Optimal Multicast in Multi-Channel Multi-Radio Wireless Networks

Ajay Gopinathan, Zongpeng Li, Carey Williamson
Department of Computer Science, University of Calgary
{agopinat,zongpeng,carey}@cpsc.ucalgary.ca

***Abstract*—Recent advances in wireless technology have made it increasingly feasible to equip wireless nodes with multiple radios, thereby allowing each radio to exploit channel diversity in the form of orthogonal, non-overlapping transmission spectrums. Multi-channel operation mitigates interference, but at the same time raises new challenges for network optimization, in terms of judicious channel assignment for efficient bandwidth utilization. While previous research mostly studies optimizing channel assignment for unicast, we focus instead on multicast, which is an efficient mechanism for one-to-many data dissemination. We derive a model for optimal multicast in multi-channel multi-radio wireless networks under the assumption that channel assignment is static. Our model employs network coding as the multicast mechanism of choice, and exploits the broadcast nature of omni-directional antennas for efficient bandwidth utilization. Based on the model derived, we formulate optimal multicast as a linear integer program. Two accompanying solutions are proposed: a greedy channel assignment scheme and an improved iterative scheme inspired by primal-dual algorithm design. The effectiveness of the two schemes are empirically examined through simulation studies, and are compared to results obtained from solving the integer program as well as its linear programming relaxation. Finally, we present an alternate model for optimal multicast under the assumption that transmission frequencies are not fixed divisions of the usable spectrum.**

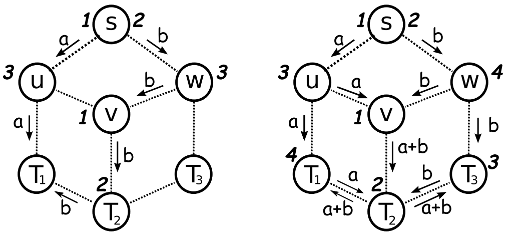
I. Introduction

Wireless networks have become increasingly prevalent as the preferred choice for network connectivity in many military and civil applications. For instance, wireless networks provide mobility-friendly network access, and wireless LANs and wireless mesh networks often constitute a cost-effective *last mile* connection to the Internet.

A fundamental problem that limits the performance of wireless networks is interference. Despite the advertised transmission speeds for wireless networks, the actual goodput achievable is approximately halved when interference effects and medium access contention are taken into account [1]. Interference can be caused by external sources, or by transmission from nearby wireless nodes. As an example of the former, the IEEE 802.11b and 802.11g standards, which operate in the 2.4 GHz spectrum, are susceptible to interference caused by microwave ovens, bluetooth devices and even older cordless telephones. To counter the effects of interference, channel diversity can be exploited. The IEEE 802.11 standard divides the usable spectrum into a number of channels; for example, the 2.4000 GHz to 2.4835 GHz band is divided into 13 channels. Three of them (channels 1, 6 and 11) occupy non-overlapping frequency spectrums, and transmissions on these orthogonal channels do not interfere.

To take advantage of channel diversity, wireless nodes can be equipped with *multiple* wireless network cards or radios. Advances in wireless technology have demonstrated that multiple radios per wireless node are a feasible and cost-effective solution [2]. Furthermore, recent research has shown that considerable performance gain can be achieved in multi-channel multi-radio wireless environments [1]–[4]. This raises interesting new challenges in the design of algorithms and mechanisms that are capable of fully exploiting channel diversity in multi-radio networks to optimize bandwidth usage, network throughput or routing cost.

Previous research in multi-channel multi-radio wireless networks has studied optimal unicast or multi-session unicast [3], [5], [6]. In contrast, we focus on the *multicast* problem instead. Multicast is an efficient way of disseminating identical data to multiple users and has numerous applications in both wireline and wireless networks [7]–[10]. Unicast and multicast differ in terms of what constitutes a judicious *channel assignment* scheme, which assigns one of the available channels to each radio at each node. The set of available channels is a limited resource, and an optimal channel-to-radio assignment in the case of unicast may not necessarily be optimal for interference-free multicast routing. Consider the example in Figure 1. Here, the source S needs to send information to potential receivers T_1, T_2 and T_3. Each node is equipped with 3 radios of unit capacity each, where separate radios are used to transmit and receive. The channels used for *transmission* are indicated next to each node. Channels are assigned so as to avoid interference, hence neighbouring nodes (indicated by the dotted lines) do not transmit on the same channel. Similarly, nodes receiving transmissions from multiple neighbours receive on different channels as well. In Figure 1(a), the channel assignment is an optimal assignment in the case of unicast from S to receiver T_1, allowing receipt of both bits a and b for a total throughput of 2 bits. Similar channel assignments can be made sequentially for unicast flows from S to T_2, and from S to T_3. However, these channel assignments would not work for a network coded multicast session to all three receivers *simultaneously*. Exploiting network coding for bandwidth efficiency requires us to send the logical XOR of bits a and b from node v to T_2, and from node T_2 to T_1 and T_3. Using the channel assignment in Figure 1(a), node v would not be able to listen to nodes u and w concurrently since both of these nodes transmit on channel 3, and thus their transmissions would interfere at v.



(a)

(b)

Fig. 1. An optimal channel assignment scheme in the case of unicast is shown in (a). The flow of bits a and b are shown from S to T_1. This assignment differs from the optimal channel assignment for multicast, shown in (b)

The optimal channel assignment for interference-free multicast is instead shown in Figure 1(b). Node w transmits on channel 4 instead of 3, allowing for interference-free reception at v.

In this paper, we seek to derive a complete mathematical model of multi-channel multi-radio wireless networks, and formulate a mathematical program for solving the joint routing and channel assignment problem for optimal multicast. We exploit the principles of network coding [11], [12] for optimal multicasting, and develop routing constraints that exploit the *broadcast property* of wireless networks for efficient bandwidth utilization. The result is a linear integer program that computes a joint channel assignment and flow routing scheme, achieving maximum end-to-end multicast throughput.

Since in practice integer programs can be solved for small networks only, we design two algorithms for larger networks. In the first algorithm, we relax the limitations imposed by channel and radio availability, and solve the resulting linear program for an 'optimistic' multicast flow. Channel assignment then proceeds in a greedy fashion, in breadth-first order along the computed flow. Then a new, feasible multicast flow is computed by plugging the channel assignment information into the optimal multicast integer program and solving the resulting linear program. The second algorithm employs the first one for generating an initial channel assignment. Then it iteratively refines the channel assignment and the routing scheme based on the new solution obtained. This is inspired by primal-dual algorithm design, where a pair of primal and dual solutions are iteratively refined until convergence. In each iteration of the primal-dual algorithm, the new solutions in the primal (dual) provide directions for updating the dual (respectively, primal) in the next round [9]. We study these two algorithms through extensive simulations, and show their efficacy by comparing them with results obtained from solving the integer program and its linear programming relaxation. Finally, we also present an alternate model for channel assignment under the assumption of flexible spectrum division.

