This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2007 proceedings.

Toward tractable computation of the capacity of multihop wireless networks

Stephan Bohacek * Peng Wang †
bohacek@udel.edu pwangee@udel.edu
Department of Electrical and Computer Engineering
University of Delaware

***Abstract*— By posing the problem of bandwidth allocation as a constrained maximization problem, it is possible to study various features of optimal bandwidth allocation, and hence the capacity of the network. However, since the typical approach to this problem requires optimizing over a space that is exponential in the number of links, the problem has appeared to be computationally intractable for all but small networks. In this paper, the problem of computing optimal bandwidth allocation is examined and a new approach is presented. While the resulting allocation cannot be guaranteed to be optimal, we find that in the networks where checking optimality is computationally feasible (i.e., networks with fewer than 23 links), the performance of the found allocation is indistinguishable from the optimal allocation. In essence, the proposed iterative scheme focuses on the space of useful bandwidth allocations. The Lagrange multipliers are used to find useful allocations.**

I. Introduction

The capacity of multihop wireless networks has been the subject of intense research ([1], [2], [3], [4], [5]). There is extensive motivation for this research. For example, in the mesh network setting, determining capacity is useful for network planning. Also, routing and capacity are closely related, and hence, the performance of routing protocols can be greatly improved if the capacity that results from a particular routing can be determined. The optimal capacity provides an upper bound on heuristic techniques, and provides a means to judge the quality of the heuristic. Also, there is theoretical interest in capacity; for example, there has been interest in how the capacity varies as a function of the number of nodes [6].

Generally, there have been two approaches to maximize the capacity of a wireless network, namely, a heuristic-based approach and an optimization-based approach. In the optimization-based approach, which is the focus of this paper, an objective function is defined and this objective is maximized subject to the constraints imposed by interference. For example, it is common to define an objective function to be a concave increasing function of flow rates; this function is often referred to as a utility function and it can be shown that maximizing such functions results in fairness among flows ([7], [8], [5]). This paper focuses on this class of problems. Other, non-utility-based approaches include [3], which minimizes a linear function of transmission powers subject to a link bit-rate constraint and other constraints imposed by interference. A different capacity problem is related to maximizing an objective function when the traffic demands are stochastic [9].

A critical problem with the optimization-based approaches is that the resulting problem is computationally difficult, if not intractable. More specifically, in order to maximize capacity, the optimal allocation of resources must be determined. This allocation is defined by a schedule that dictates when links transmit and at which frequencies and powers they transmit. However, the space of schedules is extremely large. For example, if power control is not used, then one must maximize over a polytope with 2^L extreme points, where L is the number of links. More specifically, we define an *assignment* to be a specification of which links are transmitting and which links are not transmitting. A schedule is the convex sum of assignments; hence the assignments are the extreme points of the space of schedules. If power control is not used, then the space of assignments contains 2^L elements. If power control is used, then an assignment specifies not only which links are transmitting, but also the transmission power. In this case[1], the space of assignments is $[0,1]^L$. If one attempts to approximate this continuous space with a discrete grid such as $\{0, 0.33, 0.67, 1\}^L$, then the space has 4^L elements. To comprehend the size of 2^L, consider a network with only 88 links, which is far smaller than the mesh networks being deployed throughout the world. However, 2^{88} *nsec* is the approximate age of the universe. Hence, it is intractable to even initialize the problem. Due to the exponential dependence on the number of links, after considerable reduction of the problem, [3] reports being unable to compute the optimal schedule for a network with more than 15 links.

This paper focuses on the optimization-based approach, however, in order to make the problem tractable, the size of the space of assignments is reduced. Through exhaustive search, we find that if power control is not permitted[2], then the proposed technique determine schedules that are either

*This work was prepared through collaborative participation in the Collaborative Technology Alliance for Communications and Networks sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAD19-01-2-0011. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

†This work was support in part by NSF grant ANI-0312851.

[1]It is important to note that in some scenarios, it can be proved that power control is not required. Hence, if one considers power control, the space over which the optimization is performed still only has 2^L elements. See [3] for details.

[2]Exhaustive search is not tractable when power control is permitted.

Algorithm 1 Estimate of network capacity

0: Select an initial set of assignments to consider.
1: Given a set of considered assignments, optimize the utility.
2: Use the Lagrange multipliers found in solving the optimization in Step 1 to form a linear test.
3: Search for an assignment that satisfies this linear test.
if an assignment is found **then**
add it to the set of considered assignments and go to Step4.
else
if no assignment *exists* that satisfies the linear constraints, then stop, the optimal solution has been found.
end if
4: Remove any redundant assignments within the space of considered assignments. Then go to Step 1.

optimal, or nearly optimal (e.g., within 0.5% of optimal).

There are two key theoretical results that underpin this approach.

- While the optimization may be performed over a space with 2^L assignments (if power control is used, the space is $[0,1]^L$), the optimal solution is the combination of no more than L assignments. Therefore, if these L assignments were somehow known in advance, then the optimization could be performed over a space with L assignments and the result would be identical to the one found by optimizing over the space of all assignments.
- Guided by this result, we focus on searching for these special L assignments. This search is greatly aided by a linear test that the unknown special assignments satisfy. This linear test is a by-product of performing the optimization over an arbitrary set of assignments.

Algorithm 1 implements these ideas.

An important contribution of this paper is that the central challenge of the problem of computing the optimal schedule is not the optimization, which is performed in Step 1, rather, the main difficulty is Step 3, finding an assignment that passes the test constructed in Step 2.

