

Kodialam and Nandagopal have studied the joint routing and scheduling problem for the *omni-directional* antenna setting with single radio nodes in [11], and multi-radio nodes in [12]. In [13], Narlikar et al. consider the multi-radio setting with directional antennas, and propose heuristic algorithms for joint routing and scheduling. We show that the routing-scheduling problem that arises in multi-radio directional antenna setting is fundamentally different from the multi-radio omni-directional antenna scenario considered in [11], [12]. Furthermore, in contrast to [13], we provide a provably near-optimal scheme for joint routing and scheduling for this scenario. Our main contributions are:

- Formulation of the problem as an (exponential-sized) linear program (LP) with *joint* routing and scheduling constraints.
- A novel scheduling algorithm based on directed edge coloring in a multi-graph.
- An algorithm for joint routing and scheduling based on solving a modified (polynomial-sized) version of the above LP, and then using the scheduling algorithm to produce a valid schedule for the LP solution.
- Analysis of the above algorithm showing that for a large class of network graphs (maximum node degree of 9), at least 40% of the maximum concurrently satisfiable demand for each source-destination pair is satisfied by our algorithm. For network graphs with maximum node

*This work was done when the authors were at Bell Labs Research India.

degree of 5, which encompass most practical mesh network deployments, our solution has a provable performance of at least 50% of the optimum.

Although we focus on only a TDMA model, our results can be easily extended to a mix TDMA-FDMA model, as well as the OFDMA model of IEEE 802.16.

II. SYSTEM MODEL AND PROBLEM FORMULATION

Consider a mesh network graph $G = (V, E)$ where V is the set of mesh nodes, and $E \subseteq V \times V$ is the set of directed links. A link is formed between two nodes by aligning the directional antennas on each of the nodes. Results in [8] show how to dimension the transmit powers of radios of different nodes so that desired SINR (Signal to Interference and Noise) requirements are met despite side lobe leakage, and concurrent reception on multiple directional links at a node is possible.¹

In the network graph, the same physical link is modeled using two opposite directed links, and we say that a directed link is active when the underlying physical link is active in that direction. Throughout this paper links refer to directed links. Let $C(e)$ be the capacity of link e . Since the nodes are static, and the links are line-of-sight, we assume that the link capacities are fixed, i.e., there are no shadow-fading time variations. However, note that slow time variations in link capacities can be easily incorporated in our framework by adapting the routing and scheduling policies over longer time scales.

We are given a set of M demands (rates), where the k^{th} demand requires a rate of r_k from source $s_k \in V$ to destination $d_k \in V$. If all the traffic is only between the mesh nodes and the Internet, and if the gateway nodes are assumed to be directly connected to the Internet through a high speed connection (a typical mesh network architecture), then the above model can be easily modified as follows. We add a node (corresponding to the Internet) to the network graph, and connect it to all the gateways using links of infinite capacity. We make the following assumptions about the capabilities of the nodes, and the interference constraints:

- Each node has a dedicated radio and a corresponding directional antenna for communicating with each of its neighbors.
- All the radios operate on a common channel.
- Transmissions are synchronized in time (TDMA operation), and in each slot, a subset of links in the network are activated.
- During each time slot, a node can concurrently transmit on all its outgoing links, or concurrently receive on all its incoming links. However, concurrent transmission on one link and reception on another link within the same slot at a node is not allowed.

¹Note that, when omni-directional antennas are employed, it is *not possible* to dimension the transmit powers for concurrent reception. Consequently, with omni-directional antennas, for a given time slot/channel, only a single radio can be actively receiving, while with directional antennas all the radios can concurrently receive packets. This important difference between omni-directional and directional antennas results in two fundamentally different scheduling problems.

The last constraint arises because of side-lobe radiation leakage from the directional antennas. Due to this leakage, a transmission on a radio can completely drown a reception on a nearby radio. Thus in a slot, the set of nodes can be bipartitioned into a set of transmitters and a set of receivers. This immediately implies that the set of active links in each slot forms a *directed bipartite subgraph* from the transmitters to the receivers.² In the rest of this paper, unless stated otherwise, a bipartite graph refers to a directed bipartite graph.

