Joint Channel Assignment and Routing for Throughput Optimization in Multi-radio Wireless Mesh Networks

Mansoor Alicherry Randeep Bhatia Li (Erran) Li*
Bell Laboratories, Lucent Technologies
{mansoor, randeep, erranlli}@bell-labs.com

ABSTRACT

Multi-hop infrastructure wireless mesh networks offer increased reliability, coverage and reduced equipment costs over their single-hop counterpart, wireless LANs. Equipping wireless routers with multiple radios further improves the capacity by transmitting over multiple radios simultaneously using orthogonal channels. Efficient channel assignment and routing is essential for throughput optimization of mesh clients. Efficient channel assignment schemes can greatly relieve the interference effect of close-by transmissions; effective routing schemes can alleviate potential congestion on any gateways to the Internet, thereby improving per-client throughput. Unlike previous heuristic approaches, we mathematically formulate the joint channel assignment and routing problem, taking into account the interference constraints, the number of channels in the network and the number of radios available at each mesh router. We then use this formulation to develop a solution for our problem that optimizes the overall network throughput subject to fairness constraints on allocation of scarce wireless capacity among mobile clients. We show that the performance of our algorithms is within a constant factor of that of any optimal algorithm for the joint channel assignment and routing problem. Our evaluation demonstrates that our algorithm can effectively exploit the increased number of channels and radios, and it performs much better than the theoretical worst case bounds.

Categories and Subject Descriptors

C.2.1. [**Computer-Communication Networks**]: Network Architecture and Design – Wireless communication

General Terms

Algorithms, Theory

*Research partially supported by NSF NRT Grant# ANI-0335244.

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

Keywords

Approximation algorithm, mathematical programming, graph theory, wireless mesh networks

1. INTRODUCTION

Wireless broadband networks are being increasingly deployed in a multi-hop wireless mesh network (WMN) configuration. These WMNs are being used on the last mile for extending or enhancing Internet connectivity for mobile clients located on the edge of the wired network. Commercial deployments of multi-hop wireless mesh networks (WMNs) are already in the works. For example, many US cities including Medford, Oregon and Chaska, Minnesota [1] have deployed mesh networks. Even big cities like Philadelphia, Pennsylvania are planning to deploy city-wide mesh networks. The deployed mesh networks will provide commercial Internet access to residents and local businesses [1].

In WMNs, the access points (or mesh routers) are rarely mobile and may not have power constraints. In addition these networks behave almost like wired networks in having infrequent topology changes, limited node failures etc. Although WMNs may be self-organizing, node additions and maintenance are still rare events. In addition since each mesh router may aggregate traffic flows for a large number of mobile clients, the aggregate traffic load of each mesh router changes infrequently. In infrastructure wireless mesh networks (IWMNs) [4] some mesh routers are also equipped with a gateway capability through which they interface with the wired network. In such networks traffic is mainly routed by the WMN wireless backbone between the mesh clients and the wired Internet and goes through the gateway nodes.

One of the major problem facing wireless networks is the capacity reduction due to interference among multiple simultaneous transmissions [11]. In wireless mesh networks providing mesh routers with multiple-radios can greatly alleviate this problem. With multiple-radios, nodes can transmit and receive simultaneously or can transmit on multiple channels simultaneously. However, due to the limited number of channels available the interference cannot be completely eliminated and in addition careful channel assignment must be done to mitigate the effects of interference. Several companies such as MeshDynamics [3] have recently announced the availability of multi-radio mesh network technology.

To make use of commodity 802.11 radios, a channel is assigned to a radio interface for an extended period of time as long as traffic demand or topology does not change. MAC protocols [5] where each radio interface can use different channels on a fast time scale such as on a per-packet ba-

sis are not supported in current 802.11 MAC. As observed in [19], assigning the first channel to the first radio, the second channel to the second radio and so on can be far from the optimal achievable performance. In addition channel assignment and routing are inter-dependent. This is because channel assignments have an impact on link bandwidths and the extent to which link transmissions interfere. This clearly impacts the routing used to satisfy traffic demands. In the same way traffic routing determines the traffic flows for each link which certainly affects channel assignments. Channel assignments need to be done in a way such that the communication requirements for the links can be met.

Heuristic approaches on channel assignments and load-aware routing [19, 18] are proposed to improve the aggregate throughput of WMNs and balance load among gateways. These heuristic approaches can still be far from the optimal performance the network can offer. Because aggregate traffic demands and network topology do not change frequently in IWMNs optimizations using measured traffic demands are feasible. The system management software can compute the optimal channel assignment and routing and configure each elements periodically. Routing protocols will still need to be run to handle topology changes. In this paper, we study the joint channel assignment and routing problem in multi-radio IWMNs. Our contributions are as follows.

- We present a formulation for the joint channel assignment, routing and scheduling problem that can model the interference and fairness constraints and is also able to account for the number of radios at each of the wireless nodes.
- We establish matching necessary and sufficient conditions under which interference free link communication schedule can be obtained and we design an efficient algorithm to compute such a schedule.
- We use a novel flow transformation technique to design an efficient channel assignment algorithm that can assign channels to node radios while ensuring that maximum data can be transmitted on specified traffic routes.
- We establish that our algorithm for the joint channel assignment, routing and scheduling problem is a constant factor approximation algorithm. To the best of our knowledge, this is the first constant factor approximation algorithm for the problem.
- Our evaluation shows that our algorithm can effectively exploit the increased number of channels and radios, and it performs much better than the worst case theoretical bounds.

The rest of the paper is organized as follows. We state our model and assumptions in Section 2. We describe our throughput optimization problem and give an overview of our algorithm in Section 3. We present our detailed algorithm in Section 4, 5, 6, 7 respectively. We show in Section 8 that our algorithm runs in polynomial time and is a constant factor approximation algorithm for the joint channel assignment, routing and scheduling problem. We evaluate our algorithms in Section 9. We present related work in Section 10 and present our conclusions and discuss future work in Section 11.

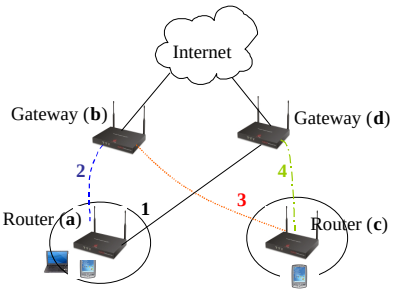


Figure 1: A wireless Mesh network with 4 nodes. Two of them, b and d are gateways. Each node has 2 interfaces each operating on a different channel among $1, 2, 3, 4$. Edge labels represent the channels used.

2. SYSTEM MODEL AND ASSUMPTIONS

This section contains basic definitions and concepts used in the rest of the paper.

2.1 System Architecture

This work pertains to multi-hop infrastructure wireless mesh networks (WMNs) [4], an example of which is shown in Figure 1. These networks consist of static wireless mesh routers and end mobile clients. The static wireless routers are equipped with traffic aggregation capabilities (e.g. Access Points) and provide network connectivity to mobile clients within their coverage areas. The wireless mesh routers themselves form a multi-hop wireless backbone for relaying the traffic to and from the clients. Some of the wireless mesh routers are equipped with gateway functionality to enable connectivity to the wired Internet. All infrastructure resources that the mobile client access (e.g. web servers, enterprise servers, Internet gateways) reside on the wired Internet and can be accessed via any wireless mesh router with gateway functionality. Thus, the wireless backbone of mesh routers mainly relays mobile clients traffic to and from the Internet via the routers with gateway functionality. Each wireless mesh router may be equipped with multiple wireless interfaces (radios) each operating on a particular channel.

We model the backbone of an infrastructure WMN as a directed graph $G = (V, E)$. Except for one node t representing the wired network, the nodes of the graph correspond to individual wireless mesh routers and for each such node $u \in V$ a value $I(u)$ denotes the number of wireless interfaces that it has. Among the nodes in V some of the nodes have gateway functionality and provide connectivity to the wired Internet (represented by node t) using high capacity bidirectional links. We denote these nodes by the set $V_G \subseteq V$. A wireless interface of a mesh router u operates on a single channel selected from the set $\mathcal{F}$. We assume there are K orthogonal channels (In this paper, when we refer to different channels, we mean orthogonal channels.) in $\mathcal{F}$ numbered from 1 to K. It may be noted that due to the wireless interference constraints there is no capacity advantage in equipping two different interfaces of a node with the same channel. We therefore assume that the interfaces of a node are equipped with distinct channels. Thus each wireless node u can be associated with a ordered set $F(u) \subseteq \mathcal{F}$ of $I(u)$ distinct channels, where the i-th interface of node

u operates on the i-th channel in $F(u)$. Each node u in V aggregates the user traffic from all the mesh clients that are associated with u. We denote the aggregate user traffic load on u by $l(u)$. The load $l(u)$ may be due to outgoing or incoming traffic. *However, for ease of exposition from now on we will assume that there is no incoming traffic to any wireless node from the wired Internet and hence $l(u)$ represents only outgoing traffic.* Our results easily extend (by flow reversal) to the more general case of both outgoing and incoming traffic. We assume $l(u)$ exhibits only long term variability and any such variations can be dealt with by re-routing and re-adjustment of the channel assignments. Thus we assume $l(u)$ to be a node dependent fixed value.

2.2 Wireless Transmission and Interference Model

For direct communication, two nodes need to be within *communication range* of each other, and need to have a common channel assigned to their interfaces. A pair of nodes that use the same channel and are within *interference range* may interfere with each other's communication, even if they cannot directly communicate. Node pairs using different channels can transmit packets simultaneously without interference. For example, in Figure 1, each node is equipped with 2 Network Interface Cards. The links shown between the nodes depict direct communication between them, and the channel used by a pair of nodes is shown as the number associated with the connecting link. This example network totally uses 4 distinct channels.

We denote by R_T the *transmission range* and by $d(u,v)$ the distance between the nodes u and v. An edge $(u,v) \in E$ if and only if $d(u,v) \leq R_T$ and implies that mesh router u can communicate with mesh router v directly (in one hop). However, such a communication is only possible if there is a common channel among the sets $F(u)$ and $F(v)$. We assume that, the channel to radio assignment $F(u)$ for each node u is fixed as long as the traffic demands do not change. In other words, we assume there is no system or hardware support to allow a radio to switch channels on a per-packet basis. We denote by $c(e)$ the rate for edge $e = (u,v)$. This is the maximum rate at which mesh router u can communicate with mesh router v in one hop on a single channel. An edge can support multiple simultaneous communications of rate $c(e)$ each one for every channel in common among the sets $F(u)$ and $F(v)$. Each such communication can be uniquely identified with one channel in common among the sets $F(u)$ and $F(v)$. For notational convenience we will use $F(e)$ to denote the common channels among $F(u)$ and $F(v)$. Thus, for an edge $e = (u,v)$, k simultaneous link transmissions each of rate $c(e)$ are possible from node u to node v if there are k channels in $F(e)$.