One approach to Step 3 is to perform an exhaustive search. If power control is not used and the network is relatively small, then exhaustive search can be performed. In this case, the lack of existence of an assignment that passes the linear test found in Step 2 can be verified, and hence it is possible to guarantee the optimality of the schedule, and hence the capacity of the network can be determined. However, such an approach is only tractable when power control is not used and when the network is relatively small (e.g., less than 23 links). In this case, some other search methods must be used.

This paper presents several methods to search for assignments that pass the linear test. One approach reduces to searching for the maximum weighted stable set (MWSS) on an adapted contention graph. To estimate the MWSS, an algorithm developed by Kako et al [10] is employed. Along with searching for a MWSS, other techniques are used to search for an assignment. Section VII examines the performance of these search methods when an exhaustive search is tractable. It is found that if the proposed search techniques fail to find an assignment, then either the exhaustive search fails to find an assignment (i.e., the schedule is optimal), or the assignments found through the exhaustive search have only a marginal impact on the performance. Thus, the techniques presented here constitute a tractable algorithm that finds the optimal or near optimal schedule.

The remainder of the paper proceeds as follows. In the next section, the system model and notation are presented. The optimization performed in Step 1 of Algorithm 1 is discussed in Section III and the computational details are explained in [11]. The details and justification for the linear test are explained in Section IV. Section V describes methods to search for new assignments. Section VII presents the results of numerical experiments. We note that these experiments are based on ray-tracing based propagation models of urban mesh networks [12], hence, the results are applicable to the mesh networks currently being deployed. Finally, Section VIII provides some concluding remarks.

This paper is focused on computing capacity. The approach discussed is centralized. Decentralized version of the computation is outside the scope of this paper.

An implementation of the algorithms presented is available online [13]. Due to space limitations, proofs and some algorithmic details cannot be included in this version, but can be found in the extended version [11].

II. Notation and System Model

Let f_ϕ be the flow rate of flow ϕ. The techniques presented can easily be extended to the case where the flow ϕ is divided among several paths, each with rate f_ϕ^k. However, to reduce notational clutter, we focus on the single path case. Let w_ϕ be the utility weight for flow ϕ. The objective considered in this paper is to maximize the utility, $\sum_{\phi\in\Phi} w_\phi U(f_\phi)$, where Φ is the set of all flows, and U is a concave increasing utility function. While a wide range of functions meet these conditions, to make the presentation more concrete, we will focus on $U(f) = \log(f)$, but this approach can be extended to any smooth concave increasing U. Let $\mathcal{P}(\phi)$ denote the set of links that flow ϕ traverses. Hence, to support flow rates $\{f_\phi\}$, the bit-rates for link l must exceed $\sum_{\{\phi: l\in\mathcal{P}(\phi)\}} f_\phi$.

Define an *assignment* to be a vector $v = \begin{bmatrix} v_1 & \cdots & v_L \end{bmatrix}$, where $v_l \in [0,1]$, indicating the transmission power of link l and $v_l = 0$ implies that link l is not transmitting, and there are L links in the network. If power control is not permitted, then $v_l \in \{0,1\}$. For a particular assignment, we get a vector of *link bit-rates* r, via,

$$r_l = \log_2\left(1 + \frac{H_{l,l}v_l}{\sum_{k\neq l} H_{k,l}v_k + \mathcal{N}}\right), \qquad (1)$$

where $\mathcal{N}$ is the thermal noise and $H_{k,l}$ is the normalized channel gain from the transmitter of link k to the receiver of link l. Hence, $H_{l,l}$ is the channel gain over link l, and

$H_{j,l}v_j$ is the strength of the interference from the transmitter of link j to the receiver of link l. H is normalized so that the full transmission power is achieved by $v_l = 1$. Due to the close connection between an assignment and the link bit-rates that result from the assignment, we refer to both v and r as assignments with the implicit understanding that r is the set of link bit-rates that arise from the assignment v. Similarly, if V is a set of assignments, we let R denote the set of vectors of link bit-rates that result from the assignments in V. Specifically, we denote the bit-rate of link l when assignment v is used as $R(v,l)$.

While it is common to use TDM scheduling, here a *FDM* scheme is used. As in the TDM case, a FDM schedule is a convex sum of assignments. Specifically, the total bandwidth is divided into slices and each assignment used is allocated a slice of bandwidth. The width of the bandwidth slice allocated to an assignment depends on the schedule. For example, let r be the vector of link rates when the assignment v is used, and s be the link rates when the assignment w is used. Then, $0.5r + 0.5s$ is the bandwidth assignment where the assignment v is applied to the lower half of the bandwidth, and w is applied to the upper half of the bandwidth. To enhance succinctness, we may sometimes refer only to the assignments, hence, a schedule may be written $0.5v + 0.5w$. Specifically, letting V be a set of assignments under consideration. Then a schedule may be written as $\sum_{v\in V}\alpha_v v$ with $\sum_{v\in\mathcal{V}}\alpha_v = 1$ and $\alpha_r \geq 0$. In this case, the vector of link bit-rates achieved is $\sum_{v\in V}\alpha_v R(v,:)$, where we use the notation $R(v,:)$ to denote the row vector of link bit-rates that result from using the assignment v.

We do not allow assignments to have a node simultaneously transmitting and receiving on the same slice of bandwidth. However, by scheduling multiple assignments, we permit a node to simultaneously transmit and receive over different slices of bandwidth. Note that 802.16 also allows nodes to simultaneously transmit and receive over different slices of bandwidth. Also, a practical implementation of a FDM schedule would require a multiple carrier approach; in fact, many physical layers employ multiple carriers.