The rest of the paper is organized as follows. Section II discusses related work. In Section III, we derive a mathematical framework to model multicast in multi-channel multi-radio wireless networks, and present an integer program that maximizes end-to-end throughput. We then proceed to outline two channel assignment algorithms. We show simulation results in Section IV. Section V contains the alternate model for channel assignment under the assumption of flexible spectrum division. The paper is concluded in Section VI.

II. Related Work

Multicast is used for one-to-many dissemination of data, and is seen as an attractive data delivery mechanism, especially with the increased proliferation of applications such as multimedia streaming. Consequently, the multicast problem has received a lot of attention from the networking research community [7], [13]–[15]. Computing optimal multicast routing (without network coding) is equivalent to the well-studied problem of finding and packing Steiner trees [7], [16]. The latter has however been shown to be NP-hard [17], though constant approximation algorithms exist using LP rounding [18], primal-dual schemes [19] and minimum spanning tree heuristics [20]. A more recent approach in the literature is network coding, which exploits the encodable as well as replicable property unique to data flows. The fundamental result of network coding [11], [12] states that a target multicast rate of d is achievable if and only if the rate d is a feasible unicast rate from the source to each and every receiver. Using this result, multicast can be reduced to computing the optimal union of unicast flows to each and every receiver. The latter can be computed efficiently using suitably formulated linear programs [9], [10], [21].

Multi-channel wireless networks have recently received attention from the research community, mostly in the context of wireless mesh networks [22]. Raniwala *et al.* considered the benefits of using multiple channels in wireless mesh networks [3]. While prior work considered modifications to the IEEE 802.11 MAC protocol, Raniwala *et al.* focused on existing IEEE 802.11 MAC protocols, and presented a load-aware channel assignment algorithm. Their algorithm is based on a greedy heuristic. Nevertheless, they show via simulations that even 2 radios per node can lead to throughput improvement by a factor of 8. Raniwala and Chiueh [1] extend this work by proposing distributed channel assignment algorithms and demonstrate its effectiveness by performing experiments on a 9-node network equipped with commodity 802.11a network cards. They find that in the context of long lived TCP flows, aggregate throughput can be 5 times higher compared to single channel operation.

The seminal work of Gupta and Kumar [23] studied and characterized the achievable capacity of wireless ad-hoc networks with a single channel. Their work is applicable to the multi-channel scenario under the assumption that radios are not able to switch channels on a per packet basis. In subsequent work, Kyasanur and Vaidya [24] extend the analysis of Gupta and Kumar and characterized the capacity region for multi-channel networks where radios are allowed to switch channels. They find that the capacity of multi-channel multi-radio networks is highly dependent on the ratio between the number of radios and available channels.

The channel assignment problem is known to be NP-hard, and has received much attention in the literature. Notably, Brzezinski *et al.* [25] study the problem from a graph theoretic point of view. They characterize interference patterns using line graphs, and present a scheme that partitions the network using *local pooling*, a greedy scheduling algorithm due to Dimakis and Walrand [26]. Brzezinski *et al.* use local pooling to partition the network and show thereafter that simple distributed channel assignment and scheduling algorithms is sufficient to achieve maximal throughput. However, the latter is true only for a restricted class of network topologies. In contrast, Xing *et al.* [27] take an altogether different approach and solve the channel assignment problem using coding theory. Specifically, they use superimposed codes that are distributed to nodes in the network. These nodes then attempt to choose a channel based on these codes. However, this scheme suffers from the large dimension of codes required, and the likelihood of failure of the code in highly connected networks. Finally, a more practical approach is considered by Ko *et al.* [4] in which a distributed greedy heuristic is used for channel assignment. The heuristic entails that nodes continuously try to choose channels that suffer the least interference. They show that this assignment scheme is guaranteed to converge to an equilibrium whereby no node can improve local interference by switching channels. Experiments performed using IEEE 802.11a/g network cards show a 50% improvement in throughput.

The use of multi-channels in multi-radio networks necessitates the design of new routing metrics. Extending the notion of the Expected Transmission Count (ETX) metric proposed by De Couto *et al.* [28] for multi-hop wireless routing, Draves *et al.* proposed the Weighted Cumulative Expected Transmission Time (WCETT) [29] for multi-channel multi-radio wireless networks. The proposed routing metric incorporates the vagaries of multi-channel routing in that it explicitly takes into account interference effects, which it tries to minimize. The routing metric was subsequently implemented as a virtual network driver, and experiments show that in a prototype 2-radio network, WCETT achieves 89% higher throughput than ETX, and in comparison with shortest path routing, an improvement of 254% in throughput is observed.

The work of Kodialam and Nandagopal [6] as well as Alicherry *et al.* [5] is most relevant to the work presented in this paper. Both form mathematical models of multi-channel multi-radio networks. Interference due to transmissions on the same channel is considered as a constraint, and linear programming is used to provide a first approximation of the optimal flow route. Kodialam and Nandagopal study multicommodity routing, and provide heuristics for both static and dynamic channel assignment. The channel assignment algorithm scales the fractional flow computed by the linear program to integral values by multiplication with a large factor, which is used to guide scheduling in which a node is assigned a channel for a set number of time slots dependent on flow values on that channel. A similar flow scaling approach for scheduling and channel assignment is presented by Alicherry *et al.*, and in addition, the authors show that the throughput achieved using their method is within a constant factor of the optimal. In contrast to both papers, we study multicast routing in multi-channel multi-radio wireless networks instead.

III. Multicast Routing and Channel Assignment

In this section, we define our wireless network model and state our assumptions. We then proceed to model optimal multicast routing in multi-radio multi-channel wireless networks as an integer program. Our integer program is complete in that the constraints capture necessary and sufficient conditions to ensure interference free routing assuming no scheduling is allowed, i.e., we consider static channel assignment only in this paper. Due to intractability of integer programs in general, we further present two channel assignment algorithms in this section. The first is a simple greedy assignment scheme, while the second is a more sophisticated algorithm that iteratively makes improvements on the residual network of the flow graph to achieve higher throughput.