A *schedule* of length T specifies for each slot 1 to T , the set of links that are active in that slot. In our setting, a schedule is *valid* if the set of links that are active in any time-slot forms a directed bipartite graph. In other words, a valid schedule is a sequence of bipartite subgraphs. Note that the bipartite subgraphs need not be disjoint and the same link may appear in multiple time-slots, either as part of different directed bipartite subgraphs or in the same subgraph which is repeated in multiple time-slots. The *utilization* of a link e in the schedule is then defined as the ratio of the number of slots in which e is active to the length of the schedule. The effective capacity of link e in a schedule is therefore the product of its utilization and its total capacity $C(e)$.

A *routing* specifies the amount of flow on every link for every source-destination demand. The *link flow vector* of a routing specifies the total flow load (due to all source-destination demands) on each link. We say that a routing can be *scheduled* if there is a valid schedule such that the effective capacity of every link in the schedule is at least as much as the total flow load on the link. Note that some routings may not have a feasible schedule. The *concurrent flow value* of a routing is the largest λ such that the routing satisfies at least λ fraction of every source-destination demand. For a graph G and a set of source-destination demands, the *maximum concurrent flow* (or max flow) is defined as the maximum λ such that there is a routing that has concurrent flow value of λ . Then, our joint routing and scheduling problem is the following: *Given a network graph and a set of source-destination demands, among all routings that can be scheduled in our model, find a routing (and its corresponding schedule) that has the maximum concurrent flow.*

An LP formulation. Let \mathcal{B} be the set of all directed bipartite subgraphs of the given graph $G = (V, E)$. By simple counting arguments, $|\mathcal{B}| = 2^n - 2$ where $n = |V|$. Thus, any feasible activation of links in a time slot is one of the subgraphs in \mathcal{B} , and any schedule can be represented as a weighted combination of elements $B_i \in \mathcal{B}$. Here the weight of a particular directed bipartite subgraph B_i is the fraction of time-slots in which B_i is active. For a schedule π , let $w_{B_1}, w_{B_2}, \dots, w_{B_{|\mathcal{B}|}}$ be the associated weights. Let $\mathbf{1}_{e, B}$ be the indicator function which is 1 if link e is present in the directed bipartite subgraph $B \in \mathcal{B}$.

²A directed bipartite graph divides the set of nodes into two sets, the transmitter set and the receiver set, such that, each edge in the graph originates from a node in the transmitter set, and terminates at a node in the receiver set.

$$\begin{aligned}
& \text{Maximize } \lambda & (1) \\
& \text{Subject to:} \\
& f(e) = \sum_{k=1}^M f(e, k), \forall e \in E & (2) \\
& \lambda r_k \leq \sum_{e \in N_{out}(s_k)} f(e, k) - \sum_{e \in N_{in}(s_k)} f(e, k), & (3) \\
& \quad \forall k = 1 \dots M \\
& \sum_{e \in N_{in}(v)} f(e, k) = \sum_{e \in N_{out}(v)} f(e, k), & (4) \\
& \quad \forall v \neq s_k, d_k, \forall k = 1 \dots M \\
& f(e, k) \geq 0, \forall e \in E, \forall k = 1 \dots M & (5) \\
& f(e) \leq C(e) \cdot \sum_{i=1}^{|\mathcal{B}|} w_{B_i} \cdot \mathbf{1}_{e, B_i}, \forall e \in E & (6) \\
& \sum_{i=1}^{|\mathcal{B}|} w_{B_i} = 1, w_{B_i} \geq 0, \forall B_i \in \mathcal{B} & (7)
\end{aligned}$$