III. Problem Formulation and Lagrange Multiplier-Based Solution

The techniques developed are based on Lagrange Multiplier Theory (e.g., see [14]). While Lagrange Multiplier Theory for optimizing networks has been extensively examined in the literature (e.g., [7], [15], [16]), the techniques developed here are based on insights into Lagrange Multiplier Theory, and hence, further examination is required. Specifically, as mentioned above, the Lagrange Multipliers form a linear constraint on the set of assignments that will increase the utility. On the other hand, this problem has been well studied and is relatively straightforward, so we neglect issues related to existence of a unique solution, regularity, etc.

As mentioned above, and as is common throughout the research literature, we focus on maximizing the utility. Specifically, we seek to solve

$$\begin{aligned} &\min -\sum_{\phi\in\Phi} w_\phi \log(f_\phi) \qquad (2)\\ &\text{subject to: } \sum_{\{\phi: l\in\mathcal{P}(\phi)\}} f_\phi \leq \sum_{v\in V}\alpha_v R(v,l) \text{ for all } l\\ &\qquad\qquad \sum_{v\in V}\alpha_v = 1,\ \alpha_v \geq 0. \end{aligned}$$

In order to solve this problem, define the Lagrange function

$$\begin{aligned} \mathcal{L}(f,\alpha,\mu,\lambda) = &-\sum_{\phi\in\Phi} w_\phi \log(f_\phi) + \lambda\left(\sum_{v\in V}\alpha_v - 1\right)\\ &+\sum_{l=1}^{L}\mu_l\left(\sum_{\{\phi: l\in\mathcal{P}(\phi)\}} f_\phi - \sum_{v\in V}\alpha_r R(v,l)\right). \qquad (3) \end{aligned}$$

After some manipulations, the dual function is found to be

$$\begin{aligned} q(\mu,\lambda) = &\inf_{f,\alpha\geq 0} -\sum_{\phi\in\Phi}\log(f_\phi) w_\phi - \lambda\\ &+\sum_{l=1}^{L}\mu_l\sum_{\{\phi: l\in\mathcal{P}(\phi)\}} f_\phi - \sum_{v\in V}\alpha_v\left(\sum_{l=1}^{L} R(v,l)\mu_l - \lambda\right). \end{aligned}$$

We immediately note that if $\sum_{l=1}^{L} R(v,l)\mu_l - \lambda > 0$ for some v, then $q(\mu,\lambda) = -\infty$. Hence, we restrict the domain of q, to be such that $\sum_{l=1}^{L} R(v,l)\mu_l - \lambda \leq 0$. On the other hand, when solving the dual problem, an objective is to maximize q with respect to λ. It is not hard to see that this is equivalent to minimizing λ over the domain $\sum_{l=1}^{L} R(v,l)\mu_l - \lambda \leq 0$. Thus,

$$\lambda^* = \max_{v\in V}\sum_{l=1}^{L} R(v,l)\mu_l. \qquad (4)$$

Therefore, we can rewrite the dual function as[3],

$$\begin{aligned} q(\mu) = &\inf_{f\geq 0} -\sum_{\phi\in\Phi}\log(f_\phi) w_\phi \qquad (5)\\ &+\sum_{l=1}^{L}\mu_l\sum_{\{\phi: l\in\mathcal{P}(\phi)\}} f_\phi - \max_{v\in V}\sum_{l=1}^{L} R(v,l)\mu_l, \end{aligned}$$

where α_v has been eliminated since λ^* results in the infimum being achieved for $\alpha_v = 0$.

We denote μ^* as the solution to the dual problem, i.e.,

$$\mu^* = \arg\max_{\mu\geq 0} q(\mu). \qquad (6)$$

The problem of computing μ^* is addressed in [11]. Briefly, while the term $\max_{veV}\sum_{l=1}^{L} R(v,l)\mu_l$ results in q being undifferentiable for some μ, subgradient techniques can be employed. Specifically, the dilation technique [17], which is similar to the conjugate gradient approach, can be used to efficiently compute μ^*. (An implementation is available online [13]). Therefore, as mentioned above, the central challenge is

[3]There are other, more straightforward ways to arrive at (5). However, these methods do not provide the important expression (4).

not to solve (6), but to reduce the size of V. Before examining this problem, a few observations can be made.

It is important to note that there may be several assignments $v \in V$ that achieve $\max_{v\in V} \sum_{l=1}^{L} R(v,l)\,\mu_l$. Define $V^*(\mu)$ be the set of such assignments, i.e.,

$$V^*(\mu) := \left\{ v : \sum_{l=1}^{L} R(v,l)\,\mu_l = \max_{v\in V} \sum_{l=1}^{L} R(v,l)\,\mu_l \right\}. \quad (7)$$

Applying the economic interpretation of the Lagrange framework, μ is the link price per bit, hence, $\sum_{\{l:v_l=1\}} R(v,l)\,\mu_l$ is the revenue generated by the assignment v. In order to maximize revenue, only assignments $v \in V^*(\mu)$ are utilized. At optimality, $V^*(\mu^*)$ typically contains several assignments. We refer to these assignments as *active assignments*. Next, we show that the active assignments are important for solving the optimization (2). Later, we use the active assignments to search for a better assignment.

Proposition 1: Given μ^*, the link costs that solve the dual problem, the optimal schedule can be found by solving the following linear programming problem

$$\begin{aligned} \min \sum_{v\in V^*} \alpha_v \quad (8)\\ \text{subject to: } \sum_{\{\phi: l\in\mathcal{P}(\phi)\}} \frac{w_\phi}{\sum_{\{m:m\in\mathcal{P}(\phi)\}} \mu_m^*} \\ \leq \sum_{v\in V^*(\mu^*)} \alpha_v R(v,l) \text{ for each } l \\ \alpha_v \geq 0 \end{aligned}$$

Remark 2: We will soon see that V^* has no more than L elements. Hence, (8) is a low dimensional linear programming problem.