A. Preliminaries

A wireless network is modeled as a graph $G = (V, E)$, where V and E denote the set of nodes and edges respectively. Nodes are equipped with multiple radios with a uniform coverage radius r. The set of radios in each node u is $R(u)$, and each radio can be tuned to any channel in the set of channels K. The bandwidth on each radio is limited, and we use $c_j(u)$ to denote the capacity of the j-th radio on node u. Each channel $i \in K$ is orthogonal to all other channels $j \in K$, in the sense that transmissions on channels $i, j \in K$ do not interfere if $i \neq j$. A node v is said to be a *neighbour* of u if v is located within distance r of u. The set of u's neighbours is denoted $N(u)$, while neighbours within two hops of u are in the set $N_2(u)$. An edge $\vec{uv} \in E$ if and only if $v \in N(u)$. Edges are directed and symmetric due to the uniform radius of coverage property. Our aim is to compute a multicast routing that is optimal in the sense that it maximizes end-to-end throughput from some designated source node s, to the set of k multicast receivers $T = \{t_1, \ldots, t_k\}$. We abstract the dissemination of data from the source to the set of receivers as a network *flow* from the former to the latter. In reality, a unit of flow can be considered to be an arbitrary unit of information. For example, a unit of flow may correspond to a bit of information, in which case the capacity of a radio would indicate the corresponding limit of information flow per time unit, e.g. bits per second.

For the purpose of computing the optimal flow, we model the flow rate on a link using flow variables both with and without an associated channel usage, and we will later define how these variables are related. Recall that with network coding, computing the optimal multicast flow is equivalent to computing the optimal *union* of unicast flows from the source to each and every receiver. We will call these unicast flows *conceptual flows* [9], [21]. Conceptual flows coexist in the sense that they do not compete for bandwidth. We denote conceptual flow on a link as $f_t(.)$, which represents the flow from the source to receiver $t \in T$. A flow on a link uses one of many available channels, and hence $f_t^i(.)$ is the conceptual flow to receiver $t \in T$ using channel $i \in K$.

We will use binary variables to express how radios on a node are assigned to channels. Further, we will distinguish radios that are transmitting data from those that are listening. The binary variables $x_j^i(u)$ and $y_j^i(u)$ for the node u have the following interpretation

$$x_j^i(u) = \begin{cases} 1 & \text{radio } j \text{ on } u \textit{ transmits} \text{ on channel } i, \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$y_j^i(u) = \begin{cases} 1 & \text{radio } j \text{ on } u \textit{ listens} \text{ on channel } i, \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

B. An Integer Programming Formulation

Having stated our model, we next need to clearly define how the flow variables as well as binary variables are related. The inter-dependence of these various flow variables is crucial and together with the constraints that capture the effect of interference, completely characterizes the feasible flow region. In this section, we will state the requirements for a multicast flow that is *interference-free and optimal*.

We begin with flow conservation. To more succinctly express flow conservation constraints, we introduce a virtual, uncapacitated directed link $\overrightarrow{ts}$ from every $t \in T$ to s. For each conceptual flow, flow conservation must therefore be observed at every node in the network, captured by the following constraint:

$$\sum_{u \in N(v)} f_t(\overrightarrow{uv}) - \sum_{u \in N(v)} f_t(\overrightarrow{vu}) = 0 \quad \forall t \in T, \forall v \in V \quad (3)$$

The above states that the sum of all incoming flow destined to each receiver t on edges $\overrightarrow{uv}$ for all neighbours u of v must equal the sum of the conceptual flow on outgoing links. This constraint must hold for all nodes in the network.

Employing network coding, a multicast rate d can only be achieved if d is a feasible unicast rate to all receivers. Hence, in the optimal multicast flow, a rate of d must be achieved by each receiver:

$$f_t(\overrightarrow{ts}) = d \quad \forall t \in T \quad (4)$$

The conceptual flow on a link $\overrightarrow{uv}$ can use one or more channels, and so:

$$\sum_{i \in K} f_t^i(\overrightarrow{uv}) = f_t(\overrightarrow{uv}) \quad \forall \overrightarrow{uv} \in E, \forall t \in T \quad (5)$$

Recall that conceptual flows do not compete for bandwidth [9], [21]. Hence, the actual outgoing flow from any node on any channel is the maximum of all conceptual flows using that channel on the node's outgoing links. Note that the $\max$ function is non-linear. However, the following constraint captures our requirement in an equivalent fashion in the context of the optimization problem:

$$\sum_{v \in N(u)} f_t^i(\overrightarrow{uv}) \le \sum_j c_j(u) x_j^i(u) \quad \forall t \in T, \forall i \in K, \forall u \in V (6)$$

There are several things worth noting about constraint (6). First, (6) implicitly takes the *wireless broadcast property* into consideration. The broadcast property is unique to wireless networks, as transmission to any node in the neighbourhood of u can be overheard by any node $v \in N(u)$. Clearly, any multicast formulation should seek to exploit this property. The constraint in (6) states that transmissions on channel i can potentially flow to any neighbouring node, and further, the bandwidth utilized by the broadcast is limited to the capacity of the radio used for transmission. Note also that (6) allows multiple radios to be used to transmit on a single channel. Clearly, this results in interference and is undesirable. Nevertheless, we will impose further restrictions to ensure that at most one radio on a node can be assigned to a given channel. The sum over all radios $j \in R(u)$ here then takes on a different semantic, namely, regardless of the choice of radio used, the flow is upper bounded by the bandwidth provisioned on that radio. Finally, the binary variable $x_j^i(u)$ is used to ensure that if a radio is used for any non-zero amount of outgoing flow on channel i, then that radio is assigned to that channel.

Next, we relate incoming flow to channel assignment on the *listening* radio of a node. We will use conceptual flow variables on incoming links to state the following requirement for channel assignment on the receiving radio:

$$\sum_{v \in N(u)} f_t^i(\overrightarrow{vu}) \le \sum_j c_j(u) y_j^i(u) \forall t \in T, \forall i \in K, \forall u \in V \quad (7)$$

The above constraint states that to receive flow on a given channel, there must be at least one radio that is tuned to listen on that channel. In the case of heterogeneous capacities on radios, the flow across a link is bounded by the capacities of both the transmitting and receiving radio, and thus the true capacity is the minimum of these two radios. This model works well in reality, since the IEEE 802.11 standard specifies mechanisms for dynamic rate shifting [30], thereby allowing the transmitting and receiving radios to agree on the acceptable data rate.