We denote by R_I the *interference range*. We assume that R_I is $q \times R_T$ where $q \geq 1$. We assume that 802.11 media access control protocol, CSMA with RTS/CTS/ACK is used to protect unicast transmissions. Thus, as a result of carrier sensing, a transmission between u and v may block all transmissions within R_I away from either u (due to sensing RTS and DATA) or v (due to sensing CTS and ACK). In particular, simultaneous link transmissions on a common channel f on two distinct edges $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$ is possible if and only if none of the four pairs of nodes $(u_1, u_2), (v_1, u_2), (u_1, v_2), (v_1, v_2)$ are at most R_I apart. *In this case we say that edges e_1 and e_2 do not interfere. Otherwise the two edges interfere with each other.* We denote by $I(e) \subseteq E$ the set of edges that interfere with an edge $e \in E$. Note that simultaneous link transmissions on two edges e_1 and e_2 that interfere is still possible, as long as these transmissions are on distinct channels. We would like to note that our results also extend to other commonly used interference models including the Protocol Model [11] which are based on certain geometric properties.

2.3 Assumptions on Scheduling, Routing and Fairness

For the algorithm presented in this paper, we assume that the system operates synchronously in a time slotted mode. The throughput we obtain will provide an upper bound for systems using 802.11 MAC.

We assume traffic between a node and the gateway nodes is routed on multiple paths to achieve the optimal load balancing and least congestion for the given WMN. Although, such a scheme results in a node's traffic to and from the wired network being split over multiple paths, the end users traffic may still be routed as a whole to the extent possible by performing the traffic split across rather than within user flows. This may be easily implemented by maintaining user flow information at the end nodes to which the mobile clients connect to. Note that load balanced routing is akin to Traffic Engineering in MPLS networks and results in better network performance.

Our goal is to maximize the capacity of the network for serving mesh clients. This capacity may not be measured by the total throughput of all mesh clients. Optimizing such a metric may lead to starvation of mesh clients which are far from gateways. We therefore need to consider fairness constraints to prevent such starvation. We consider the fairness constraint that, for each node $u \in V$, demands be routed in proportion to its aggregate user traffic load $l(u)$. Note that, the nodes in V correspond to wireless routers. Our solution works with no change if V also includes mesh clients.

3. PROBLEM FORMULATION AND ALGORITHM OVERVIEW

Formally, we are given a wireless mesh backbone network modeled as a graph (V, E). We have a total of K channels. Each node $u \in V$ has $I(u)$ network interface cards, and has an aggregated demand $l(u)$ from its associated users. For each edge e the set $I(e) \subset E$ denotes the set of edges that it interferes with. We seek to maximize λ where at least $\lambda l(u)$ amount of throughput can be routed from each node u to the Internet (represented by a node t). In order to achieve $\lambda l(u)$ throughput for each node u, we need to compute (1) a network flow that associates with each edge $e = (u,v)$ values $f(e(i)), 1 \leq i \leq K$ where $f(e(i))$ is the rate at which traffic is transmitted by node u for node v on channel i; (2) a feasible channel assignment $F(u)$ (recall that $F(u)$ is an ordered set where the $i-th$ interface of u operates on the $i-th$ channel in $F(u)$) such that, whenever $f(e(i)) > 0$, $i \in F(u) \cap F(v)$; In this case we say edge e uses channel i (3) a *feasible* schedule S that decides the set of edge channel pair (e,i) (edge e using channel i) scheduled at time slot τ, for $\tau = 1, 2, \ldots, T$ where T is the period of the schedule. A schedule is feasible if the edges of no two edge pairs $(e_1, i), (e_2, i)$ scheduled in the same time slot for a common channel i interfere with each other ($e_1 \notin I(e_2)$ and $e_2 \notin I(e_1)$).

Even the interference free edge scheduling sub-problem given the edge flows is NP-hard. We present an approximation algorithm for the overall problem. We refer to this algorithm as the the joint routing, channel assignment and link scheduling (RCL) algorithm. We now give an overview of the RCL algorithm. The algorithm performs the following five steps in the given order:

1. **Solve LP:** Since the optimal problem with mixed linear and integer program formulation is NP-hard, we first solve the LP relaxation of the problem optimally. This results in a flow on the flow graph along with a not necessarily feasible channel assignment for the node radios. Specifically, a node may be assigned more channels than the number of its radios. However, this channel assignment is "optimal" in terms of ensuring that the interference for each channel is minimum. This step also yields a lower bound on the λ value which we use in establishing the worst case performance guarantee of the overall algorithm.

2. **Channel Assignment:** In this step, we present a channel assignment algorithm which is used to adjust the flow on the flow graph (routing changes) to ensure a feasible channel assignment. This flow adjustment also strives to keep the increase in interference for each channel to a minimum.

3. **Post Processing:** In this step the flow on the flow graph is re-adjusted (routing changes) to ensure that the maximum interference over all channels is minimized. This step does not change the channel assignment.

4. **Flow Scaling:** In this step the flow on the flow graph is scaled to ensure that all interference for all channels is eliminated. It results in a feasible routing and channel assignment which satisfies a sufficient condition for an interference free edge communication schedule.

5. **Interference Free Link Scheduling:** In this step for the edge flows corresponding to the flow on the flow graph, we obtain an interference free link schedule.

In subsequent sections, we will first describe each step in detail. We then present the analysis on the algorithm and show that it achieves a constant factor approximation. We use the example network shown in Figure 1 to illustrate the steps of our algorithm. For ease of explanation, we begin with interference free edge communication scheduling.

4. INTERFERENCE FREE LINK SCHEDULING: ALGORITHMS AND NECESSARY CONDITION

In this section we study the following question. Given a channel assignment to the wireless node radios (interfaces) and a traffic routing that is able to route $l(v)$ outgoing traffic load for each node $v \in V$ what are the necessary and sufficient conditions under which the edge communications of the routing solution can be scheduled free of interference. Our main results are a necessary condition which is "almost" also sufficient: it can be made sufficient by reducing the amount of routed traffic by a small constant factor. In addition we design an algorithm that finds such an interference free link communication scheduling, whenever the sufficient condition is satisfied by the channel assignment and traffic routing solution. Our results are obtained by extending those of [16] for the single channel case and for the Protocol Model of interference [11]. We first study the interference free link scheduling problem independently for a single channel, say channel 1. We will then show later how these solutions obtained independently can be easily "merged" to obtain an interference free link schedule for all channels.

4.1 Interference free Link Flow Scheduling for a single channel

We assume a periodic (with period T) time slotted schedule S in which in each slot the links scheduled for transmission do not interfere with each other. Let $X_{e,i,\tau}, e \in E, i \in F(e), \tau \geq 1$ be the indicator variable where $X_{e,i,\tau}$ is 1 if and only if link e is active in slot τ on channel i. Recall that $F(e)$ is the set of channels in common among the set of channels assigned to the end-nodes of edge e. Thus the fraction of time link e is active in schedule S on channel i is $\frac{1}{T}\sum_{1\leq\tau\leq T} X_{e,i,\tau}$. We denote this fraction by $\alpha(e,i)$ the *fractional link utilization* for channel i. Since we are focusing on one channel (channel 1), we will for ease of exposition in this section denote this fraction by $\alpha(e)$ and the indicator variable as $X_{e,\tau}$.

Now we make the connection with flows on the edges of G and the edge communication schedules. Note that we must have $f(e(1)) = \alpha(e)c(e)$. Recall that $f(e(i))$ is the rate (denoted by flow here on) at which traffic is transmitted by node u for node v on channel i for an edge $e = (u,v) \in E$. Since edge e is active $\alpha(e)$ fraction of time on channel 1 and its rate (when active) is $c(e)$ the equality follows. This connection between the edge flows and the edge schedules is useful in deriving some necessary conditions that the edge flows in any feasible joint channel assignment and routing solution must satisfy to be feasible under interference constraints. The following arguments for our interference model are based on geometric considerations.

4.1.1 Link Flow Scheduling: Necessary and Sufficient Conditions

For an edge e we denote by $I(e)$ the set of edges that interfere with it as defined in Section 2.2. Recall that this interference is defined by an interference range R_I which is assumed to be q times a transmission range R_T for some small fixed value q. Based on geometric arguments we can show the following.

LEMMA 1. *For any slot τ any valid interference free edge communication schedule S must satisfy*

$$X_{e,\tau} + \sum_{e' \in I(e)} X_{e',\tau} \leq c(q),$$

where $c(q)$ is a constant that depends only on q. For example $c(q) = 4, 8, 12$ for $q = 1, 2, 2.5$ respectively [15].

PROOF. We prove the following for $q = 2$ and proofs for other values of q can be derived along similar lines. Recall that an edge $e' \in I(e)$ if there exist two nodes $x, y \in V$ which are at most $2R_T$ apart and such that edge e is incident on node x and edge e' is incident on node y. Let $e = (u,v)$. Consider the region C formed by the union of two circles C_u

and C_v of radius $2R_T$ each, centered at node u and node v respectively. Then $e' = (u', v') \in I(e)$ if an only if at least one of the two nodes u', v' is in C; Denote such a node by $C(e')$.

Given two edges $e_1, e_2 \in I(e)$ that do not interfere with each other we must have that the nodes $C(e_1)$ and $C(e_2)$ are at least $2R_T$ apart. Thus an upper bound on how many edges in $I(e)$ do not pair-wise interfere with each other can be obtained by computing how may nodes can be put in C that are pair-wise at least $2R_T$ apart. For an even looser upper bound we can extend C to a circle C_e of radius $3.5R_T$ which is centered in the middle of the line joining the end-points of edge e and re-formulate the above question as a circle packing problem: how many maximum circles of radius R_T can be packed (without overlap) in the circle C_e of radius $3.5R_T$? From [15] it follows that this number is 8. Thus among the edges in $I(e)$ every "independent" set is of size at most 8. Thus in schedule S in a given slot only one of the two possibilities exist: either edge e is scheduled or an "independent" set of edges in $I(e)$ of size at most 8 is scheduled implying the claimed bound. □

A necessary condition: From Lemma 1 it follows that a necessary condition (*Link Congestion Constraint*) for the edge flows (for each channel) is:

LEMMA 2. *Any valid "interference free" edge flows must satisfy for channel* 1 *(and equivalently for other channels) the Link Congestion Constraint:*

$$\frac{f(e(1))}{c(e)} + \sum_{e' \in I(e)} \frac{f(e'(1))}{c(e')} \leq c(q). \quad (1)$$

PROOF. Recall that we are only considering channel 1. Adding up the in-equality in Lemma 1 for the first T time slots and dividing the resulting by T we obtain:

$$\frac{1}{T} \sum_{1 \leq \tau \leq T} X_{e,\tau} + \sum_{e' \in I(e)} \frac{1}{T} \sum_{1 \leq \tau \leq T} X_{e',\tau} \leq c(q).$$

This is equivalent to

$$\alpha(e) + \sum_{e' \in I(e)} \alpha(e') \leq c(q)$$

and the result follows from the definition of $\alpha(e)$. □

Next we formulate a matching sufficient condition for an interference free edge communication schedule.