IV. Considered Assignments

A. Introduction

The problem above suffers from the drawback that the set of all assignments, which we denote by V^{**}, has 2^L elements if power control is not permitted, and is $[0,1]^L$ if power control is allowed. While obvious assignments can be eliminated (e.g., a node simultaneously receiving and transmitting on the same slice of bandwidth), the size of the set of all reasonable assignments is very large and results in the problem being intractable. One approach is to reduce the size of V. Indeed, under a slightly different problem formulation [18] follows this approach. We refer to this reduced set of assignments as the set of *considered assignments*. The two key questions are 1. what is the impact on the value of (2) when a reduced set of considered assignments is used, and 2. how can the set of considered assignments be constructed so that the value of (2) with the reduced sized V is the same as or is near to the value when V contains all possible assignments? The answer to the first question is provided next, and much of the rest of the paper is devoted to this second question.

Theorem 3: There exists $\tilde{V}$ with L assignments such that the solution to (2) with V replaced with $\tilde{V}$ yields the same solution if V is replaced with the V^{**}.

This result, which follows from Caratheodory's Theorem (e.g., Theorem B.6 in [14]), implies that the optimal schedule can be found by considering a set, $\tilde{V}$, that is relatively low small. To emphasis the reduction in the size of V, consider $L = 88$, then $\#V^{**} = 2^{88}$, and $\#\tilde{V} = 88$, where the first problem is obviously intractable, while the second can be solved with limited computational effort. Clearly, the key to computing the optimal schedule is determining the set $\tilde{V}$.

B. Evaluating Candidate Assignments

A brute force approach to constructing a good set of assignments is to start with a particular set of assignments, V, select an assignment $v^+ \notin V$, and evaluate the resulting utility with the set of assignments $v^+ \cup V$. However, this approach is computationally complex in that (2) must be repeatedly solved. Furthermore, it is not clear if the utility of v^+ is only apparent when it is added to V along with a particular set of other assignments. Next a technique is provided to efficiently determine whether an assignment should be added to the set of considered assignments. Other aspects of determining a good set V include selecting the initial set V^0, selecting which $v^+ \notin V$ to consider, and removing assignments from V. These issues are addressed in sections VI, V, and IV-C, respectively.

The question of whether an assignment $v^+ \notin V$ will increase the utility when the set of considered assignments is changed from V to $v^+ \cup V$ is answered by the following theorem.

Theorem 4: For the set of assignments V, let μ^* and λ^* be the optimal values of the multipliers when (2) is solved with this V. Now consider an assignment $v^+ \notin V$, where the link bit-rates provided by v^+ is r^+ (by applying (1)). Then, the utility provided by $v^+ \cup V$ is greater than that provided by V if

$$r^+\mu^* - \lambda^* > 0. \quad (9)$$

Corollary 5: If there is no assignment in V^{**} such that $r^+\mu^* - \lambda^* > 0$, then $\tilde{V} \subset V$, that is, the optimal schedule has been found.

Corollary 6: Let μ and λ denote the multipliers found when solving (2) with the set of considered assignments V, and suppose that V^*, which is defined by (7), has exactly L elements and is such that no element is in the convex hull of the other elements. Then the optimal schedule, r^{**}, satisfies $r^{**}\mu \geq \lambda$.

Theorem 4 provides the main tool for constructing a good set of assignments. Figure 1 provides a geometric view of Theorem 4. Several comments are in order.

- Under a slightly different problem formulation, [18] considered arbitrarily adding assignments to the set of considered assignments. Theorem 4 provides a more sophisticated technique to decide whether assignments should be added.
- In the proof of Theorem 4, it is seen that a linear approximation of the improvement of utility is given by $r^+\mu^* - \lambda^*$. Hence, the larger $r^+\mu^* - \lambda^*$, the larger the expected improvement in the utility. Therefore, if several suitable assignments are found, it is reasonable

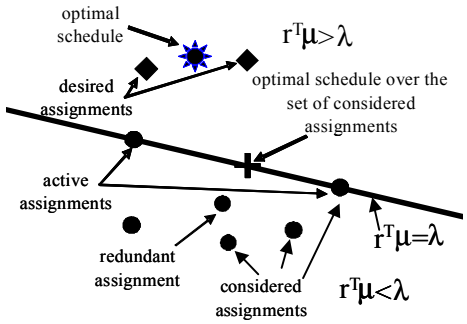

Fig. 1. A geometric view of the optimal scheduling problem. The above shows the region of bit-rates, where we assume that there are only two links, and hence the space of bit-rates is the plane. The Lagrange multipliers found from optimizing over the set of considered assignments divides the space of bit-rates, r, into two regions, according to whether $r^T\mu < \lambda$ or $r^T\mu > \lambda$. The active assignments and the schedule found by optimizing over the considered assignments are on the boundary of this division. An assignment will only improve the performance if $r^T\mu > \lambda$. The goal is to find the two desired assignments. The optimal schedule is a convex combination of these assignments.

to only add the assignment that maximizes $r^+\mu^* - \lambda^*$. Through numerical experiments, Section VII confirms this relationship between $r^+\mu^* - \lambda^*$ and the improvement in utility.