Next we deal with channel assignment and node-radio constraints. First, a radio can either be transmitting or listening, but never both:

$$x_j^i(u) + y_j^i(u) \le 1 \quad \forall i \in K, \forall j \in R(u), \forall u \in V \quad (8)$$

Second, the number of radios on any node is limited, and so:

$$\sum_i \sum_j x_j^i(u) \le |R(u)| \quad \forall u \in V \quad (9)$$

$$\sum_i \sum_j y_j^i(u) \le |R(u)| \quad \forall u \in V \quad (10)$$

Third, a node can only transmit (receive) on a given channel using a single radio to avoid local interference:

$$\sum_j x_j^i(u) \le 1 \quad \forall i \in K, \forall u \in V \quad (11)$$

$$\sum_j y_j^i(u) \le 1 \quad \forall i \in K, \forall u \in V \quad (12)$$

Finally, we require interference-free transmission. We will impose constraints based on the protocol interference model [6]. The IEEE 802.11 RTS-CTS mechanism requires nodes to request clearance from the potential receiver before transmitting. The protocol also acts as a virtual carrier sensing mechanism, in that neighbouring nodes are capable of knowing when there is an impending transmission by other nodes within interference range. This substantially reduces the number of collisions. In this model, neighbours of both the transmitter and intended recipients should not transmit at the same time on the same channel. The following constraint captures this behaviour:

$$\sum_j \Big(x^i_j(u) + \sum_{v\in N_2(u)} x^i_j(v)\Big) \le 1 \quad \forall i \in K, \forall u \in V \tag{13}$$

This final constraint requires that for a node u to transmit on a given channel, no other node v within two hops of u is allowed to transmit on that channel. Observe that this constraint can easily be modified to work under the *primary interference* model [5], [25] as well by replacing the set $N_2(u)$ with $N(u)$.

Our objective is to maximize the target multicast rate d. We now present our integer program for optimizing end-to-end multicast throughput in its entirety:

$$\text{Maximize} \quad d \tag{14}$$

Subject To

$$f_t(\overrightarrow{ts}) = d \qquad \forall t \tag{14a}$$

$$\sum_{u\in N(v)} f_t(\overrightarrow{uv}) - \sum_{u\in N(v)} f_t(\overrightarrow{vu}) = 0 \qquad \forall t, \forall v \tag{14b}$$

$$\sum_{i\in K} f^i_t(\overrightarrow{uv}) = f_t(\overrightarrow{uv}) \qquad \forall\, \overrightarrow{uv}, \forall t \tag{14c}$$

$$\sum_{v\in N(u)} f^i_t(\overrightarrow{uv}) \le \sum_j c_j(u) x^i_j(u) \qquad \forall t, \forall i, \forall u \tag{14d}$$

$$\sum_{v\in N(u)} f^i_t(\overrightarrow{vu}) \le \sum_j c_j(u) y^i_j(u) \qquad \forall t, \forall i, \forall u \tag{14e}$$

$$x^i_j(u) + y^i_j(u) \le 1 \qquad \forall i, \forall, \forall u \tag{14f}$$

$$\sum_i \sum_j x^i_j(u) \le |R(u)| \qquad \forall u \tag{14g}$$

$$\sum_i \sum_j y^i_j(u) \le |R(u)| \qquad \forall u \tag{14h}$$

$$\sum_j x^i_j(u) \le 1 \qquad \forall i \in K, \forall u \tag{14i}$$

$$\sum_j y^i_j(u) \le 1 \qquad \forall i \in K, \forall u \tag{14j}$$

$$\sum_j \big(x^i_j(u) + \sum_{v\in N_2(u)} x^i_j(v)\big) \le 1 \qquad \forall i, \forall u \tag{14k}$$

$$f_t(\overrightarrow{uv}), f^i_t(\overrightarrow{uv}) \ge 0 \qquad \forall\, \overrightarrow{uv}, \forall t, \forall i \tag{14l}$$

$$x^i_j(u) \in \{0,1\}, y^i_j(u) \in \{0,1\} \qquad \forall i, \forall j, \forall u \tag{14m}$$

C. Linear programming relaxation

The integer program presented in (14), is both accurate and complete. However, solving it to exact optimality is computationally infeasible when the input size is large. This is due to the nature of the channel assignment problem, which is inherently NP-Hard. If one relaxes the binary variables $x^i_j(u)$ and $y^i_j(u)$ to freely take on fractional values between 0 and 1, then the linear integer program in (14) degrades into a linear program with continuous variables only, and can be solved efficiently using interior-point algorithms or simplex algorithms. Since the solution to the LP relaxation may assign fractional values to $x^i_j(u)$ and $y^i_j(u)$, it is therefore not feasible in general. Nonetheless, the maximum throughput computed from the LP relaxation provides an upper-bound for the maximum throughput from the original integer program. We next describe two algorithms for solving the integer program, and use the LP relaxation as a benchmark for their performance in simulation studies later on. Note that if the throughput achieved by the algorithm is close to that achieved in the LP relaxation, then it is even closer to the real feasible maximum throughput (the optimal solution to the integer program).

D. A greedy channel assignment algorithm

The first of our two channel assignment algorithms is relatively straightforward. The algorithm seeks to greedily assign channels to create a viable, interference-free route from the source to each and every receiver. We begin by solving the following linear program (see also [21]) that computes the optimal multicast flow using network coding.

$$\text{Maximize} \quad d \tag{15}$$

Subject To

$$f_t(\overrightarrow{ts}) = d \qquad \forall t \tag{15a}$$

$$\sum_{u\in N(v)} f_t(\overrightarrow{uv}) - \sum_{u\in N(v)} f_t(\overrightarrow{vu}) = 0 \qquad \forall t, \forall v \tag{15b}$$

$$\sum_{v\in N(u)} f_t(\overrightarrow{uv}) \le \sum_j c_j(u) \qquad \forall t, \forall u \tag{15c}$$

$$f_t(\overrightarrow{uv}) \ge 0 \qquad \forall t, \forall\, \overrightarrow{uv}, \forall u \tag{15d}$$

The constraint in (15c) captures the wireless broadcast property, and further bounds the total outgoing flow from a node u to the total available capacity of all radios on u. The linear program in (15) does not compute a meaningful flow since channel assignment and interference effects are not taken into account. Nonetheless, it does give us an indication of how important nodes are in the optimal flow. While a more fine-grained approach can be considered by examining the *amount* of outgoing flow on each node, we choose instead to consider all nodes with non-zero flow as a first approximation. Let these nodes be denoted by the set W. We can then safely prune all other nodes in the network and consider only nodes in W for channel assignment.