Fig. 1. LP for joint scheduling-routing problem

Let $N_{in}(v)$ represent the set of incoming links at node v , and $N_{out}(v)$ represent the set of outgoing links at node v . As mentioned in Section II, we are given a demand vector of M flows, such that, demand k consists of source node s_k and destination node d_k with a desired rate r_k . Let $f(e, k)$ denote the load on link e due to demand k , and let \mathcal{S} and \mathcal{D} denote the set of source and destination nodes for these demands. Then the maximum concurrent flow problem in our model can be formulated as the LP in Figure 1. In the above LP, Eq. (2)-(5) are the regular flow constraints, while Eq. (6)-(7) are the scheduling constraints. The LP finds the optimum weights w_{B_i} for all the bipartite graphs, and the corresponding link flow values. While solving the above problem gives the optimal max-flow solution, its computational complexity is very high. This is because the number of bipartite subgraphs in \mathcal{B} is exponential in the number of nodes, and hence the LP has exponential number of variables w_{B_i} . We therefore look for approximation algorithms for the above problem. Our general plan in designing the approximation algorithms is as follows. Consider a particular routing with a link flow vector $f : E \rightarrow \mathcal{R}_0^+$ representing the total flow load on each directed link in E . The corresponding routing can be scheduled in our setting provided there exists a schedule which can achieve a utilization of at least $f(e)/C(e)$ for each link $e \in E$. We first derive a set of necessary conditions for scheduling a particular link flow vector. In other words, if there exists a schedule that achieves the required utilization, then these conditions have to be satisfied. The optimal flow vector given by the solution to the LP must therefore satisfy the necessary conditions. Then, we derive a set of sufficient conditions for the same problem, i.e. if these sufficient conditions are satisfied, then the link flow vector can be scheduled. Our sufficient conditions are based on a scheduling algorithm that can schedule any link flow vector that satisfies these conditions. Adding the sufficient conditions to the LP formulation possibly changes the optimal value of λ . However, by comparing the necessary and sufficient conditions, we bound the decrease in the optimal

LP value when the sufficient conditions are added to the LP. The sufficient conditions replace the exponential number of constraints involving w_{B_i} in the LP and make the LP tractable. We solve the LP to obtain a flow vector which can now be scheduled since it satisfies the sufficient conditions. The ratio between the necessary and sufficient conditions gives the approximation bounds on our algorithm. We may note that the overall plan of our algorithm is similar to [11], however our necessary and sufficient conditions as well as our scheduling algorithms are very different.

III. NECESSARY CONDITION

To schedule a link flow vector f corresponding to a routing, each link e must have a utilization of at least $f(e)/C(e)$. Since a node cannot activate an incoming and an outgoing link simultaneously, the sum of utilization of an incoming link and that of an outgoing link (at the same node) must add up to at most 1.

Theorem 1: If a link flow vector f can be scheduled, then

$$\frac{f(e)}{C(e)} + \frac{f(e')}{C(e')} \leq 1, \forall v \in V, e \in N_{in}(v), e' \in N_{out}(v). \quad (8)$$

The above necessary condition at a node depends only on the most heavily loaded pair of incoming and outgoing links. This is because in our model we are allowed SynOP at a node (synchronous reception on all the links, and synchronous transmission on all the links). In sharp contrast, the model considered in [11] has omni-directional antennas, and therefore only allows one link to be active per channel, at a node. Thus their necessary condition depends on the sum of the loads on *all* links of a node.

IV. SUFFICIENT CONDITIONS USING EDGE COLORING OF A DIRECTED MULTIGRAPH

In this section, we give a new scheduling algorithm and an associated set of sufficiency conditions. We first introduce some terminology that we use throughout this section. A *multi-graph* is a graph in which there could be multiple parallel edges between two nodes. In the rest of the paper, unless explicitly specified, by multi-graph, we refer to a directed multi-graph. The *underlying simple (directed) graph* of a multi-graph is produced by replacing each multi-edge by a single (directed) edge. The *multiplicity* of an edge in the underlying simple graph is defined as the number of copies of the edge that are present in the multi-graph. Any two different copies of the same simple edge in the multi-graph are said to be *parallel* edges. Note that an edge from u to v is *not* a parallel edge of an edge from v to u . Such edges are referred to as *anti-parallel* or *opposite* edges. For any directed graph (either a multi-graph or a simple graph), the *corresponding undirected graph* is produced by replacing each set of parallel and anti-parallel edges (i.e. all edges between two vertices u and v in both directions) by a single undirected edge.

Let τ be the duration of a time slot. Since $C(e)$ bps is the capacity of link e , in order to carry a traffic of $f(e)$ bps, link

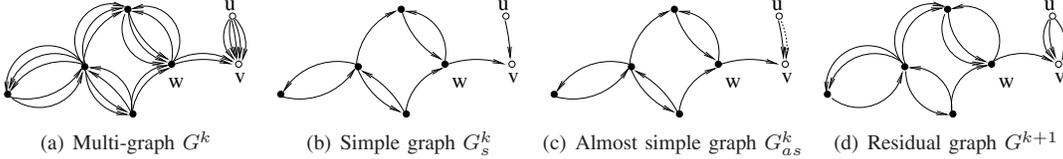


Fig. 2. Evolution of residual graph G^k in the Scheduling Algorithm. Edge between u and v is tight, and therefore contributes two copies in the almost simple graph G_{as}^k , while all other edges contribute one copy. The solid edges in G_{as}^k are in G_s^k , while the dotted edge belongs to the set E_{as}^k .