- Corollary 5 provides a means to determine whether the current schedule is optimal. However, its practical implication is moderate since verifying that (9) does not hold for any assignment in V^{**} requires iterating through all possible assignments, which, if power control is not used, has complexity $O\left(2^L\right)$. On the other hand, evaluating (9) can be performed quickly, and hence an exhaustive search of V^{**} for moderate sized networks (e.g., with fewer than 23 links) is feasible. Also, it is far easier to evaluate (9) 2^L times than it is to solve (2) with V containing all 2^L elements. Note that if L is 15, it takes about two minutes to evaluate (9) 2^L times, whereas, under a slightly different problem formulation, [3] developed a method that was unable to compute the optimal schedule for a network with more than 15 links.
- Step 2 in Algorithm 1 is now justified.

C. Removing Assignments from the Set of Considered Assignments

The chief goal of this approach is to solve problem (2), where V is a small set of assignments. The motivation of this is that if V is small, then (2) can be solved quickly. However, as more assignments are added to V, then its size will grow, defeating this goal. Thus, it is useful to remove assignments if they are guaranteed to never be used as active assignments. To see how this is done, suppose that V, the set of considered assignments, yields a set of considered link bit-rates, R, and let r be an element in R. Then, to see if the assignment r can be removed from R, we check whether

$$r \in \text{interior of } Co\left(R \backslash r\right), \tag{10}$$

where $Co\left(R \backslash r\right)$ is the convex hull of $R \backslash r$ and $R \backslash r$ is the set R with r removed. If (10) holds, then the link bit-rates achieved by r can also be achieved by using a set of bit-rates in $R \backslash r$. Hence, the assignment r can be removed without impacting $Co\left(R\right)$. Figure 1 illustrates a redundant assignment that can be removed.

From Theorem 3, we know that the optimal vector of bit-rates is the convex sum of no more than L assignments. Similarly, the vector of bit-rates that solves (2) for any set of assignments is also the convex sum of no more than L assignments. Therefore, the set of active assignments, $V^*\left(\mu^*\right)$, should contain no more than L elements. If $V^*\left(\mu^*\right)$ does contain more than L elements, then there must be some redundancies in the set of considered assignments, and hence some assignments can be removed. Hence, Step 4 in Algorithm 1 checks whether $\#V^*\left(\mu^*\right) > L$ or if more than 10 assignments have been added to V since the last time a check for redundant assignments was performed. If either of these conditions is true, a check of redundancy in the considered assignments is performed.

Determining if (10) is true for some $r \in R$ can be determined by solving a linear programming problem. Hence, a check for redundancy requires solving $\#V$ LP problems. See [13] for details.

V. Constructing the Set of Considered Assignments

Theorem 4 shows how to decide whether the addition of an assignment into the set of considered assignments might improve the performance. Unless the network is relatively small, it is not possible to iterate through all possible assignments in search of one that obeys (9). Here, several techniques are described that search for assignments that satisfy (9). These techniques cannot guarantee to find the best assignment, and hence, if they fail to find any assignment that improves the performance, it does not imply that the current schedule is optimal. Section VII examines the performance of these methods.

A. The Case Where Power Control is Not Permitted

1) Maximum Weighted Stable Set and an Adapted Contention Graph : In [18], it was shown how contention due to interference could be represented with a contention graph. Here, the contention graph is extended to support multiple bit-rates. Then, it is shown that finding an assignment that improves the performance is the same as finding the maximum weighted stable set (see below for a definition) over the contention graph. We apply a well known algorithm for estimating the maximum weighted stable set. However, there are a large number of techniques to approximate, or perhaps, solve the maximum weighted stable set problem (e.g., see [19] and references therein).

a) Weighted Adapted Contention Graph: A contention graph is formed by representing a link that transmits at a particular bit-rate with a vertex. From (1), it is clear that in order to achieve a particular bit-rate we require that the noise and interference must be small enough. In fact, even if there is no interference, due to thermal noise, some bit-rates are not

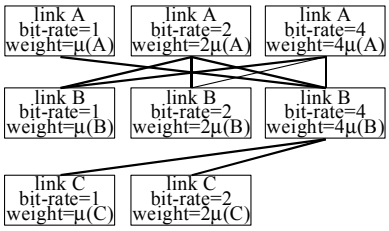

Fig. 2. Contention graph. Link A is unable to transmit at 2 or 4 bps if link B is transmitting. However, link A may transmit at 1 bps if link B is transmitting. On the other hand, link B cannot transmit at 4 bps if link A is transmitting or if link C is transmitting. Link C is unable to transmit at 4 bps even if no other links are transmitting, so this vertex is not included. The edges between vertices representing the same link are not shown.

feasible. A bit-rate that cannot be achieved even when no other links are transmitting is not included in the contention graph. On the other hand, some bit-rates can be achieved only if some links are not transmitting. For example, if link A is unable to transmit at 4 bps if link B is transmitting, then edges are drawn between the vertex that represents link A transmitting at 4 bps and the vertices that represent link B transmitting at *any bit-rate*. See Figure 2 for an example. Since we assume links are not able to transmit simultaneously at two bit-rates over the same bandwidth, there are edges between all vertices that represent the same link. To reduce clutter, these edges are not shown in Figure 2.

Referring to Figure 2, link A is able to transmit at 1 bps even if link B or link C is transmitting. What is not captured by the contention graph is the possibility that link A is not be able to transmit at 1 bps if *both* links B and C are transmitting. The adapted contention graph is better at capturing this effect. An *adapted contention graph* is adapted to a given assignment. The adapted contention graph is made in the same way as the non-adapted version, but the bit-rates that a link can achieve take into account the links that are transmitting in the given assignment. For example, if link C is selected to be transmitting in the given assignment, and link A is not able to transmit at 1 bps if *both* B and C are transmit, then an edge is drawn between link A at 1 bps and link B at all bit-rates.