The greedy channel assignment algorithm proceeds in a breadth-first search fashion, beginning at the source. If there is an available radio at a node, we assign the first available channel to that radio for transmission. Channel availability depends on the interference model we choose; in the case of the protocol interference model, we choose the first available channel that is not being used for transmission by nodes within 2 hops. Next, we assign a free radio at each neighbour of the

```
Input: $W$, set of nodes to consider for assignment
NotDone := True ;
while NotDone do
    Visited := $\emptyset$ ;
    enqueue(ToVisit, $s$);
    while ToVisit $\neq \emptyset$ do
        $u$ := dequeue(ToVisit) ;
        Visited := Visited + $\{u\}$ ;
        foreach $v \in N(u)$ — $v \in W$ do
            if $v \notin$ Visited and $v \notin$ ToVisit then
                enqueue(ToVisit, $v$)
        Pick available radio of $j$ with highest capacity ;
        Assign first available channel $i$ to $j$ ;
        if no available radio then
            if $u = s$ then
                NotDone := false ;
            else
                continue ;
```

Algorithm 1: A greedy channel assignment algorithm

current node to listen on the chosen channel. This establishes a connection between the node and its neighbours. We continue to assign channels in this fashion (breadth-first) until each receiver has been assigned a listening channel.

At this point, we have a feasible channel assignment to accommodate flow from the source to all receivers. However, the process has thus far only used one radio each for transmission and reception on every node. Since nodes may have more than two radios, the breadth-first assignment process repeats again beginning at the source, while ensuring that these new channel assignments do not interfere with the existing assignment. This process repeats until there are no longer any unassigned radios. Once this channel assignment stage is complete, the variables $x_j^i(u)$ and $y_j^i(u)$ in (14) take on fixed values, and we can then solve the resulting degraded linear program of IP (14) for the optimal multicast flow. The complete greedy channel assignment algorithm is shown in Algorithm 1.

E. An iterative channel assignment scheme

The greedy channel assignment scheme in Algorithm 1 can be improved. In this section, we pursue a solution that *iteratively* improves the existing channel assignment based on the computed flow graph. Our solution is inspired by primal-dual algorithms, in particular the scheme used by Li [9]. The primal-dual method of Li seeks to maximize multicast flow in undirected networks iteratively, where in each iteration, the computed flow is used to update the bandwidth allocation in each orientation of the undirected links in the network. Our iterative algorithm operates in a similar manner, using the computed flow to guide and improve the existing channel assignment in each iteration. The improved algorithm proceeds in two phases. In the first phase, we run the greedy assignment scheme iteratively on the residual network resulting from the optimal flow computed by (14), and in the second phase, we make adjustments to the assignment obtained in the first phase to improve channel and radio usage.

```
Input: $G = (V, E)$, graph model of wireless network
Compute Maxflow, let $W$ be set of nodes with non-zero flow ;
Run greedy assignment scheme in Algorithm 1 ;
NotDone := True ;
while NotDone do
    Compute optimal flow with (14), let $f$ be flow graph ;
    Construct residual network $G'$ using $f$ ;
    Compute Maxflow, let $W$ be set of nodes with non-zero flow ;
    Run greedy assignment scheme in Algorithm 1 ;
    if No Available Radio/Channel or Maxflow is 0 then
        NotDone := False ;
Visited := $\emptyset$ ;
$I$ := $\emptyset$ ;
Enqueue(ToVisit, $s$) ;
while ToVisit $\neq \emptyset$ do
    $u$ := Dequeue(ToVisit) ;
    if $\exists i' \in K, j' \in R(u) | y_{j'}^{i'}(u) = 1$ then
        foreach $v \in N(u)$ do
            if $\exists j \in R(v) | x_j^{i'}(v) = 1$ then
                if $f^{i'}(v) > 0$ and $\exists r \in R(u) | y_r^i(u) = 0$ and $x_r^i(u) = 0, \forall i \in K$ then
                    $x_r^{i'} := 1$ ;
                else
                    $x_j^{i'}(v) := 0$ ; $y_{j'}^{i'}(u) := 0$ ;
                    $y_{j'}^{i'}(w) := 0, \forall w \in N(v) | y_r^{i'}(w) = 1$ ;
                    $I := I + \{u, v, w\}$ ;
    foreach $v \in N(u)$ do
        Enqueue(ToVisit, $v$) ;
NotDone:=True ;
while NotDone do
    Run Algorithm 1 on all nodes excluding $I$ ;
    Compute optimal flow using (14) ;
    if No improvement in flow rate then
        NotDone:= False ;
```

Algorithm 2: An iterative channel assignment scheme

Similar to Algorithm 1, we begin by computing the max flow and considering only nodes with non-zero flow for channel assignment. Channels are assigned in a greedy fashion via a breadth-first search beginning at the source node as before. At this point, we have an *initial* channel assignment, which we then use to solve the degraded linear program of (14). Using the flow graph obtained from the degraded linear program of (14), we construct the *residual network* by removing radios with non-zero flow from each node. This results in decreased capacity in each node, and once again, we repeat the process of computing maxflow and greedy assignment. We continue in this fashion until we no longer have any free radios/channels or the max flow computation yields a zero flow rate.

At the end of the first phase, we may still have inefficient channel assignments. In the second phase, we examine the channel assignment as well as the flow graph, and adjust the assignment to either improve bandwidth utilization or free channel usage for possible use in future assignments.

Using a breadth-first search, we look for a node u that is listening to some channel i, such that the incoming flow on that channel is zero. This indicates either one of the following two possibilities: u has been assigned to listen to some neighbour $v \in N(u)$ that is not forwarding flow on channel i, or v is indeed forwarding data but they are not intended for u. In either case, we can take remedial action to improve channel assignment. If the outgoing flow from v on channel i is zero, then we can safely release the channel from use by u, v and all $w \in N(v)$ that are also listening on i. If v is transmitting a non-zero flow rate, then we assign a new free radio of v to transmit on free channel j (if both are available), and assign u to listen on j instead of i.