A sufficient condition: A sufficient condition (*Link Congestion Constraint*) for the edge flows (for each channel) is:

LEMMA 3. *If the edge flows satisfy for channel* 1 *(and equivalently for other channels) the following* Link Schedulability Constraint *than an interference free edge communication schedule can be found.*

$$\frac{f(e(1))}{c(e)} + \sum_{e' \in I(e)} \frac{f(e'(1))}{c(e')} \leq 1. \quad (2)$$

The proof of this Lemma is established by demonstrating an algorithm which can find an interference free edge communication schedule and is presented next.

4.1.2 Link Flow Scheduling: An Algorithm

Now we present an algorithm that given the edge flows can find an interference free schedule of the edges for channel 1 (equivalently for any given channel). We present a centralized version of the algorithm. We would also like to note that it is also possible to design a *distributed* version of this algorithm along similar lines as in [16]. Given the edge flows for channel 1 the algorithm finds an interference free periodic schedule S of period T (for some large number T) for communication on edges for channel 1 (equivalently for any given channel) such that the schedule satisfies the following two requirements:

- The fraction of time slots in which edge e is scheduled is given by
$$\frac{1}{T} \sum_{1 \leq \tau \leq T} X_{e,\tau} = \frac{f(e(1))}{c(e)}.$$
Recall that $f(e(1))$ is the flow on channel 1 for edge e.
- Two edges that interfere with each other are not assigned to the same slot.

Note that these conditions ensure that edge communication schedule S is interference free and achieves the edge flows for channel 1 (equivalently for any given channel).

Algorithm 1 presents the pseudo code for the scheduling algorithm. Here we assume T is chosen to be a large number. The algorithm schedules each edge e in $N(e) = T\frac{f(e(1))}{c(e)}$ slots. We assume edges in E are ordered as $e_1, e_2 \ldots e_m$. Since S is periodic, the algorithm only outputs schedule for first T slots. We use $S(e)$ to denote the set of time slots in which edge e is scheduled in S. Note that by construction Algorithm 1 outputs an interference free schedule.

We omit the proof of the following Lemma regarding Sufficient Condition for Schedulability:

LEMMA 4. *If the edge flows for channel* 1 *(equivalently for any given channel) satisfies the Link Schedulability Constraint (2) then an interference free schedule for the edges for channel* 1 *(equivalently for any given channel) can be found by the Algorithm 1.*

ALGORITHM 1. *LINK SCHEDULING—single channel case*

Set Available slots to $1, 2 \ldots T$.
Initialize $S(e_i) = \emptyset$, $\forall i = 1, \ldots, m$
for $i = 1, \ldots, m$ {
  Set $S(e_i)$ *to first available* $N(e_i)$ *slots such that*
    $S(e_i) \cap \cup_{e_j \in I(e_i)} S(e_j) = \emptyset$.
}

4.2 Interference free Link Flow Scheduling for all channels

In this section we provide necessary and sufficient conditions under which the edge communications for more than one channels can be scheduled free of interference for a given assignment to the wireless node radios and a traffic routing that is able to route $l(v)$ outgoing traffic load for each node $v \in V$. We also design an algorithm for finding an interference free schedule for the flow on the edges for all channels. We use Algorithm 1 for this purpose.

Necessary Condition: The edge flows for any given channel must satisfy the Link Congestion Constraint (1).
Sufficient Condition: The edge flows for any given channel must satisfy the Link Schedulability Constraint (2).

If the solution satisfies the Sufficient Condition then the following "merging" algorithm can be used for finding an interference free edge communication schedule. Here T is picked to be a large number. We denote by S_i the schedule for edge flows for channel i obtained using Algorithm 1. We denote by $S_i(\tau)$ the set of edges scheduled at time slot τ by S_i. We denote by $S(\tau)$ the set of tuples denoting an edge and a channel pair corresponding to the edge in the tuple being scheduled on the channel in the tuple at time slot τ by S.

ALGORITHM 2. *LINK SCHEDULING–multiple channel case*

---

for $\tau = 1, \ldots, T$ {
$S(\tau) = \cup_{1 \le i \le K} \cup_{e \in S_i(\tau)} (e, i)$.
}

The interference free edge communication scheduling problem, given the set of edge flows is as hard as edge coloring even for very simple interference models and hence is NP-hard in general. We can only hope for an efficient approximation algorithm for this problem. We can establish the following regarding Algorithm 2:

THEOREM 1. *Algorithm 2 can be used to design a $c(q)$-approximation algorithm for finding interference free edge communication schedule, where $c(q)$ is a constant defined in Lemma 1.*

PROOF. Note that given an edge flow it must satisfy the necessary condition (Link Congestion Constraint (1) for the edge flow for each channel) or otherwise it is not schedulable. Thus by scaling the given edge flows by a factor $c(q)$ the edge flows must also satisfy the Link Schedulability Constraint based sufficient condition (2) for the flow on each channel. Algorithm 2 is then able to find an interference free edge communication schedule for these edge flows. Note that scaling edge flows by a factor of at most $c(q)$ has the effect that in the corresponding routing at least $1/c(q)$ fraction of the $l(v)$ outgoing traffic load for each node $v \in V$ is routed and the approximation bound follows. □

Thus Algorithm 2 is able to compute an interference free link schedule if the routing and channel assignment is known. How to do the latter efficiently is what we focus on in the rest of the paper.

5. ROUTING

The section is organized as follows. We first present a mathematical model of our problem in terms of a "flow graph" for the given number of channels K and the underlying wireless network G. Next we present a Linear Programming based approach for computing the best routing which is optimal in maximizing the system throughput subject to fairness constraints. The given routing is also efficient: satisfies the necessary condition (*Link Congestion Constraint*) that any valid "interference free" edge flows must satisfy.

As mentioned in previous sections our goal is to find a solution that achieves optimal throughput subject to fairness constraints. For ease of exposition we will assume a simple fairness constraint in which the minimum load routed for the nodes in V is maximized. Thus we are interested in finding a solution that can maximize λ while routing $\lambda l(v)$ aggregate load for each node. Our results easily extend to the more general case: non-uniform node aggregate load and other fairness constraints. For instance to distribute network capacity that is left over after satisfying $\lambda l(v)$ aggregate load our algorithm can be iteratively invoked, thus resulting in efficient utilization of the available resources.

5.1 Network Flows and Flow Graphs for Multiple Channels

The traffic routed between the nodes of the WMN and the gateway nodes can be mathematically described in terms of networks flows. For ease of exposition in describing this connection we start out by assuming a single channel and a single radio at each wireless node. Given a traffic routing for the WMN G that is able to satisfy the loads $l(v)$ at all nodes $v \in V$, one can associate with each edge $e = (u, v)$ a value $f(e)$ which is the rate at which traffic is transmitted by node u for node v on the only available channel (for the edges between the gateway nodes and the wired network this traffic goes over wired links). These $f(e)$ values form the edge flows for a *max-flow* on a *flow graph* H constructed as follows: We start out with the graph G and introduce a source node s and sink node t. Every gateway node $v \in V$ has an unlimited capacity directed edge to a designated sink node $t \in V$ representing the wired Internet. Node s has a directed edge to every node $v \in V$ of capacity $l(v)$ where $l(v)$ is the outgoing traffic load at node v. We denote these edges by the set E_s. The resulting graph is the required flow graph with source node s and sink node $t \in V$. The flow on an edge $e \in E$ in this flow graph is $f(e)$. Any edge $e \in E_s$ in this flow graph has a flow equal to its capacity. Since the traffic routing in G that resulted in the flow values $f(e)$ for the edges of G is able to satisfy the loads $l(v)$ at all nodes $v \in V$, the above defined flow must satisfy flow conservation at every node of G and hence is a valid network flow. Note also that the total flow outgoing from the source node s is equal to the sum of the capacities of the edges outgoing from s. Thus, the resulting flow is a max-flow on the flow-graph for the given source and sink nodes.

We now describe the general case when there are multiple channels and multiple radios. We assume there are K orthogonal channels in $\mathcal{F}$, which are numbered $1, 2 \ldots K$. Given a traffic routing and channel assignment for the WMN G that is able to satisfy the loads $l(v)$ at all nodes $v \in V$, one can associate with each wireless edge $e = (u, v)$ and channel i a value $f^i(e)$ which is the rate at which traffic is transmitted by node u for node v on channel i. Note that $f^i(e) = 0$ if $F(e)$ does not contain channel i. Note also that when there are multiple channels a packet may be received by a node u on a channel i but may be transmitted by u to a next hop neighbor on a different channel j. Thus for a wireless node $u \in V$ we define $f^{(i,j)}(u)$ as the rate at which data arriving on channel i at node u leaves it on channel j. Note that $f^{(i,j)}(u) = 0$ for all channels i and j that are not in $F(u)$.

For each channel $i, 1 \le i \le K$ we create a copy of the flow graph for the single channel case (described earlier), except the nodes s and t, and denote it by $H(i)$. For each node $v \in V$ and for each edge $e \in E$ we denote the corresponding node and edge in $H(i)$ by $v(i)$ and $e(i)$ respectively. We

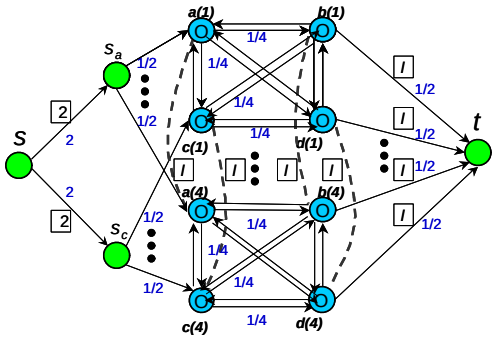


Figure 2: A flow graph for the wireless network in Figure 1. The boxed edge labels represent edge capacities. All wireless edges have capacity 1 (not shown). The label I stands for infinite capacity. Non-boxed edge labels are edge flow values. The dashed vertical edges (curved) connect corresponding wireless nodes (labeled with an inner circle) in different copies. The traffic load for each non-gateway node v is $l(v) = 2$ and it is 0 for gateway nodes.

stitch these K copies together to create an overall flow graph H as follows. We introduce $K(K-1)/2$ unlimited capacity edges E_v to connect copies of each node $v \in V$:

$$E_v = \{(v(i), v(j)), 1 \leq i, j \leq K-1, i \neq j, v \in V\}.$$

These edges enable moving flow from one channel to another within a node. We create a super source node s and super sink node t. For every node $v \in V$ a dummy source node s_v is introduced along with the edges $E_s = (s, s_v), \{(s_v, v(i)), 1 \leq i \leq K\}$ each of capacity $l(v)$. Intuitively the dummy source node s_v can be thought of doing the splitting of the outgoing traffic for node v for transmission on different channels. We denote the edges terminating on the sink node t by E_t. The resulting graph is the required flow graph with source node s and sink node t. In this flow graph the flow $f(e(i))$ on an edge $e(i), e \in E$ is set to $f^i(e)$, the flow $f((s, s_v))$ on edges $(s, s_v), v \in V$ is set to $l(v)$ and the flow $f((v(i), v(j)))$ on edges in

$$E_v = (v(i), v(j)), 1 \leq i, j \leq K-1, i \neq j, v \in V$$

is set to $f^{(i,j)}(v)$. The flow $f((s_v, v(i)))$ on edges $(s_v, v(i)), v \in V, 1 \leq i \leq K$ is set to the difference of the total outgoing traffic on channel i on node v and the total incoming traffic on channel i on node v. Likewise for the gateway nodes $v \in V_G$ the flow $f((v(i), t))$ to the sink node for each channel i is set to the total incoming traffic on channel i on node v that gets sent to the wired gateway by node v. By an argument similar to the one used for the single channel case it can be shown that the resulting flow forms a max flow between the source node s and the sink node t over the flow graph. Figure 2 shows the flow graph and the various flows for the mesh network depicted in Figure 1.