Finally, a weighted adapted contention graph applies the weight $\mu(l) \times b$ to the vertex that represents link l transmitting at b bps, where μ is given by (6).

In order to construct a contention graph, a discrete set of link bit-rates must be selected. This is done as follows. Given the set of considered assignments, determine the range of bit-rates at which a particular link transmits. The contention graph is formed by including a vertex for each of these bit-rates, as well as for these bit-rates divided by two and multiplied by two. However, if noise or interference from the given assignment results in some bit-rates not being feasible, the bit-rate is reduced to a value that is feasible.

b) Maximum Weighted Stable Set and Kako's Algorithm: An assignment is a selection of links that transmit simultaneously. A link bit-rate can be achieved if the vertex representing the bit-rate does not have any neighbors in the contention graph that are also transmitting. Therefore, an assignment and the bit-rates achieved by the assignment can be represented by a selection of vertices such that no vertex in the selection is a neighbor of another vertex in the selection. Such a selection is known as a stable set or an independent set. Now, the revenue generated by the assignment/bit-rate combination is given by $\sum_{l=1}^{L} r(l)\,\mu(l)$ which is the sum of the vertex weights of the selected vertices. Therefore, the goal of finding a good assignment is the same as finding a stable set with maximum total weight. This problem is the maximum weighted stable set problem.

Extensive research has focused on the maximum stable set, and its dual, the maximum clique. In general these problems are NP-hard, however, in specialized cases (e.g., perfect graphs) they can be solved in polynomial time [20]. Here we apply a technique to find a good weighted stable set developed by Kako et al [10]. Kako's algorithm is a greedy algorithm and is as follows.

For each non-selected vertex, compute the weighted degree, which is defined as the sum of a vertex's neighbors' weights divided by the vertex's weight. Hence, a vertex has a small weighted degree if its weight is large and it has only a few neighbors and/or its neighbors' weights are small. Once the weighted degrees are computed, the vertex with the smallest weighted degree is selected, and all of the selected vertex's neighbors are removed from the graph. The process is repeated until there are no vertices left in the graph. The performance of Kako's method depends on the inductiveness of the graph, see [10] for details.

We propose two ways to use Kako's algorithm. First, we form a contention graph and apply Kako's algorithm, we refer to this method as *Simple Kako*. Alternatively, once a vertex is selected, the adapted contention graph is reformed, reflecting the impact that the selected links have on the other link bit-rates. We refer to this method as *Adapted Kako*. Section VII shows that Adapted Kako provides the best performance.

2) Modifying Existing Assignments: A straightforward way to construct new assignments is to make small changes to a currently considered assignment. For example, for each assignment, the state of a single link could be changed (i.e., the link could be switched from transmitting to not transmitting or vice versa). Rule (9) can be applied to determine whether the new assignment provides better performance. One could also consider allowing the state of two or more links to be changed. However, the performance of such extensions was not investigated. The reader is referred to [13] for further details on modifying the existing assignments.

B. The Case Where Power Control is Permitted

When power control is allowed, a link is not merely turned on or off, but its transmission power may vary. Nonetheless, the techniques described above can be extended to the case where power control is permitted. Specifically, an adapted contention graph can be constructed in a fashion similar to what is described in Section V-A.1. The only difference is that a vertex represents a link transmitting at a particular bit-rate and with a particular transmission power. Figure 3 illustrates

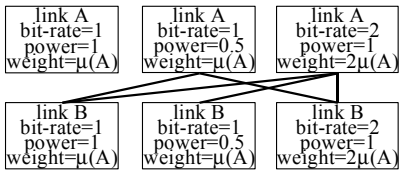

Fig. 3. Contention graph with power control.

this graph. In Section V-A.1, it was noted that the bit-rates that are included in the contention graph are related to the bit-rates that are found in the set of currently considered assignments. We take the same approach with transmission powers, i.e., we include vertices that represent transmission powers that are twice and one half of the transmission powers found in the set of currently considered assignments[4]. Kako's algorithm can then be applied to this contention graph as described above.

It is also possible to modify the assignments found in the currently considered set of assignments. Specifically, note that (9), the rule to determine whether an assignment will improve the utility, can be written as $G(v) > \lambda$ where

$$G(v) := \sum_{l=1}^{L} \mu(l) \log_2 \left(1 + \frac{H(l,l)\, v(l)}{\sum_{k \neq l} H(k,l)\, v(k) + N} \right),$$

and where $v(l)$ is the transmission power of link l. Hence, a new assignment can be found by maximizing G, and if $\max_{v \in [0,1]^L} G(v) = \lambda$, then the optimal schedule has been found. Unfortunately, G is difficult to maximize; it has many local maximums. Nonetheless, by selecting an initial condition, v_0, one can apply steepest ascent to find a local maximum of G. Therefore, the performance of an assignment can be improved by using this assignment as an initial condition to steepest ascent.

Since active assignments provide the best performance (i.e., for these assignments $r\mu = \lambda$), it is sensible to maximize G by using an active assignments as the initial condition. If the maximization fails to find a better assignment (i.e., the active assignment is a local maximum of G), then all considered assignments can be used as initial conditions for the maximization.

At the expense of exponential computational complexity, a more thorough search is achieved by trying each of the 2^L initial conditions v_0 that has each link transmitting at either full power or not transmitting at all. While such a search is more thorough, unlike the case where transmission power is fixed, we cannot guarantee that if such a search fails to produce a better performing assignment, then the current schedule is optimal. Clearly, further work in this area is required.