Due to the process of releasing channels that are not being used, we now have more channels available, which may be used by other nodes. We then iteratively do the following; run the greedy assignment scheme once again while excluding the nodes that released channels in the beginning of the second phase, and compute the flow. If radios newly assigned with the newly available channels have non-zero flow, we keep the assignment; otherwise, we release it and repeat the assignment phase while excluding these nodes from assignment as well. We continue in this manner until the achieved flow can no longer be improved. We state the entire procedure in Algorithm 2. For succinctness, we slightly abuse notation in Algorithm 2 and denote the outgoing flow on node u on channel i with the variable $f^i(u)$, which has the following interpretation

$$f^i(u) = \max_{t \in T} \sum_{v \in N(u)} f_t^i(\vec{uv})$$

IV. Simulation Results

In order to study the effect of the number of radios and available channels on end-to-end multicast throughput, we ran simulations on various network topologies. We generated wireless networks using the unit disk graph model, where all nodes have a uniform radius of communication, r. Using a square grid of appropriate size, we generated nodes by assigning them to cells in the grid at random. Links between nodes were created accordingly based on the parameter r. For a given network, the choice of nodes in the multicast group was chosen at random. Each node is equipped with multiple radios, and the capacities of these radios are randomly distributed between 10 and 50.

As stated before, integer programs are NP-hard and hence, solving the integer program in (14) for large networks is a computationally intensive process. Hence, for small networks, we solved the integer program (IP) in (14), the linear program (LP) relaxation of (14) and the solutions based on Algorithm 1 (Greedy) and Algorithm 2 (Iterative), while for larger networks, only the results of the last three are compared. The throughput shown for each solution is the average computed from 4 separate simulation runs over randomly generated topologies.

To solve integer and linear programs, we employed the GNU Linear Programming Kit [31]. In particular, the branch and cut method was used when solving integer programs while the classic simplex algorithm of George Dantzig [32] was used to solve linear programs.

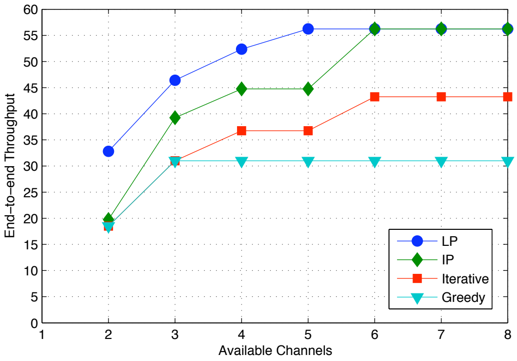


Fig. 2. End-to-end throughput for networks of size 10 with a multicast group size of 5. Each node is equipped with 2 radios and the number of available channels is varied from 2 to 8

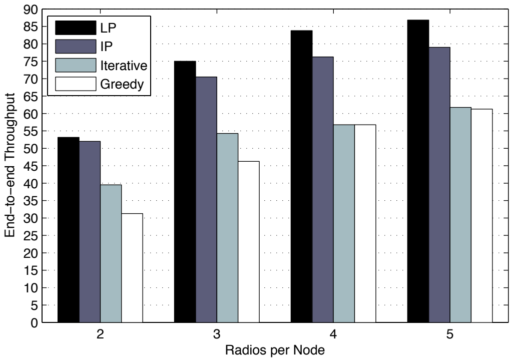


Fig. 3. End-to-end throughput as a function of the number of radios in each node for networks of size 10. The multicast group size is 5, and a total of 6 channels are available for use

Figure 2 shows the end-to-end throughput as computed by the IP, LP as well as the Greedy and Iterative solutions of Algorithm 1 and Algorithm 2 respectively for networks of size 10, with a multicast group size of 5 nodes, where each node is equipped with two radios. We varied the number of available channels and studied the effect on the computed throughput. With fewer than 2 channels, no feasible multicast flow was found, since with a single channel, multicast would only be feasible if all receivers were neighbours of the source node. As we increase the number of channels, the upper bound as computed by the LP increases, but saturates at 5 channels onward. At this point, the number of radios per node is the limiting factor and further improvement is not possible unless more radios per node are provisioned. The optimal throughput as computed by the IP shows increased throughput as the number of channels is increased, and the *true* saturation point is achieved at 6 channels onward. The Greedy algorithm is only able to improve throughput when the number of channels increases from 2 to 3, and is unable to utilize further additional channels. In contrast, the Iterative assignment scheme

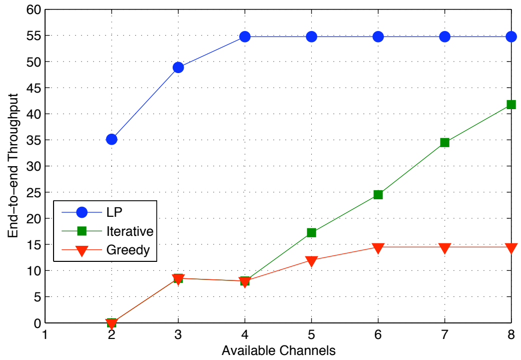


Fig. 4. End-to-end throughput for networks of size 100 with 10 multicast receivers and 2 radios per node, as a function of the number of available channels

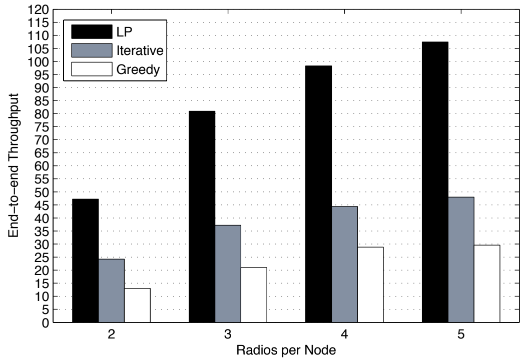


Fig. 5. Effect of the number of radios per node on end-to-end throughput for networks of size 100. The number of available channels is 8, and there are 10 multicast receivers

achieves higher throughput than the Greedy solution from 4 channels onward, and continues to increase throughput with the increased availability of channels before saturating at 6 channels onward. At this point, the Iterative algorithm achieves a throughput that is close to 75% of the optimal.

In Figure 3, we study the effect of the number of radios per node on the achieved throughput, while fixing the number of available channels at 6, on networks with 10 nodes. With a single radio, achieved throughput is 0. This is because at least two radios are needed, one to transmit and one to listen, as captured by the constraints in (14). As we increase the number of radios, the throughput computed by the IP increases. More radios lead to increased bandwidth per node as long as there are sufficient channels available for an interference-free channel assignment. At 2 and 3 radios per node, the Iterative algorithm significantly outperforms the Greedy solution. The performance of the Greedy algorithm only improves when more radios are available. This indicates that the Iterative algorithm is more efficient at utilizing available resources when there is a scarcity of radios.