e should be scheduled in at least $\frac{f(e)}{C(e)\tau}$ slots every second. We therefore associate weight $w(e)$ with link e ,

$$w(e) = \frac{1}{\tau} \cdot \frac{f(e)}{C(e)}, \quad (9)$$

and consider a multi-graph \mathcal{G} where link $e = (u, v)$ is replaced by $w(e)$ directed links from u to v . (We assume that $f(e)/C(e)$ are rational, and we choose τ so that all the $w(e)$ are integers.)

Now, our scheduling problem reduces to allocating time-slots to each edge in the multi-graph \mathcal{G} , where the duration of each time-slot is τ seconds. Considering each time-slot as a color, the slot allocation problem is an edge coloring problem for a directed multi-graph such that, the following two types of interference constraints are satisfied.

Local Overlap Constraint: At every node, the color (time-slot) allocated to an outgoing edge is different from the colors allocated to the incoming edges. This ensures that no node concurrently transmits over one radio, and receives over another radio.

Parallel Overlap Constraint: No two parallel edges are allocated the same color. If two parallel edges get the same color, then it implies that the corresponding link needs to be activated twice in the same time-slot; clearly, this is not possible.

A similar multi-graph edge coloring problem was considered in [11] for *undirected multi-graphs* where the local overlap constraints don't exist. In another related work in [10], a similar edge coloring problem was studied in the context of *simple (directed) graphs* where the parallel overlap constraint does not exist. The authors called this problem the *directed edge coloring* problem or the DEC problem. In the following subsection, we provide a brief overview of the DEC algorithm proposed in [10].

DEC algorithm for simple graphs. Let G be the given simple directed graph with maximum vertex degree Δ . Consider the vertex coloring of the undirected graph corresponding to G with $K \leq \Delta + 1$ colors.³ Let each vertex color class (set of vertices with the same color) form a supervertex and let N be the set of supervertices in G . First, note that each supervertex is an independent set of vertices (i.e., no pair of vertices with

³Recall that the vertex coloring problem on an undirected graph aims to color all the vertices in the graph using the minimum number of colors such that no two vertices sharing an edge are assigned the same color. It is well-known that the vertex coloring problem is NP-hard [14]. However, for our purpose a (sub-optimal) polynomial-time vertex coloring algorithm suffices: the well-known greedy algorithm using $\Delta + 1$ colors where Δ is the maximum degree of a vertex in a graph. Thus we assume $K \leq \Delta + 1$.

K	$\xi(K)$	K	$\xi(K)$
2	2	6	4
3	3	9	5
4	4	10	5
5	4	20	6

Fig. 3. Values of $\xi(K)$ for different values of K

the same color have an edge connecting them). Thus, the graph induced by G on these supervertices $G_N = (N, E_N)$ has all the edges in E , except that the terminals of each edge are no longer vertices in V but the corresponding supervertices in N . Now, suppose we are given $n = \xi(K)$ edge colors, where $\xi(K)$ is the smallest integer k such that $\binom{k}{\lfloor k/2 \rfloor} \geq K$. Then, for each supervertex $v \in N$, we assign a distinct set of $\lfloor n/2 \rfloor$ edge colors. By the definition of $\xi(K)$, $\binom{n}{\lfloor n/2 \rfloor} \geq K$; therefore, we have sufficiently many edge colors for each supervertex to get a distinct edge color set. The edge color set at a supervertex v represents the colors that may be used to color an edge leaving v . Now, it can be easily shown that corresponding to each edge in G , there is at least one edge color which is in the edge color set at its origin supervertex and is not in the edge color set at its destination supervertex. This color is assigned to the corresponding edge. Thus, this algorithm uses $\xi(K)$ colors for the DEC problem. It was shown in [10] that $\xi(K) \simeq \log K$ for large values of K . For smaller values of K , see Figure 3.