VI. Initial Assignments and Pair-Wise Compatibility

Through numerical experiments, we have found that the initial set of considered assignments has a significant impact on the number of iterations until no new assignments can be found. Two types of initial assignments are used. First, assignments that have a single link transmitting are included. Second, assignments composed of links such at any pair of links that are turned on in the assignment are *pair-wise compatible*.

To understand this pair-wise compatibility, consider two links, a and b. Let R_a^a be the bit-rate across link a when only link a is transmitting, and let $R_a^{a,b}$ be the bit-rate across link a when both links are transmitting. We say that links a and b are pair-wise incompatible if there exists an $0 \leq \alpha \leq 1$ such that $R_a^{a,b} < \alpha R_a^a$ and $R_b^{a,b} < (1-\alpha) R_b^b$, that is, the same bit-rate that is achieved by a and b simultaneously transmitting can be achieved by correctly multiplexing between the two links individually transmitting. Hence, the assignment where a and b simultaneously transmit should never be consider. Interestingly, the pair-wise incompatibility of a, b and c does not imply that the assignment where all three links transmit simultaneously should not be considered. Hence, pair-wise compatibility only provides a partial view of compatibility.

The second set of initial assignments is constructed in a greedy way such that all links that are turned on in an assignment are pair-wise compatible. Details of the greedy selection do not seem to impact the quality of the initial set of assignments, and hence for details on the algorithm, the reader is referred to [13].

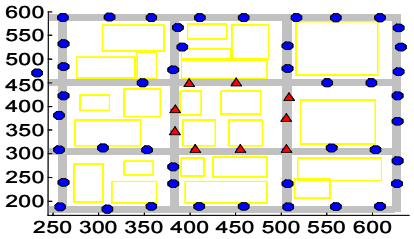

Fig. 4. Nine block region of Downtown Chicago. The base stations are displayed as triangles and the mesh nodes are circles. The buildings are shown as rectangles.

VII. Numerical Experiments

In order to gauge the performance of the proposed algorithm, we examine the utility of a part of an urban mesh network. Specifically, Figure 4 shows the 9 block region of downtown Chicago. The mesh nodes are indicated in the figure. Ray tracing was performed to determine the channel gain between the mesh nodes (See [12] for details). In order to examine the performance, various randomly selected networks were generated. In each case, there was a single base station that was selected at random from the base stations marked with triangles. Then, a random set of destinations was selected. However, only one destination was selected form each block; hence, no more than 24 destinations were selected. Finally, the number of destinations was varied so that the total number of links in the randomly generated topologies ranged from 14 to 22. Approximately 200 topologies were examined.

We first focus on the case where power control is not used.

[4]Of course, the transmission power is not permitted to exceed 1.

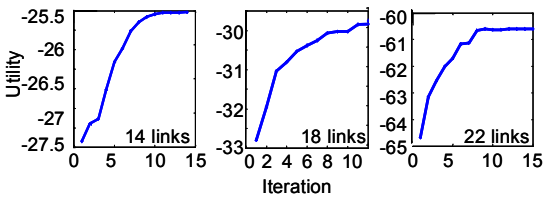

Fig. 5. Examples of the variation of utility as new assignments are considered.

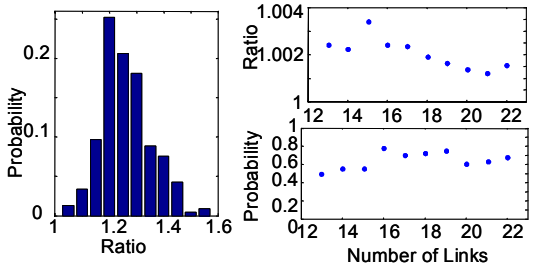

Fig. 6. The left-hand frame shows the distribution of the ratio of the average flow rate after adding assignments until the proposed search methods found no new assignments and the average flow rate the was determined from the initial set of considered assignments. The upper right frame shows the ratio of the average flow rates found from exhaustive search and the average flow rates after adding assignments until the proposed search methods found no new assignments. The lower right frame shows the probability that the exhaustive search finds a better assignment. Power control was not used in this experiment.

Figure 5 shows a set of typical variations in the utility as a function of the iterations. Note that in all cases, no new assignments could be found after 15 iterations. The left frame in Figure 6 shows a histogram of the difference between the initial flow rates found by optimizing over the initially considered assignments described in Section VI, and the final flow rates once no more new assignments could be found.

The upper right frame in Figure 6 shows the ratio of the flow rates found after no more assignments could be found and the flow rates after an exhaustive search was performed. Note that the exhaustive search increases the flow rate by no more than a few tenths of a percent. We conclude that the methods described in this paper are able to find good assignments in the sense that an exhaustive search results in a minor improvement in performance. On the other hand, the search methods typically do not find the best assignment. To see this, consider the lower right frame in Figure 6, which shows the probability that the exhaustive search finds an assignment that provides an improvement in performance. Figure 6 shows that for the probability of finding a better assignment through exhaustive search typically exceeds 0.5. Therefore, these results support the conclusion that the proposed method either finds the optimal schedule (this occurs with a probability between 0.5 and 0.25), or finds a schedule that is nearly optimal.