Note that in general there may not exist a traffic routing for the WMN G that is able to satisfy the loads $l(v)$ at all nodes $v \in V$. In this case we assume that the flow graph (and the flow there on) is constructed for the traffic routing that satisfies the loads $\lambda l(v)$ at all nodes $v \in V$ for the largest possible value of λ. We will show later that such a traffic routing can also be efficiently found.

5.2 A Linear Programming based Routing Algorithm

In this section we show how to efficiently find such a routing that also satisfies all the necessary constraints for the joint channel assignment, routing and interference free link scheduling problem. We start out by describing these constraints first.

5.2.1 *Routing and Channel Assignment Constraints: Necessary Conditions*

As described in Section 5.1 the wireless traffic routed between the nodes of the WMN and the gateway nodes for a given channel assignment can be mathematically described in terms of networks flows. Specifically, this connection can be made in terms of a flow graph H and the flow routed on it. We now define a set of constraints that must be satisfied by the flow on H. Our goal is then to find an optimal flow on H that satisfies these constraints and can also be used to derive an efficient routing and channel assignment solution for our problem. We denote the set of nodes (edges) in H by V^H (E^H). We denote by $f(e)$ ($c(e)$) the flow (capacity) on an edge e of H. For more information about the flow graph H and the notation used in its description the reader is referred to Section 5.1.

Flow constraints: At any node $v \in V^H$ which is not the source node s or the sink node t the total incoming flow must equal the total outgoing flow:

$$\sum_{(u,v) \in E^H} f((u,v)) = \sum_{(v,u) \in E^H} f((v,u)), \ \forall v \in V^H - \{s,t\} \quad (3)$$

We would like the source node to send out the maximum possible amount of flow. Specifically the source node must send out the maximum flow on each of its outgoing edge limited only by the edge capacity. Note that this corresponds to routing the $l(v)$ aggregate traffic load at each node $v \in V$ of the WMN G. However this may not always be possible. In this case we require a flow on each outgoing edge which is λ times its capacity (or $\lambda l(v)$ aggregate traffic load at each node $v \in V$) for the largest possible λ. Note that this way we can ensure fairness in the allocation of resources among wireless nodes. This results in one constraint and one objective function ($\max \lambda$):

$$f((s, s_v)) = \lambda l(v), \ \forall v \in V \quad (4)$$

We must ensure that no capacities are violated:

$$f(e) \leq c(e), \ \forall e \in E^H \quad (5)$$

Node Radio Constraints: Recall that a WMN node $v \in V$ has $I(v)$ radios and hence can be assigned at most $I(v)$ channels from $1 \leq i \leq K$. One way to model this constraint is to observe that due to interference constraints v can be involved in at most $I(v)$ simultaneous communications (with different one hop neighbors). Thus in any interference free schedule, for any node $v \in V$, at any time slot τ it must be the case that

$$\sum_{1 \leq i \leq K} \sum_{e=(u,v) \in E} X_{e,i,\tau} + \sum_{1 \leq i \leq K} \sum_{e=(v,u) \in E} X_{e,i,\tau} \leq I(v).$$

Note that this is not a very tight constraint for modeling the number of radios of a node. However, we will show later

that this results in a relaxation which is not too far from the optimal. Adding these sets of equations for all time slots T and dividing by T this results in the constraint:

$$\sum_{1\le i\le K}\sum_{e=(u,v)\in E}\alpha(e,i)+\sum_{1\le i\le K}\sum_{e=(v,u)\in E}\alpha(e,i)\le I(v).$$

Recall that the fraction $\alpha(e,i)$ denotes the fractional link utilization for link e for channel i. In other words $\alpha(e,i)=\frac{f(e(i))}{c(e)}$ for any $e\in E$ and channel i. This therefore results in the following linear constraints for each node $v\in V$.

$$\sum_{1\le i\le K}\sum_{e=(u,v)\in E}\frac{f(e(i))}{c(e)}+\sum_{1\le i\le K}\sum_{e=(v,u)\in E}\frac{f(e(i))}{c(e)}\le I(v) \quad (6)$$

Link Congestion Constraints: We restate the Link Congestion Constraint (1) in terms of link flows:

$$\frac{f(e(i))}{c(e)}+\sum_{e'\in I(e)}\frac{f(e'(i))}{c(e')}\le c(q),\ \forall e\in E,\ 1\le i\le K \quad (7)$$

Objective Function: As stated before the objective function is

$$\max\lambda \quad (8)$$

Note that all the constraints listed above are necessary conditions for any feasible solution. However, these constraints are not necessarily sufficient. Hence if a solution is found that satisfies these constraints it may not be a feasible solution. Our approach is to start with a "good" but not necessarily feasible solution that satisfies all of these constraints and use it to construct a feasible solution without impacting the quality of the solution.

5.2.2 Traffic Routing Algorithm

We now formulate a Linear Program (LP) to find a flow that maximizes λ subject to the *Flow Constraints* (3), (5) and (4), *Node Radio Constraints* (6) and *Link Congestion Constraints* (7). The resulting LP (LP1) is given below:

$$\max\lambda \quad (9)$$

Subject to

$$\sum_{(u,v)\in E^H}f((u,v))=\sum_{(v,u)\in E^H}f((v,u)),\ \forall v\in V^H-\{s,t\}$$
$$f((s,s_v))=\lambda l(v),\forall v\in V$$
$$f(e)\le c(e),\ \forall e\in E^H$$
$$\sum_{1\le i\le K}\left(\sum_{e=(u,v)\in E}\frac{f(e(i))}{c(e)}+\sum_{e=(v,u)\in E}\frac{f(e(i))}{c(e)}\right)\le I(v),\ v\in V$$
$$\frac{f(e(i))}{c(e)}+\sum_{e'\in I(e)}\frac{f(e'(i))}{c(e')}\le c(q),\forall e\in E,\ 1\le i\le K$$

The optimal solution to this LP is a flow on the flow graph H that maximizes λ and satisfies all of the above mentioned constraints. Although the solution yields the best possible λ (say λ^*) from a practical point of view some more improvements may be possible:

- The flow may have directed cycles. This may be the case since the LP does not try to minimize the amount of interference directly. By removing the flow on the directed cycle (equal amount off each edge) flow conservation is maintained and in addition since there are fewer transmissions the amount of interference is reduced.
- Flow may be going on long paths. Note that longer paths imply more link transmissions. In this case many times by moving the flow to shorter paths, system interference may be reduced. This is especially the case for instance on flows going on 2 hop paths. Consider a flow on a 2 hop path from node u to node w when there is a direct edge from u to w. Assuming all three edges have equal capacity, one can move the maximum possible flow (limited by capacity constraints and the flow on the edges) off the 2 hop path to the direct edge while maintaining flow conservation thus reducing total interference. This transformation is shown in Figure 3. The scenario outlined here regarding 2 hop paths that can be bypassed by a single edge is typical for wireless network due to their underlying geometrical structure. For example all nodes in a circle of radius $R_T/2$ are connected to each other thus forming a clique and any 2 hop path involving these nodes is bypassed by a direct edge.

The above arguments suggests that it would be practical to find among all solutions that attain the optimal λ value of λ^* the one for which the total value of the following quantity (which by the Link Schedulability Constraint (2) is an intuitive measure of total interference) is minimized:

$$\sum_{1\le i\le K}\sum_{e=(v,u)\in E}\frac{f(e(i))}{c(e)}.$$

We thus re-solve the LP with this objective function and with λ fixed at λ^*:

$$\min\sum_{1\le i\le K}\sum_{e=(v,u)\in E}\frac{f(e(i))}{c(e)} \quad (10)$$

Subject to

$$\sum_{(u,v)\in E^H}f((u,v))=\sum_{(v,u)\in E^H}f((v,u)),\ \forall v\in V^H-\{s,t\}$$
$$f((s,s_v))=\lambda^* l(v),\forall v\in V$$
$$f(e)\le c(e),\ \forall e\in E^H$$
$$\sum_{1\le i\le K}\left(\sum_{e=(u,v)\in E}\frac{f(e(i))}{c(e)}+\sum_{e=(v,u)\in E}\frac{f(e(i))}{c(e)}\right)\le I(v),\ v\in V$$
$$\frac{f(e(i))}{c(e)}+\sum_{e'\in I(e)}\frac{f(e'(i))}{c(e')}\le c(q),\forall e\in E,\ 1\le i\le K$$

The optimal solution to this LP (LP2) is a flow on the flow graph H that maximizes λ, satisfies all of the above mentioned constraints and also tries to minimize the maximum interference per channel.

We now illustrate the routing step using the 4-node example network shown in Figure 1. Suppose there are a total of 4 channels and each node has 2 radio interfaces. Suppose the edge capacities are 1 unit each and $q=2$ ($C_q=8$). We assume that the interference range is large enough so that only one node can transmit at any given time on a given channel. We assume both nodes a and c have a demand of 2 units each. Let us denote $e_1=(a,b)$, $e_2=(a,d)$, $e_3=(c,b)$, $e_4=(c,d)$. After completing the routing step, we may have the following set of edge flows as an optimal solution to the LP-based routing algorithm: $f(e_j(i))=1/4$, $\forall i,j=1,2,3,4$. It is easy to see that this edge flows satisfies all the linear constraints in LP1 and is optimal. However,

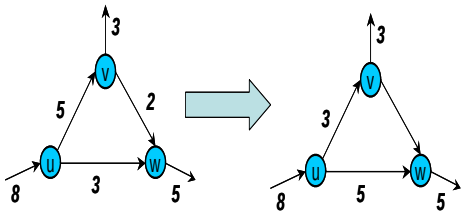


Figure 3: Example of a flow transformation (edge label denote its flow) where the maximum possible flow (2 units) is moved from a 2 hop path (u, v, w) to a direct edge bypass (edge (u, w)). Note that this does not violate flow conservation at any node. Also interference resulting from node v's communications are reduced. Since node u and w have the same amount of traffic communication to do before and after no new interference is introduced.

this flow is not feasible to the original throughput optimization problem since the channel assignment is not feasible (2 radios utilize 4 channels).