A. Multi-DEC scheduling algorithm

Since our problem of edge coloring the multi-graph generated by weights $w(e)$ in Eq. (9) is similar to the DEC problem, we call it the *directed multi-edge coloring* problem or the multi-DEC problem. Formally, we are given a multi-graph $\mathcal{G} = (V, \mathcal{E})$, and we need to color the edges of this multi-graph with the minimum set of colors such that the local and the parallel overlap constraints are satisfied. Roughly speaking, the algorithm we propose for multi-DEC splits the given multi-graph into several ‘‘almost’’ simple subgraphs, and edge colors each of these subgraphs using a modified DEC algorithm.

We first present some definitions required to describe our scheduling algorithm. A (directed or undirected) multi-graph is said to be *almost simple* if the maximum multiplicity of any edge in the graph is 2. The *weight* of a node v in a multi-graph $H = (V, \mathcal{E})$ (denoted by $W(v, H)$) is the sum of multiplicities of the most heavily loaded incoming and outgoing links of the node. Thus, if $w(e)$ is the multiplicity of edge e in H , then

$$W(v, H) = \max_{(u,v), (v,w) \in \mathcal{E}} \{w(u, v) + w(v, w)\}.$$

Also, let $W_{max}(H) = \max_{v \in V} W(v, H)$. A directed edge (u, v) is said to be *tight* if $w(u, v) = W_{max}(H)$. Similarly, a

vertex v is said to be *tight* if $W(v, H) = W_{max}(H)$. For example, the edge between nodes u and v in Fig. 2(a) is tight in multi-graph G^k , since $W_{max}(G^k) = W(u, G^k) = W(v, G^k) = 5$.

Let \mathcal{G}_s and \mathcal{G}_u be the underlying simple graph and corresponding undirected graph of the multi-graph \mathcal{G} that is to be directed edge colored. Suppose that \mathcal{G}_u can be vertex-colored using K colors. Then, from [10], \mathcal{G}_s or any of its subgraphs can be directed edge colored using $\xi(K)$ colors. We now give a scheduling algorithm that assigns colors to the edges of \mathcal{G} .

Scheduling Algorithm. Case (i): $\xi(K) = 1$: \mathcal{G}_s is a directed bipartite graph. We partition \mathcal{G} into $W_{max}(\mathcal{G})$ directed bipartite graphs, each of which is given a distinct color.

Case (ii): $\xi(K) \geq 2$: We use the following iterative procedure. Let G^k be the subgraph of \mathcal{G} used in the k^{th} iteration of the algorithm. Initialize $G^1 = \mathcal{G}$. In each iteration k , perform the following steps (each iteration uses a new set of colors):

- 1) Form an almost simple graph, G_{as}^k , as follows. Initialize $G_{as}^k = G_s^k$, the underlying simple graph of G^k . If edge (u, v) is tight in G^k , then add another copy of (u, v) to G_{as}^k (see Fig. 2(c)). Let these second copies of edges form the set \hat{E}_{as}^k .
- 2) Color the edges in G_s^k using the DEC algorithm from [10] (at most $\xi(K)$ new colors). For any edge $e \in \hat{E}_{as}^k$, assign one of the $\xi(K)$ colors (that were used in coloring the edges in G_s^k) excluding the color assigned to its parallel edge in G_s^k .
- 3) The subgraph G^{k+1} used in the next iteration is the *residual graph* obtained by removing the edges colored in this iteration from G^k (see Fig. 2(d)). If G^{k+1} has no edge then the algorithm terminates. Otherwise, increment k , and go to Step 1.

It can be shown that the color assignment given by the above algorithm does not violate local and parallel overlap constraints [15]. Also we can show the following theorem (proof in [15]) on the number of colors required by the above algorithm.

Theorem 2: The multi-DEC algorithm uses W_{max} colors when $\xi(K) = 1$, and $\lceil \xi(K)W_{max}/2 \rceil$ colors when $\xi(K) \geq 2$, where the corresponding undirected graph of \mathcal{G} can be vertex colored using K colors.

This theorem gives the following sufficiency conditions.

Corollary 1: Using the Scheduling Algorithm, a link flow vector f can be scheduled if

$$\frac{f(e)}{C(e)} + \frac{f(e')}{C(e')} \leq \min \left\{ 1, \frac{2}{\xi(K)} \right\} \quad (10)$$

for each pair of edges $e \in N_{in}(v)$ and $e' \in N_{out}(v)$ for each vertex v .