In order to study the effect of available radios and channels in a larger network, we abandon the IP and focus on the end-to-end throughput as computed by the LP and the proposed algorithms. The former provides an upper bound on the optimal throughput, while we wish to study the performance of the latter in larger networks. Figure 4 shows the throughput as computed by the LP and both algorithms for networks of 100 nodes and 10 multicast receivers, with two radios per node. Similar to Figure 2, we observe an increase in the upper bound of the throughput as the number of channels increases up to a point before saturation, which occurs at 4 channels onward. Once again, we attribute this behaviour to the limited number of radios. The Greedy solution is able to increase throughput marginally up to 6 channels before saturation occurs. The Iterative assignment algorithm, on the other hand, performs significantly better than the Greedy algorithm, and is able to continually improve throughput, achieving close to 75% of the upper bound on end-to-end throughput when 8 channels are available.

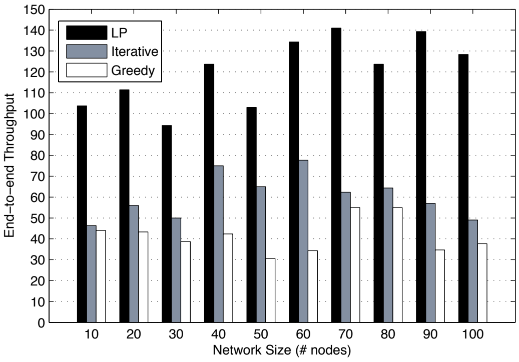


Fig. 6. End-to-end throughput for various network sizes. Each node has 4 radios and there are 8 channels available for assignment

We also examined the effect of varying the number of radios in the same network setting with 100 nodes and 10 multicast receivers. Available channels were fixed at 6, and Figure 5 shows the end-to-end throughput as computed by the LP and both algorithms. As the number of radios increases, the upper bound on achievable throughput grows. The Iterative algorithm outperforms the Greedy solution in all cases, and is able to increase throughput as the number of available radios increases. The Greedy solution is also able to increase throughput with more radios, but shows only marginal improvement when the number of radios increases from 4 to 5. Again, this shows that the Iterative algorithm computes channel assignments that are more efficient at utilizing the available radios for higher multicast throughput.

Finally, we study the effect of network size on the performance of the LP and the algorithms, as shown in Figure 6. The multicast group size is 10, each node is equipped with 4 radios and there are 8 available channels. Here, we see no noticeable trend as the network size increases. However, we note that the Iterative algorithm significantly outperforms the Greedy solution for all network sizes.

V. Further Work

Due to the NP-hard nature of the channel assignment problem, it is infeasible to compute the optimal multicast flow for very large multi-channel multi-radio networks. A little thought shows that the problem inherently lies in the discrete nature of orthogonal channels, with *non-overlapping spectrum.* The latter is a concern, since the IEEE 802.11 standard specifies 13 channels, of which only 3 do not overlap.

The above motivates a new possible solution. Instead of considering the predefined central frequencies for channels, we allow radios to tune into any frequency in some specified spectrum range. We seek to answer the following question: what is the opportunity cost of a predefined spectrum to channel allocation as opposed to a freely tunable radio in the spectrum? Specifically, taking into account interference effects, can freely tunable radios achieve better throughput in multi-radio environments? We have formulated a network model and a convex program to compute optimal multicast throughput in such a scenario.

First, we translate the network in the following fashion; we create as many copies of each node as there are radios. If node u in the original network has R radios, then in the new network we will have nodes $u_1 \dots u_R$. Then, for each neighbour node $v \in N(u)$ in the original network, we add edges between every node u_i and v_i in the new network. Further, we add edges with no capacity limit between every node u_i and u_j to model the unrestricted internal flow from one radio to another in a single node. Let this new network be the graph $G = (V, E, E^I)$ where V and E are the sets of nodes and edges respectively, while E^I is the set of internal edges between nodes. Figure 7 shows an example of the network translation model. We now state the convex program to compute optimal throughput in this network

$$\text{Maximize} \qquad d \tag{16}$$

$$\text{Subject To}$$

$$f_t(\vec{ts}) = d \qquad \forall t \in T \tag{16a}$$

$$\sum_{u \in N(v)} f_t(\vec{uv}) - \sum_{u \in N(v)} f_t(\vec{vu}) = 0 \qquad \forall v \in V, \forall t \in T \tag{16b}$$

$$\sum_{v \in N(u)} f_t(\vec{uv}) \leq f_t(u) \qquad \forall t \in T, \forall u \in V \tag{16c}$$

$$f_t(u) \leq c'(u) \qquad \forall t \in T, \forall u \in V \tag{16d}$$

$$c'(u) = f(\alpha(u), \alpha(v) \forall v \in N_I(v)) c(u) \qquad \forall u \in V \tag{16e}$$

$$K_1 \leq \alpha(u) \leq K_2 \qquad \forall u \in V \tag{16f}$$

$$f_t(\vec{uv}), f_t(u) \geq 0 \qquad \forall u \in V, \forall \ \vec{uv} \in E \tag{16g}$$

The convex program in (16) is similar to the linear program in (15), but differs in that the capacity of a node is now a function f, of the transmission frequency chosen by the node, $\alpha(u)$, and the transmission frequencies chosen by the set of nodes within interference range, $N_I(u)$. The function f represents the amount of interference caused by the choice of transmission frequencies of nodes, and has range between 0 and 1. Ko *et al.* [4] choose a convex function that depends on the amount of channel overlap between adjacent channels in their work. We believe a similar idea would work here, but we were unable to run simulations using (16) due to time constraints. Finally, note also that K_1 and K_2 in the convex program represent lower and upper bounds on the usable frequency spectrum.

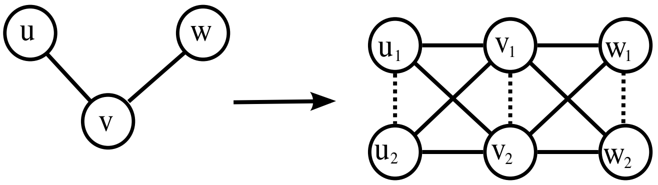


Fig. 7. An example of network translation to an equivalent network in which each node represents a radio. The links shown using dotted lines represent internal, uncapacitated links between radios within a single node in the original network. The solid lines show links between nodes in the new network that represent connectivity in the original network

There is one more possibility to explore. One can consider the linear programming relaxation of (14) and treat the fractional values of $x^i_j(u)$ and $y^i_j(u)$ obtained as an indicator of *how long* radio j should spend transmitting and receiving on channel i respectively. This raises a new algorithmic problem, given fractional values of $x^i_j(u)$ and $y^i_j(u)$, can we use this to guide a *scheduling* algorithm that assigns channels for a fraction of a fixed time slot? We believe that using this dynamic channel assignment method, we can achieve higher throughput than the static assignment we have studied throughout this paper.