6. A CHANNEL ASSIGNMENT ALGORITHM

In this section we present a channel assignment algorithm that operates on a flow on the flow graph H for a traffic routing that satisfies the loads $\lambda^* l(v)$ at all nodes $v \in V$. Although this given flow satisfies the Link Congestion Constraints (1) for all the channels, the *induced* channel assignment may not be feasible. The channel assignment algorithm transforms the given flow to fix this infeasibility. In other words it ensures that for each v, the number of channel i such that $f(e(i)) > 0$, $\forall e \in E$ and e incident on v, is no more than $I(v)$. When we scale the resulting flow of the channel assignment step, it is at least a $\phi = \frac{c(q)K}{I}$ fraction of the original flow and hence satisfies at least the load $l(v)\lambda/\phi$ at all nodes $v \in V$. The scaled flow also satisfies the Link Schedulability Constraint (2) for all channels thus implying an interference free schedule for each channel can be obtained.

The algorithm works in phases and on termination ensures that the number of channels assigned to any node v is at most the number of its interfaces $I(v)$. We will assume without loss of generality that every edge $e \in E$ has positive flow on at least one edge $e(i)$ $(f(e(i)) > 0)$ for some channel i in the flow graph H. This is because if there is one such edge $e \in E$ with zero flow on all the edges $e(i)$ then the algorithm would not route any flow on any of the edges $e(i)$ and hence edge e and all the edges $e(i)$ may be removed from the network. In particular the basic operation that the algorithm does is to move some flow from edge $e(i)$ to edge $e(j)$ in the flow graph H for some link $e = (u, v) \in E$. Here i and j are two distinct channels. Note that in order to maintain flow conservation, this operation may also involve adjusting flows on the infinite capacity links between nodes $u(i)$ and $u(j)$, and between nodes $v(i)$ and $v(j)$ on the flow graph H. It is easily seen that this adjustment to maintain flow conservation is always possible, and hence we will not explicitly mention this adjustment when talking about the operation. In this flow adjustment step we disregard edge capacities. Thus it is possible that the flow adjustment may lead to edge capacity violations. However, we will show later (when analyzing the performance of the algorithm in Section 8) that these capacity violations can be easily dealt with by a flow scaling.

Besides ensuring that the number of channels assigned to each node v is at most the number of its interfaces $I(v)$ the algorithm also strives to spread the interference evenly among the different channels, thus ensuring that for each channel the amount of interference is minimized. We use the following expression for measuring the interference on a link $e \in E$ for a given channel i:

$$Int(e, i) = \frac{f(e(i))}{c(e)} + \sum_{e' \in I(e)} \frac{f(e'(i))}{c(e')} \tag{11}$$

The choice of this expression for measuring link interference follows from the observation that the Link Schedulability Constraint (2) which is based on this expression is sufficient for finding an interference free schedule for the channel. Based on this expression we define the interference on a channel i as

$$Int(i) = max_{e \in E} Int(e, i) \tag{12}$$

We now describe the channel assignment algorithm. We say a node v is assigned to channel i if there exists an edge e incident on node v for which $f(e(i)) > 0$. Let $I \geq 1$ denote the minimum number of radios at each wireless node. Let $\mu(v)$ denote the "aggregate fractional flow" on an edge e:

$$\mu(e) = \sum_{1 \leq i \leq K} \frac{f(e(i))}{c(e)}.$$

Let $\mu(v)$ denote the total "aggregate fractional flow" on the edges incident on node v:

$$\mu(v) = \sum_{e=(u,v) \in E} \mu(e) + \sum_{e=(v,u) \in E} \mu(e).$$

The algorithm operates in three phases.

In the first phase the given network $G = (V, E)$ is transformed into a network $G' = (V', E')$ so that all nodes in G' have approximately I (between I and $2I - 1$) radios each. Given that the Node Radio Constraint for a node $v \in V$ is satisfied in the flow graph H from the routing step, this phase ensures that the total aggregate fractional flow on the edges incident on a node $v \in V'$ is at most $I(v)$.

The network G' is created as follows. Any node $v \in V$ with $Ir_1 + r_2$ radios, for $r_1 \geq 2$ and $I > r_2 \geq 0$, is replaced by r_1 nodes in G'. All these nodes except at most one has I radios and the one exceptional node has $I + r_2$ radios. Let these nodes (in V') be denoted by $v_1, v_2 \ldots v_{r_1}$. Next the edges incident on node $v \in V$ are distributed among these nodes so as to assign approximately the same fractional flow $\mu(v_i)$ to all nodes v_i. This is done while maintaining the constraint that $\mu(v_i) \leq I(v_i)$ for all v_i. In this step the algorithm iterates over the edges incident on node v. When considering an edge e, let v_i denote the node with minimum current value for $\mu(v_i)$ such that $\mu(v_i) < I(v_i)$. In case $I(v_i) - \mu(v_i) \geq \mu(e)$, then in G' edge e is made incident on node v_i. Otherwise a new copy e' of edge e is created. We set $\mu(e') = \mu(e) - (I(v_i) - \mu(v_i))$ and then set $\mu(e) = I(v_i) - \mu(v_i)$. Edge e is made incident on node v_i and e' is the next edge considered in the edge iteration by the algorithm. Pseudocode for Phase 1 is given in Algorithm 3.

ALGORITHM 3. *Phase 1 - Channel Assignment Algorithm*

Input: Network $G = (V, E)$ with Aggregate Fractional Flow Values $\mu(e), e \in E$ and $\mu(v), v \in V$
$V' = \emptyset$,$E' = E$
for $v \in V$ {
  Let $I(v) = Ir_1 + r_2, 0 \leq r_2 < I$
  $V' = V' \cup \{v_1, v_2, \ldots v_{r_1}\}$
  $I(v_1) = I + r_2$
  if $r_1 > 1$, $I(v_k) = I, 2 \leq k \leq r_1$
  $\mu(v_i) = 0, \forall 1 \leq i \leq r_1$
  Stack $Q = \{e \in E', e = (u, v)$ *or* $e = (v, u)\}$.
  While $Q \neq \emptyset$ {
    Pop Stack Q *to get edge* $e = (u, v)$
    Let v_i *has minimum* $\mu(v_i)$ *and* $\mu(v_i) < I(v_i)$
    $\mu = \min\{\mu(e), I(v_i) - \mu(v_i)\}$
    if $\mu < \mu(e)$ {
      Copy edge e *to* e'
      $\mu(e') = \mu(e) - \mu$
      $E' = E' \cup \{e'\}$
      Push e' *on top of Stack* Q
    }
    $\mu(v_i) = \mu(v_i) + \mu$
    $e = (u, v_i)$, $\mu(e) = \mu$
  }
}

Note that at the end of phase 1 of the algorithm the number of radios for each node is in the range I to $2I - 1$. In phase 2 of the algorithm the node radios are assigned channels between 1 and I. At the end of this phase it is possible that two radios on a node get assigned the same channel (since due to node splitting in phase 1 these two radios got assigned to different nodes in V'). Phase 3 of the algorithm addresses these issues. The goal of phase 2 is to assign channels to nodes in V in such a way that for any given channel i the network formed by the set of edges e with $f(e(i)) > 0$ has a large number of connected components with small intra-component interference. This is useful because then by assigning different channels to connected components with high inter-component interference the system interference is reduced. This is done in Phase 3 of the algorithm. The algorithm also ensures that at the end of Phase 2 the channel interference $Int(i)$ is at most $\frac{Kc(q)}{I}$, for all channels i.

In the following for ease of presentation we denote by $G = (V, E)$ also the network output by Phase 1 with aggregate fractional flow values $\mu(e), e \in E$ and $\mu(v), v \in V$, and number of radios $I(v), v \in V$. Recall that $Int(i)$ denotes the interference for channel i and is computed based on the edge interference $Int(e, i)$. For a given channel and flow assignment $f(e(i))$, let A be a connected component of the network formed by edges $e \in E$ with $f(e(i)) > 0$. We denote by $Int(e, i, A)$ the interference on edge $e \in A$ for channel i by only considering the edges in A. We can then define the interference $Int(i, A)$ for component A for channel i as the maximum value of $Int(e, i, A)$ over all edges e in A. Finally we can define the component interference $CompInt(i)$ for channel i as the maximum value of $Int(i, A)$ for all connected components A for the network formed by edges $e \in E$ with $f(e(i)) > 0$.

The algorithm starts out with an empty channel assignment: it sets $f(e(i)) = 0$ for all edges e and channels i. Thus, for each channel i, its interference $Int(i) = 0$ and also its component interference $CompInt(i) = 0$. The algorithm iterates over the nodes of the network in non-increasing order of the $\mu(v)$ value. When considering a node v, it iterates over the edges e incident on node v that have not been considered by the algorithm in the non-increasing order of $\mu(e)$ values. When considering an edge e, the algorithm operates as follows. It makes I copies of edge e: $e(1), e(2), \ldots e(I)$ and partitions the total edge e flow $\mu(e)c(e)$ among these I copies as follows.

For each channel $i, 1 \leq i \leq I$ the algorithm independently computes the maximum possible flow increase $\gamma(e(i)) \leq c(e)$ on edge $e(i)$ such that the resulting total flow on edges $e(k)$ for all channels $k, 1 \leq k \leq I$ is at most $max\{c(e), \mu(e)c(e)\}$ and such that for this new flow the channel i interference $Int(i)$ does not exceed $\frac{Kc(q)}{I}$. Let I' be the set of channels $i, 1 \leq i \leq I$ for which $\gamma(e(i)) > 0$. If $I' = \emptyset$ the algorithm proceeds to consider the next edge. Otherwise let γ be the minimum value of $\gamma(e(i))$ among channels $i \in I'$. Let $k \in I'$ be a channel for which an increase in the flow $f(e(k))$ by γ results in the minimum (among all the channels in I') component interference $\max_{i \in I'} CompInt(i)$. The algorithm increments $f(e(k))$ by $\gamma(e(k))$ for channel k. The above is repeated for edge e until (as mentioned above) $I' = \emptyset$. Pseudocode for Phase 2 is given in Algorithm 4.