B. Joint Routing and Scheduling Algorithm

We exploit the sufficiency conditions of Corollary 1 in our algorithm. In the original LP formulation (Eq. (2)-(7)), we replace the scheduling constraints (Eq. (6)-(7)) with the

constraints:

$$\frac{f(e)}{C(e)} + \frac{f(e')}{C(e')} \leq Y, \quad \forall v \in V, e \in N_{in}(v), e' \in N_{out}(v) \quad (11)$$

where $Y = \min \left\{ 1, \frac{2}{\xi(K)} \right\}$. Note that the above constraint amounts to no more than Δ^2 constraints for each node. Thus, the LP is tractable. The link flow vector in the solution of the LP is now scheduled using the Scheduling Algorithm. The approximation ratio of this algorithm is given by the following theorem (full proof in [15]).

Theorem 3: The above LP (along with the Scheduling Algorithm) provides a $2/\xi(K)$ -approximation to the joint routing and scheduling problem when $\xi(K) > 2$, and provides an exact solution for the cases of $\xi(K) = 1$ and $\xi(K) = 2$.

Proof Sketch: Comparing the necessary condition in Eq. (8) and the sufficient condition in Eq. (10), we note if $\xi(K) = 1$ or $\xi(K) = 2$, the necessary and sufficient conditions are identical, and for $\xi(K) > 2$, their ratio is at most $2/\xi(K)$. Thus, any flow vector that satisfies the sufficient condition is at least $2/\xi(K)$ times the optimum. ■

Recall that, $K \leq \Delta + 1$, where Δ is a maximum degree of a node in a graph. Noting the values of $\xi(K)$ from Figure 3, we observe that for a large class of practical network deployments (maximum node degree of 9), our proposed algorithm obtains a solution that is within 40% of the optimum. For more typical network deployments (maximum node degree of 5), our solution comes to within 50% of the optimum.

REFERENCES

- [1] K. Chebrolu, B. Raman, and S. Sen. Long-distance 802.11b links: performance measurements and experience. In *Mobicom*, 2006.
- [2] R. Patra, S. Nedeveschi, S. Surana, A. Sheth, L. Subramanian, and E. Brewer. Wildnet: Design and implementation of high performance wifi based long distance networks. In *NSDI*, 2007.
- [3] E. Brewer, M. Demmer, M. Ho, R. Honicky, J. Pal, M. Plauché, and S. Surana. The challenges of technology research for developing regions. *IEEE Pervasive Computing*, 5(2):15–23, 2006.
- [4] P. Dutta, S. Jaiswal, and R. Rastogi. Routing and channel allocation in rural wireless mesh networks. In *Infocom*, 2007.
- [5] K. Chebrolu and B. Raman. Fractal: A fresh perspective on (rural) mesh networks. In *NSDR*, 2007.
- [6] S. Sen and B. Raman. Long distance wireless mesh network planning: problem formulation and solution. In *WWW*, 2007.
- [7] D. Panigrahi, P. Dutta, S. Jaiswal, K. V. M. Naidu, and R. Rastogi. Minimum cost topology construction for rural wireless mesh networks. In *Infocom*, 2008.
- [8] B. Raman and K. Chebrolu. Design and evaluation of a new mac protocol for long-distance 802.11 mesh networks. In *Mobicom*, 2005.
- [9] B. Raman. Channel allocation in 802.11-based mesh networks. In *Infocom*, 2006.
- [10] P. Dutta, S. Jaiswal, D. Panigrahi, and R. Rastogi. A new channel assignment mechanism for rural wireless mesh networks. In *Infocom mini-conference*, 2008.
- [11] M. Kodialam and T. Nandagopal. Characterizing achievable rates in multi-hop wireless mesh networks with orthogonal channels. *IEEE/ACM Transactions on Networking*, 13(4):868–880, 2005.
- [12] M. Kodialam and T. Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *Mobicom*, 2005.
- [13] G. Narlikar, G. Wilfong, and L. Zhang. Designing multihop wireless backhaul networks with delay guarantees. In *Infocom*, 2006.
- [14] M. R. Garey and David S. Johnson. *Computer and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [15] P. Dutta, V. Mhatre, D. Panigrahi, and R. Rastogi. Joint routing and scheduling in wireless mesh networks. *Bell Labs Technical Memorandum*, ITD-07-47872K, 2007.