VI. Conclusion

In this paper, we study multicast using network coding in wireless environments in which nodes are equipped with multiple radios, capable of using channel diversity in the form of non-overlapping transmission frequency bands. Previous work on multi-channel networks have focused on unicast or multi-session unicast. In contrast, we focus on the optimal multicast problem.

We have developed a model that is complete and optimal. Moreover, our model exploits the broadcast property in wireless networks, by employing network coded transmissions to efficiently utilize available bandwidth. The channel assignment problem, similar to colouring problems in graphs, is NP-hard, which is reflected in the integer program model derived. We have presented a simple, greedy channel assignment algorithm as well as an improved solution that performs channel assignment iteratively using the residual network of the flow computed.

The results from simulations performed have provided us with an insight into how end-to-end throughput is affected by the availability of both channels and radios per node. We find that the iterative algorithm utilizes the available radios and channels more efficiently and thereby achieves higher end-to-end throughput as compared to the greedy channel assignment scheme. We intend to pursue the work presented here further with the ultimate goals of characterizing the feasible flow region of multi-channel multi-radio wireless networks, and developing efficient algorithms to approximate the optimal flow via scheduling.

Acknowledgement

The authors would like to thank the anonymous referees for their comments. This work was supported in part by NSERC and iCore.

References

[1] A. Raniwala and T. Chiueh, "Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network," in *Proceedings of IEEE INFOCOM*, 2005.

[2] P. Bahl, A. Adya, J. Padhye, and A. Walman, "Reconsidering wireless systems with multiple radios," *SIGCOMM Computer Communication Review*, vol. 34, pp. 39–46, 2004.

[3] A. Raniwala, K. Gopalan, and T.-c. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *SIGMOBILE Mobile Computer Communications Review*, vol. 8, pp. 50–65, 2004.

[4] B.-J. Ko, V. Misra, J. Padhye, and D. Rubenstein, "Distributed channel assignment in multi-radio 802.11 mesh networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2007.

[5] M. Alicherry, R. Bhatia, and L. E. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *Proceedings of ACM MOBICOM*, 2005.

[6] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *Proceedings of ACM MOBICOM*, 2005.

[7] S. Chen, O. G*ünlük*, and B. Yener, "The Multicast Packing Problem," *IEEE/ACM Transactions on Networking*, vol. 8, no. 3, pp. 311–318, 2000.

[8] B. Li and J. Liu, "Multirate video multicast over the internet: an overview," *Network, IEEE*, vol. 17, no. 1, pp. 24–29, 2003.

[9] Z. Li and B. Li, "Efficient and Distributed Computation of Maximum Multicast Rates," in *Proceedings of IEEE INFOCOM*, 2005.

[10] D. S. Lun, N. Ratnakar, R. Koetter, M. M*é*dard, E. Ahmed, and H. Lee, "Achieving Minimum-cost Multicast: a Decentralized Approach Based on Network Coding," in *Proceedings of IEEE INFOCOM*, 2005.

[11] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[12] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 5, pp. 782–795, 2003.

[13] J. Feigenbaum, C. Papadimitriou, and S. Shenker, "Sharing the Cost of Multicast Transmissions," *Journal of Computer and System Sciences*, vol. 63, pp. 21–41, 2001.

[14] W. Wang, X.-Y. Li, and Z. Sun, "Sharing the Multicast Payment Fairly," in *Proceedings of the 11th International Computing and Combinatorics Conference (COCOON)*, 2005.

[15] N. Garg, R. Khandekar, K. Kunal, and V. Pandit, "Bandwidth Maximization in Multicasting," in *Proceedings of the 11th European Symposium on Algorithms (ESA)*, 2003.

[16] K. Jain, M. Mahdian, and M. R. Salavatipour, "Packing Steiner Trees," in *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2003.

[17] M. Thimm, "On The Approximability Of The Steiner Tree Problem," in *Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science*, 2001.

[18] K. Jain, "A factor 2 approximation algorithm for the generalized steiner network problem," *Combinatorica*, vol. 21, pp. 39–60, 2001. [Online]. Available: http://dx.doi.org/10.1007/s004930170004

[19] M. X. Goemans and D. P. Williamson, "A general approximation technique for constrained forest problems," in *Proceedings of the 3rd annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, 1992, pp. 307–316.

[20] A. Z. Zelikovsky, "An 11/6-approximation algorithm for the network steiner problem," *Algorithmica*, vol. 9, pp. 463–470, May 1993. [Online]. Available: http://dx.doi.org/10.1007/BF01187035

[21] Z. Li, B. Li, and L. C. Lau, "On Achieving Optimal Multicast Throughput in Undirected Networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2467–2485, June 2006.

[22] T. Liu and W. Liao, "On routing in multichannel wireless mesh networks: Challenges and solutions," *IEEE Network*, vol. 22, pp. 13–18, 2008.

[23] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, pp. 388–404, 2000.

[24] P. Kyasanur and N. H. Vaidya, "Capacity of multi-channel wireless networks: impact of number of channels and interfaces," in *Proceedings of ACM MOBICOM*, 2005.

[25] A. Brzezinski, G. Zussman, and E. Modiano, "Enabling distributed throughput maximization in wireless mesh networks: a partitioning approach," in *Proceedings of ACM MOBICOM*, 2006.

[26] A. Dimakis and J. Walrand, "Sufficient conditions for stability of longest queue first scheduling: second order properties using fluid limits," *Advances of Applied Probability*, vol. 38, pp. 505–521, 2006.

[27] K. Xing, X. Cheng, L. Ma, and Q. Liang, "Superimposed code based channel assignment in multi-radio multi-channel wireless mesh networks," in *Proceedings of ACM MOBICOM*, 2007.

[28] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of ACM MOBICOM*, 2003.

[29] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of ACM MOBICOM*, 2004.

[30] B. O'Hara and A. Petrick, *The IEEE 802.11 Handbook: A Designer's Companion*. IEEE Press, 1999.

[31] "GNU Linear Programming Kit," http://www.gnu.org/software/glpk/.

[32] G. B. Dantzig, *Linear Programming and Extensions*. Princeton University Press, 1998.