ALGORITHM 4. *Phase 2 - Channel Assignment Algorithm*

Input: Network $G = (V, E)$ with Aggregate Fractional Flow Values $\mu(e), e \in E$ and $\mu(v), v \in V$
$f(e(i)) = 0, \forall e \in E$ *and* $1 \leq i \leq I$.
$C = \emptyset$ */* Set of edges considered so far */*
for $v \in V$ *in non-increasing order of* $\mu(v)$ {
  for $e \notin C$, e *incident on* v, *in non-increasing order of* $\mu(e)$ {
    Add e *to* C
    while(true) {
      For all $1 \leq i \leq I$ {
        Compute $\gamma(e(i)) \leq c(e)$, *the maximum possible flow increase on* $e(i)$ *such that* $\sum_{k=1}^{I} f(e(k)) + \gamma(e(i)) \leq \mu(e)c(e)$ *and* $Int(i) \leq \frac{Kc(q)}{I}$ *even with* $f(e(i))$ *increased by* $\gamma(e(i))$
      }
      Let $I' = \{i | \gamma(e(i)) > 0\}$
      If $I' = \emptyset$ *then break*
      Let $\gamma = \min_{i \in I'} \gamma(e(i))$
      For $j \in I'$, *let* $\nu(j) = max_{i \in I'} CompInt(i)$ *when* $f(e(j))$ *increased by* γ
      Let $\nu(k)$ *be minimum among* $\nu(j), j \in I'$
      $f(e(k)) = f(e(k)) + \gamma(e(k))$
    }
  }
}

Note that in the channel assignment obtained in Phase 2 nodes (each of which has at least I radios) are assigned at most I channels $(1, 2, \ldots, I)$ each. In Phase 3 of the algorithm the channel assignment is further modified such that each node is still assigned at most I channels. However, these channels may range anywhere from 1 to K now. This channel assignment is done with the goal of minimizing the overall intra-channel interference.

For the channel and flow assignment $f(e(i))$ that results from Phase 2 consider a connected component A of the network formed by edges $e \in E$ with $f(e(i)) > 0$ for some channel i. Note that all nodes in A are assigned channel i in this channel assignment. We say A is assigned channel i in this channel assignment. Consider re-assigning channel $k \neq i$ to A. This entails moving $f(e(i))$ flow from edge e's copy $e(i)$ to copy $e(k)$ for all edges e in A. Note that after

this transformation all edges e incident on node $u \in A$ have $f(e(i)) = 0$. Thus after the re-assignment no node $u \in A$ is assigned channel i anymore but is assigned channel k. Thus the number of distinct channels assigned to any node does not increase with this re-assignment.

If after Phase 2 of the algorithm there are at most K connected components A among all channels $(1, 2, \ldots, I)$ then in Phase 3 each of the connected components is assigned one of the K distinct channels. Otherwise the connected components within the channels are grouped to make K groups. This grouping is done as follows. Initially each connected component is in a group of its own. Analogous to the interference $Int(i, A)$ within a component A for channel i, we can compute $Int(i, P)$ the interference within a group P for channel i and we can define $GroupInt(i)$ as the maximum value of $Int(i, P)$ for all groups P in channel i. The algorithm greedily merges pair of groups belonging to the same channel to a single group such that the merging causes the least increase in $\max_{i=1}^{I} GroupInt(i)$ and until there are K groups. The connected components of the i-th group are then assigned channel i for $1 \leq i \leq K$.

In the last step of Phase 3 the channel and flow assignment is mapped back to the original network G and its flow graph H. Recall that in Phase 1 an edge e may have been split into multiple edges. Thus after the channel assignment in Phase 3 multiple copies (say $e_1, e_2, \ldots e_m$) of edge e may have positive flow in a given channel i. The flow on edge $e(i)$ is then set as the sum total $f(e(i)) = \sum_{k=1}^{m} f(e_k(i))$. Pseudocode for Phase 3 is given in Algorithm 5.

ALGORITHM 5. *Phase 3 - Channel Assignment Algorithm*

Input: Network $G' = (V', E')$ with $f(e(i))$ values for all channels $i \leq I$ and original network $G = (V, E)$
Let $\theta(i) = \{A | A$ *is a connected component assigned channel* $i\}$
while$(\sum_{i=1}^{I} |\theta(i)| > K)$ {
  Let $\eta_1, \eta_2 \in \theta(i)$ *such that* {
    Removing the groups η_1, η_2 *and adding the group* $\eta_1 \cup \eta_2$ *to channel* i *causes least increase in* $\max_{i=1}^{I} GroupInt(i)$
  }
  Remove η_1, η_2 *from* $\theta(i)$
  Add $\eta_1 \cup \eta_2$ *to* $\theta(i)$
}
$\theta = \cup\theta(i)$
Assign channel i *to the* i*-th group in* θ
For all $e \in E$ *and* $1 \leq i \leq K$ {
  Let $e_1, e_2, \ldots e_m \in E'$ *correspond to edge* e, $f(e(i)) = \sum_{k=1}^{m} f(e_k(i))$
}

Now let us continue our RCL algorithm on the 4-node example network from the routing step. Since all 4 nodes have the same number of radios, Phase 1 is not needed. Now let us consider Phase 2. From the routing step, we have $\mu(e_i) = 1, \forall i = 1, 2, 3, 4$ and $\mu(v) = 2, \forall v = a, b, c, d$. Since $\mu(e_i)$ and $\mu(v)$ are all the same, the algorithm picks nodes in the order a, b, c, d and edges in the order e_1, e_2, e_3, e_4. Note that for the edges e_1, e_2 that are incident on node a, $\gamma(e_1(i)) = \gamma(e_2(i)) = \mu(e_1) = \mu(e_2)$, for i=1,2 ($Int(i) = 1 < KC_q/I = 2C_q$). Therefore the algorithm sets $f(e_1(1)) = 1$. For the second iteration of the while loop for edges incident on a, e_2 causes the least intra-component interference if assigned channel 2 ($CompInt(1) = 2$, $CompInt(2) = 1$). Thus, the algorithm sets $f(e_2(2)) = 1$. Similarly, the algorithm sets $f(e_3(2)) = 1$ since e_3 would increase the component interference for channel 1 more than channel 2. Finally, we have $f(e_4(1)) = 1$. Note that there are 4 connected components in total after Phase 2: $\{e_1\}, \{e_4\}$ corresponding to channel 1 and $\{e_2\}, \{e_3\}$ corresponding to channel 2. Since $K = 4$, Phase 3 assigns each edge e_i a separate channel i. Thus the only non-zero edge flows are $f(e_1(1)) = f(e_2(2)) = f(e_3(3)) = f(e_4(4)) = 1$. This implies that nodes a, b, c, d are assigned channel pairs $(1, 2), (1, 3), (3, 4), (2, 4)$ respectively. Note that, this flow is already feasible. That is, the maximum interference is 1 and the flow scaling step in 7 has no effect.

7. POST PROCESSING AND FLOW SCALING

7.1 Post Processing

In this step of the algorithm the aim is to reduce the maximum interference suffered by the K channels, without effecting the feasibility of the channel assignment output by the previous step. This is done by re-distributing for each edge $e = (u, v) \in E$ the flows on its copies $e(i)$ for a channel i which is assigned to both nodes u and node v. Note that this includes all channels i for which $f(e(i)) > 0$. This re-distribution is done subject to the constraint that the total flow on the copies $\sum_{1 \leq i \leq K} f(e(i))$ does not change. Note that this re-distribution only results in an adjustment in the routing and it does not have any effect on the feasibility of the channel assignment. The latter is due to the fact that if an end-node u of edge e is not assigned channel i before this step then $f(e(i)) = 0$ after the step and hence node u is not assigned channel i after this step. Thus the number of channels assigned to a node can only decrease after this step.

In order to optimally solve the problem of flow re-distribution to minimize the maximum interference on the K channels we formulate the flow re-distribution problem as a Linear Program. Recall that $F(v)$ is the set of channels assigned to node v. A channel i is in $F(v)$ if and only if there exists an edge $e \in E$ incident on node v (outgoing or incoming) such that $f(e(i)) > 0$. We denote the total flow assigned to the copies of edge e by the previous step by $\rho(e)$. We will denote the new flows assigned by the LP to an edge e and its copy for channel i also by $f(e(i))$.

$$\min \beta \tag{13}$$

Subject to

$$\begin{aligned}
& f((s, s_v)) = \lambda^* l(v), \forall v \in V \\
& f(e(i)) \leq \beta c(e), \ \forall e \in E, \ 1 \leq i \leq K \\
& f(e(i)) = 0, \ \forall e = (u, v) \in E, \ \forall i \notin F(u) \cup F(v) \\
& \sum_{1 \leq i \leq K} f(e(i)) = \rho(e), \ \forall e = (u, v) \in E \\
& \frac{f(e(i))}{c(e)} + \sum_{e' \in I(e)} \frac{f(e'(i))}{c(e')} \leq \beta, \quad \forall e \in E, \ 1 \leq i \leq K
\end{aligned} \tag{14}$$

Note that in the RHS of the constraint (14) we use the term $\beta c(e)$ as the capacity of edge e. This is done to deal with edge capacity violations due to the channel assignment algorithm.

7.2 Flow Scaling

In this step the algorithm computes the maximum value of interference for the K channels, namely it computes a

scale L given by: $\zeta = \max\{1, \max_{1 \leq i \leq K} Int(i)\}$. Next the algorithm scales all flow values in the flow graph H by ζ: thus the new flow value for any edge e in the flow graph is set to: $f(e) = \frac{f'(e)}{\zeta}$ where $f'(e)$ is the flow value on edge e of the flow graph after the Post Processing Step of the algorithm. Note also that $\lambda^A = \frac{\lambda^*}{\zeta}$, where λ^A is the the new λ value corresponding to this scaled flow. Recall that λ^* is the optimal value for LP1 in Section 5. Finally that at the end of this step it is guaranteed that for each channel i the interference $Int(i) \leq 1$. This also implies that for any edge $e \in E$ and any channel i the interference $Int(i, e) \leq 1$.

8. ALGORITHM ANALYSIS

We now show that the algorithm RCL outlined in the overview section finds a feasible solution to the joint channel assignment, routing and interference free edge communication scheduling problem, is computationally efficient and has a provable worst case performance bound (a constant that depends only on the total number of channels). Since it is clear that routing, scheduling, post processing and scaling takes polynomial time, we only need to show that channel assignment step takes polynomial time in order to show that RCL runs in polynomial time.

LEMMA 5. *Algorithm 3 (Phase 1) runs in time polynomial in $|V|, |E|, K/I$ and ensures that the total aggregate fractional flow on all the edges introduced in E' for every edge $e \in E$ in Phase 1 equals the aggregate fractional flow on edge e in the original network G.*

PROOF. Referring to the pseudocode for Algorithm 3 for a given node $v \in V$ each time a copy $e' \in E'$ of its incident edge $e \in E$ is created to be added to the top of stack Q at least one new node $v_i \in V'$ among $v_1, v_2, \ldots v_{r_1}$ gets saturated ($\mu(v_i) = I(v_i)$). We show below that at or before all nodes $v_1, v_2, \ldots v_{r_1}$ get saturated the aggregate fractional flow on every edge $e \in E$ incident on node $v \in V$ is assigned to the edges introduced for it in E'. Thus at most $r_1 \leq K/I$ additional edges are added to Q in total for each vertex $v \in V$ implying a polynomial running time for phase 1.

In the original flow graph the total aggregate fractional flow on any node v is at most the number of its radios. Thus $\mu(v) \leq r_1 I + r_2 = \sum_{i=1}^{r_1} I(v_1)$. Note that at any step in the algorithm $\sum_{i=1}^{r_1} \mu(v_i) \leq \mu(v)$ with the inequality being strictly less than as long as the aggregate fractional flow $\mu(e)$ on some edge $e \in E$ incident on vertex v is not fully assigned to the edges introduced for it in E'. Thus as long as there is an edge $e \in E$ incident on vertex $v \in V$ whose aggregate fractional flow $\mu(e)$ is not fully assigned to the edges introduced for it in E' we have $\sum_{i=1}^{r_1} \mu(v_i) < \sum_{i=1}^{r_1} I(v_1)$ and therefore some node v_i must not be saturated. Thus the algorithm cannot reach a state where all nodes are saturated and the aggregate fractional flow $\mu(e)$ is not fully assigned to the edges introduced for it in E'. □

LEMMA 6. *Algorithm 4 (Phase 2) runs in time polynomial in $K, |V|, |E|$.*

PROOF. Referring to the pseudocode for Algorithm 4 for a given node $v \in V$ any of its incident edge $e \in E$ is processed only once in the inner while loop. This is because before processing the edge e in the inner while loop it is added to C so that it is not processed again. For a given edge e the inner while loop iterates at most $I+K$ times. This is because after each iteration either the channel k gets saturated for edge e: the maximum possible flow increase on edge $e(k)$ is done in the iteration or there is a $c(e)$ flow increase on some edge $e(i)$. The latter can only happen at most K times since $\mu(e(i)) \leq 1, 1 \leq i \leq K$ in the original flow graph and hence total flow on edge e is at most $\mu(e)c(e) \leq Kc(e)$. In the former case channel k does not appear in the set I' in any subsequent iterations for edge e. Thus with every such iteration $|I'|$ decreases by at least once and since $|I'| \leq I$ in the first iteration, the bound on the number of iterations follow. Thus, the algorithm runs in polynomial time. □

LEMMA 7. *If in the original flow graph H the flow satisfies the Link Congestion Constraint (1) for every channel $i, 1 \leq i \leq K$ then in the Algorithm 4 (Phase 2) the flow $\mu(e)c(e)$ on every edge $e \in E$ gets assigned to the edges $e(k)$ for channels $1 \leq k \leq I$. In other words on termination the following holds for all edges:*

$$\mu(e)c(e) = \sum_{i=1}^{I} f(e(i)).$$

PROOF. By Lemma 5 the total aggregate fractional flow and hence the total flow on all the edges introduced in E' for every edge $e \in E$ in Phase 1 equals the aggregate fractional flow and hence the flow on edge e in the original network G. Also note that the interference relationships between edges does not change in Phase 1. So the only change is that an edge $e \in E$ in the original network E may be split into multiple (mutually interfering edges) edges in E' whose total flow in the new network G' equals the flow on edge e in the network G.

Consider the flow on the original flow graph H. Since it satisfies the Link Congestion Constraint (1) for every channel $i, 1 \leq i \leq K$ we have in the flow graph H, $Int(i) \leq c(q)$ for every channel $i, 1 \leq i \leq K$. Thus for the flow in the flow graph H we have the total interference over all channels satisfies $\sum_{i=1}^{K} Int(i) \leq Kc(q)$. Thus in Phase 2 when the total aggregate fractional flow and hence the total flow on the edges in the network output by Phase 1 is distributed over the different channels the total interference over all channels must still be bounded above by $Kc(q)$. In particular this must hold at every iteration of the inner while loop of the Algorithm 4.

Referring to the pseudocode for Algorithm 4 and the proof of Lemma 6 it follows that if on termination of the inner while loop for an edge e it holds that $\sum_{i=1}^{I} f(e(i)) < \mu(e)c(e)$ then on the termination of the while loop all channels must get saturated due to the interference constraints. In other words after the iterations of the while loop for the edge e it must hold that the interference due to edge e on channel k satisfies $Int(e, k) = \frac{Kc(q)}{I}$ for all channels $k, 1 \leq k \leq I$. Also note that in Phase 2 the interference on any other channel is 0: $Int(i) = 0, i > I$. Hence it holds that the total interference due to edge e on all channels after the iterations of the while loop is given by $\sum_{k=1}^{I} Int(e, k) = Kc(q)$. But since $\sum_{i=1}^{I} f(e(i)) < \mu(e)c(e)$ there is still some positive flow left over for edge e that is not assigned to any channel at this point. When this flow is assigned to any of the channels the total interference due to edge e for all channels would strictly exceed $Kc(q)$. But as shown before no matter what distribution of edge flows over the channels is used the total channel interference and hence the total channel interference due to edge e

cannot exceed $Kc(q)$, a contradiction. This establishes that $\sum_{i=1}^{I} f(e(i)) = \mu(e)c(e)$ for all edges e in the flow distribution resulting from the phase 2 of the algorithm. □

It is easy to see that Phase 3 (Algorithm 5) runs in polynomial time since in each iteration the number of groups are reduced by at least one. Hence the total running time is bounded by the number of connected components in the K channels, and is therefore bounded by $|E|K$.

LEMMA 8. *After flow scaling, the resulting flow satisfies the link capacity constraints for each channel.*

PROOF. Recall that the flow scaling ensures that $Int(i) \leq 1$ for all channels i. This in turn implies that $Int(e, i) \leq 1$ for every edge $e(i)$ in every channel i. In particular this implies that $\frac{f(e(i))}{c(e)} \leq 1$ for all edges $e(i)$ in every channel i. This implies that the flow $f(e(i))$ on edge $e(i)$ is at most the capacity of edge e for every edge e with positive flow in any channel i. □

LEMMA 9. *At the end of Phase 3 (Algorithm 5) the resulting channel assignment is feasible.*

PROOF. Recall that in the channel assignment a node $v \in V$ is assigned channel i if and only if $f(e(i)) > 0$ for some edge $e \in E$ incident on node v. In Phase 1 of the algorithm a node $v \in V$ with $r_1 I + r_2$ is split into r_1 nodes in V' each with at least I radios. In Phase 2 each of these r_1 nodes in V' is assigned at most I channels each from among the channel set $\{1, 2, \ldots, I\}$. In phase 3 channels are reassigned to these r_1 nodes without increasing the number of channels assigned to each node. Thus these r_1 nodes in V' are assigned at most I channels each. Finally the set of channels assigned to these r_1 nodes are assigned to the corresponding node $v \in V$. Thus node v which has $r_1 I + r_2$ radios gets at most $r_1 I$ channels implying the feasibility of the channel assignment. □

THEOREM 2. *The RCL algorithm is a $\frac{Kc(q)}{I}$ approximation algorithm for the Joint Routing and Channel Assignment with Interference free Edge Scheduling problem.*

PROOF. Referring to the pseudocode for Algorithm 4 it follows that on termination of Phase 2 the interference on all channels is bounded as $Int(i) \leq \frac{Kc(q)}{I}$. Phase 3 (Algorithm 5) re-distributes the edge flows over the K channels without increasing the interference on any channel. This is because in Phase 3 the flows moved to a channel j all come from the edges $e(i)$ in a single channel i. Thus $Int(j) \leq Int(i) \leq \frac{Kc(q)}{I}$. Hence in Phase 3 we must have $\zeta \leq \frac{Kc(q)}{I}$. Postprocessing only reduces the maximum interference. In the flow scaling step, the flow is scaled by ζ. Therefore the scaled flow corresponds to a λ value which is at least $\lambda^A = \frac{\lambda^*}{\frac{c(q)K}{I}}$. Since the optimal λ value is at most λ^* and since the scaled flow satisfies the sufficient condition (Link Schedulability Constraint (2) for all channels) for it to be scheduled by Algorithm 2 the approximation bound follows. □

9. EVALUATION

In this section our goals are two fold: to evaluate the performance of our algorithm in realistic settings and to use our algorithm to study the performance gain of using multiple radios and multiple channels for wireless mesh networks (WMNs). For the first goal ideally we should compare the performance of our algorithm against the optimal solution. However the joint channel assignment and routing problem for WMNs quickly becomes intractable and any meaningful scenarios cannot be optimally solved in any practical setting. The other option therefore is to compare the average case performance of our algorithm versus the worst case bound we established earlier in the paper. This is what we evaluate here. For the second goal our evaluation is based on two sensitivity analysis studies that evaluate the improvement in the network throughput as the number of channels are increased and as the number of radios are increased.

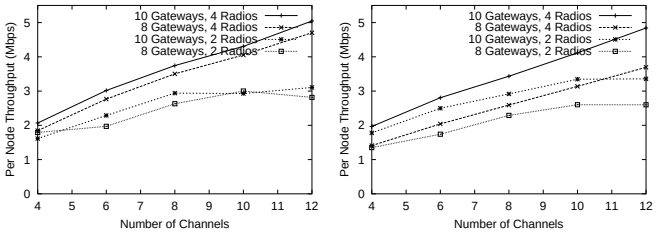
We solve the three linear programs in our RCL algorithm using CPLEX [2]. Our channel assignment algorithm uses the solution of LP2 as input. The channel assignment together with the total edge flow and λ^* from LP2 are the inputs to LP3. After postprocessing by LP3, we obtain ζ to get the feasible per-node throughput λ^*/ζ. We performed our evaluation on many realistic topologies and our simulation setup is as follows. The WMNs in our setting use 802.11a radios. We assume a simple wireless channel model in which link rates depend only on the distance between the links two end mesh network nodes. Adopting the values commonly advertised by 802.11a vendors, we assume that the link rate when the two end mesh nodes are within 30 meters is 54 Mbps, 48 Mbps when within 32 meters, 36 Mbps when within 37 meters, 24 Mbps when within 45 meters, 18 Mbps when within 60 meters, 12 Mbps when within 69 meters, 9 Mbps when within 77 meters and 6 Mbps when within 90 meters. We assume the maximum transmission range R_T of 90 meters and the maximum interference range of 180 meters. We assume there are 12 channels available according to 802.11a specification. For simplicity, we assume that the gateway nodes have sufficient wired backhaul capacity for them not to be a bottleneck.

We generate grid and random topologies. We run our simulation with different parameter settings. We report results with the following parameters. We have a total of 60 nodes. For the grid topology, the grid size is 8×8, the distance between two adjacent grid points is $0.65R_T$, and nodes are placed in grid points randomly. We generate 9 random connected topologies by placing nodes randomly in a 500×500 square meter area. we choose a random sample of 20 nodes to have a traffic demand of 20 Mbps each. We vary the number of gateway nodes from 2 to 12, the number of radios from 1 to 4 and the number of available channels from 1 to 12. We assume a uniform number of radios at all nodes.

9.1 The performance impact of multi-channel

In this evaluation, we varied the number of channels in the 60-node grid topology and random topologies to study its impact on the network throughput. The results are shown in Figure 4 for four settings with varying number of gateways and radios. Each data point for random topologies is averaged over the 9 topologies. As expected, we observe the trend that, as the number of channels increase, the per-node throughput generally increases. However, we remark that, the per-node throughput our algorithm computes may not always increase when the number of channels increase. This is because the channel assignment algorithm is not necessarily optimal and its performance depends on the network flow output by the routing step. In practice, one can use the so-

lution with the highest throughput output by the algorithm. From Figure 4, we see that our algorithm in general can effectively exploit the increasing number of channels available. For example with 10 gateways and 4 radios, as the number of channels goes from 4 to 12, the per-node throughput goes from 2.1Mbps to 5.0Mbps for the grid topology case; it goes from 2.0Mbps to 4.8Mbps for the random topologies case.

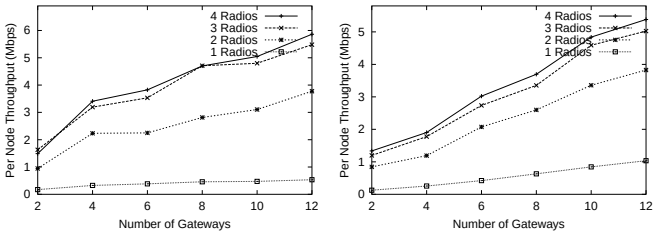


(a) Grid topology (b) Random topologies

Figure 4: Throughput improvement with increasing number of channels

9.2 The performance impact of multi-radio

In this evaluation, we varied the number of radios and gateway nodes in the 60-node grid topology and random topologies to study their impacts on the network throughput. We fix the number of channels to be 12. Each data point for random topologies is averaged over the 9 topologies. As can be seen from Figure 5, our algorithm is able to exploit the increase in the number of radios and gateways to obtain a solution with improved per-node throughput. We see that the per-node throughput increases significantly from the one radio case to two radio case, much more than the percentage increase from 2 to 3 and from 3 to 4 radio case. For example, when they are 12 gateways, for the grid topology, the throughput corresponding to 1,2,3,4 radio case is 0.53, 3.8, 5.5 and 5.9Mbps respectively; for random topologies, it is 1.0, 3.8, 5.0 and 5.4Mbps. With one more radio, we see a 620% and 280% increase in per-node throughput for the grid topology and random topologies respectively. This result justifies the use of a small number of radios.

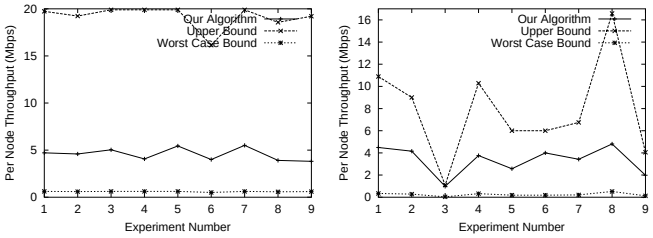


(a) Grid topology (b) Random topologies

Figure 5: Throughput improvement with increasing number of radios and gateways

9.3 Performance comparison with upper bound and worst case bound

Even to compute the worst case bound we need the optimal value for λ, the computation of which remains intractable in our setting. Thus we used a upper bound λ^* on this value provided by the linear program LP1(Eqn. 9). Therefore an estimate on the worst case throughput of the algorithm is $\frac{\lambda^*}{W} l(v)$ where $W = c(q)\frac{K}{I}$ (for any $v \in V$). We compare this value and λ^* with the actual throughput that our algorithm is able to achieve. The results are shown in Figure 6. In this evaluation we used 9 different grid and random topologies, each with 60 nodes. 20 nodes have traffic demands and there are 8 gateway nodes. Nodes have 3 radios each. We fix the number of channels to be 12. We can see that the algorithm's average case performance is around 5.3 to 7.9 and 8.3 to 28.7 times better than the worst case estimated performance for grid and random topologies respectively. Our algorithm is at most 4.0 and 2.4 times worse than the upper bound for grid and random topologies respectively. Note that, the upper bound is also very loose since the integrality gap may be large.



(a) Grid topologies (b) Random topologies

Figure 6: Comparison with upper bound and worst case bound

10. RELATED WORK

The work that is most closely related to this paper is that of [19, 18, 14, 17]. Like ours, the work in [19, 18] assumes that there is no system or hardware support to allow a radio interface to switch channels on a per-packet basis. Raniwala et al. propose a centralized joint channel assignment and multi-path routing algorithm. The channel assignment algorithm considers high load edges first. The routing algorithm uses both shortest path routing and randomized multi-path routing (a set of paths is used between any pair of communicating node pair). The joint channel assignment and multi-path routing algorithm proceeds in an iterative fashion. However, their algorithm is based on heuristics and a worst performance bound on its performance is not known. In addition in their scheme no guarantees on fair allocation of bandwidth is provided. In [18], Raniwala and Chiueh propose a distributed heuristic algorithm. The algorithm also is not known to have any worst case performance bound. Unlike ours, the work in [14, 17] assume a radio interface is capable of switching channels rapidly and is supported by system software. In [14], Kodialam and Nandagopal presents channel assignment and routing algorithms to characterize the capacity regions between a given set of source and destination pairs. In [17], Kyasanur and Vaidya study how the capacity of multi-channel wireless networks scale with respect to the number of radio interfaces and the number of channels as the number of nodes grow.

Algorithms aspects of wireless networks has been an active area of research. Jain et al. [12] consider throughput optimization using a general interference model. Their algorithm can be computationally intensive to achieve close to

optimal performance. In addition, their algorithm does not exploit the properties of interference using 802.11 MAC for better performance. Kumar et al. [16] consider the throughput capacity of wireless networks between given source destination pairs for various interference models. However, they do not take channel allocation into account as they consider a single-channel network. Kodialam and Nandagopal [13] investigate the same problem using a simple interference model where a node can not send and receive at the same time. Objectives other than throughput have also been considered, e.g. power optimization [8].

There have also been approaches that consider routing and channel assignment separately. In [10], Draves et al. propose a routing metric that exploits multi-channel diversity. In particular, paths with more channel diversity and fewer hops are preferred. In [5], Bahl et al. present a MAC protocol that exploits the availability of multiple channels. However, they change channel assignment in a fast time scale on a per-packet basis which may not work with existing commodity hardware.

In this paper, we assume the network is given. The problem of how to design a multi-hop mesh network has been studied in [6, 9]. Their goal is to place a minimal set of gateways to meet certain performance requirements. They do not consider multi-radio and multi-channel mesh networks and their algorithms do not apply in our setting.

Finally, fair bandwidth allocation and load balancing has been considered in single-radio wireless LAN context [7]. Channel assignment has been extensively studied in cellular networks. However, there is no multi-hop routing in that context.

11. CONCLUSION AND FUTURE WORK

Infrastructure mesh networks (IWMNs) are increasingly been deployed for commercial use and law enforcement. These deployment settings place stringent requirements on the performance of the underlying IWMNs. Bandwidth guarantee is one of the most important requirements of applications in these settings. For these IWMNs, topology change is infrequent and the variability of aggregate traffic demand from each mesh router (client traffic aggregation point) is small. These characteristics admit periodic optimization of the network which may be done by a system management software based on traffic demand estimation.

In this paper, we rigorously formulate the joint channel assignment and routing problem in IWMNs. Our goal is to maximize the bandwidth allocated to each traffic aggregation point subject to fairness constraint. We propose a constant approximation algorithm for this NP-hard problem. Our algorithm takes interference constraint into account and is based on flow transformation. Our evaluation shows that the algorithm performs much better that the worst case bounds.

For future work, we would like to investigate the problem when routing solutions can be enforced by changing link weights of a distributed routing protocol such as OSPF. We would also like to improve the worst case bounds of our algorithms.

12. REFERENCES

[1] Chaska wireless solutions. `http://www.chaska.net/`.

[2] ILOG CPLEX mathematical programming optimizers. `http://www.ilog.com/products/cplex/`.

[3] Meshdynamics inc. `http://www.meshdynamics.com/`.

[4] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: A survey. *Computer Networks Journal (Elsevier)*, 2005.

[5] P. Bahl, R. Chandra, and J. Dunagan. SSCH: slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks. In *Proc. ACM MOBICOM*, pp. 216–230, 2004.

[6] Y. Bejerano. Efficient integration of multi-hop wireless and wired networks with QoS constraints. In *Proc. ACM MOBICOM*, pp. 215–226, 2002.

[7] Y. Bejerano, S.-J. Han, and L. E. Li. Fairness and load balancing in wireless lans using association control. In *Proc. MOBICOM*, pp. 315–329, 2004.

[8] R. Bhatia and M. Kodialam. On power efficient communication over multi-hop wireless networks: Joint routing, scheduling and power control. In *Proc. IEEE INFOCOM*, pp. 1457–1466, 2004.

[9] R. Chandra, L. Qiu, K. Jain, and M. Mahdian. Optimizing the placement of internet TAPs in wireless neighborhood networks. In *Proc. IEEE ICNP*, pp. 271–282, 2004.

[10] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proc. ACM MOBICOM*, pp. 114–128, 2004.

[11] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, IT-46(2):388–404, Mar. 2000.

[12] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *Proc. ACM MOBICOM*, pp. 66–80, 2003.

[13] M. Kodialam and T. Nandagopal. Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem. In *Proc. ACM MOBICOM*, pp. 42–54, 2003.

[14] M. Kodialam and T. Nandagopal. Characterizing the capacity region in multi-radio and multi-channel mesh networks. In *Proc. ACM MOBICOM*, 2005.

[15] S. Kravitz. Packing cylinders into cylindrical containers. In *Math. Mag.*, volume 40, pp. 65–70, 1967.

[16] V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. Algorithmic aspects of capacity in wireless networks. In *Proc. ACM SIGMETRICS*, pp. 133–144, 2005.

[17] P. Kyasanur and N. Vaidya. Capacity of multi-channel wireless networks: Impact of number of channels and interfaces. In *Proc. ACM MOBICOM*, 2005.

[18] A. Raniwala and T.-C. Chiueh. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *Proc. IEEE INFOCOM*, 2005.

[19] A. Raniwala, K. Gopalan, and T.-C. Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *ACM Mobile Computing and Communications Review (MC2R)*, volume 8(2), pp. 50–65, 2004.