The rule for adding assignments is $r\mu - \lambda > 0$. From the proof of Theorem 4, it can be seen that the difference $r\mu - \lambda$ is a linear approximation of the improvement in performance. Based on this, it is reasonable to first add assignments that have large $r\mu - \lambda$ to the set of considered assignments before adding

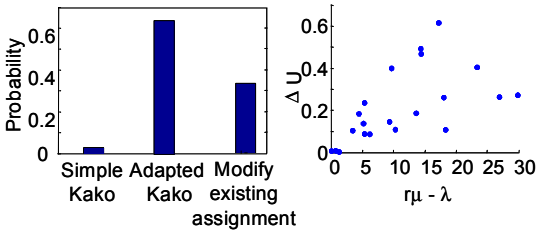

Fig. 7. The left frame shows the probability that a particular search method finds an assignment with the largest $r\mu - \lambda$. The right frame shows different values of $r\mu - \lambda$ and the corresponding difference in the utilities when the assignment r is added to the set of considered assignments.

assignments with small $r\mu - \lambda$. The right frame in figure 7 confirms this idea and shows that the resulting improvement in utility is well predicted by the size of $r\mu - \lambda$. Exploiting this relationship helps the method converge quickly, as displayed in Figure 5.

Section V developed several techniques for searching for new assignments. The left frame in Figure 7 shows the probability that a particular method finds an assignment with the largest $r\mu - \lambda$. Clearly, the adapted Kako algorithm provides the best performance. However, the other methods have some merit, and hence, these methods should be used to search for new assignments.

Next, the impact of power control is investigated. The left-hand frame of Figure 8 shows the distribution of the ratio of the utility found after searching for new assignments that may use power control and the utility that results from searching for assignments without power control. The left-hand frame of Figure 8 shows that allowing power control has only a small impact on performance. To further investigate the impact of power control, a thorough search was performed as described at the end of Section V-B. This search failed to produce any significant improvement in performance. Note that in [3], a slightly different problem was investigated and it was proved that power control has no impact, that is, with the optimal schedule, either a node transmits at full power or does not transmit at all. Figure 8 demonstrates that this is not the case here; power control does have a positive impact on performance. However, while [3] showed that allowing power control has no impact, here we find that power control has only a small impact.

The right-hand frame Figure 8 provides further insight into power control. This plot shows the distribution of the transmission power of the active assignments. This plot does not show the probability that the link transmission power is zero or one. The reason for this omission is that these probabilities are far larger than the ones shown in the plot. In fact, the probability that a link transmission power in an active assignment is zero or one exceeds 0.99. However, as this plot shows, if the power is not zero of one, then it is fairly uniformly distributed between zero and one, but has a slight tendency to be close to zero.

To demonstrate the ability to estimate the optimal schedule for large networks, we consider the 6×6 block region of

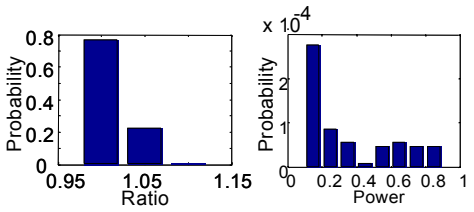

Fig. 8. The left-hand frame shows the distribution of the relative improvement in average flow rates offered by power control. The right-hand frame shows the distribution of the transmission power of the active assignment, however, the probability of the transmit power being one or zero is not shown.

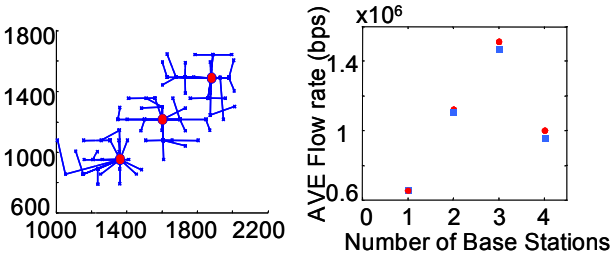

Fig. 9. The left frame shows the nodes positions in a network with three base stations and 56 destinations. The right frame shows the average flow rate when the optimal schedule is found. The rates that result from not using and using power control are shown. The rates that arise when power control is used are nearly the same as those found without using power control.

Chicago shown in the left-hand frame of Figure 9. Figure 9 shows the set of destinations and also an example of three base station locations. As a toy problem, the role of number of base stations was explored. Specifically, between one and four base stations were randomly distributed throughout the region occupied by destination. The resulting networks had between 63 and 73 links, in particular, there were 71, 72, 70, and 63 links for the case where there were 1, 2, 3, and 4 base stations, respectively. An estimate of the optimal utility was found for each network for the case where power control is permitted and for the case where power control is not permitted. The resulting average flow rate per destination is shown in the right-hand side of Figure 9. In these experiments it is assumed that the bandwidth is 20 MHz wide (as is the case in 802.11g).

Like the results shown in Figure 8, Figure 9 shows that power control has a minimal impact on the utility. Figure 9 also illustrates that arbitrary distribution of base stations may lead to poor performance, specifically, the scenario with four base stations resulted in worst performance than two or three base stations. This behavior is likely due to the increased congestion due to the fewer links used in the four base station case. This example demonstrated that the proposed computational technique can support capacity estimates for moderate size networks. We believe that with further effort, the proposed techniques can support networks with several hundred links.

VIII. Conclusions

Extensive research has focused on the capacity of a multihop wireless network. In this paper a tractable computational technique was presented to estimate the optimal schedule. In the cases where exhaustive search is tractable, the presented method was found to either find the optimal schedule or find a schedule that results in utility that is within 0.5% of the optimal schedule. The key to the approach is to focus on determining the set of bandwidth assignments over which the optimization is performed. Several techniques were presented to search for good assignments.

Disclaimer

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U. S. Government.

References

[1] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proceedings of IEEE INFOCOM*, April 2006.

[2] L. Chen, S. H. Low, and J. C. Doyle, "Joint congestion control and media access control design for wireless ad hoc networks," in *Proceedings of IEEE INFOCOM*, March 2005, pp. 2212–2222.

[3] R. Cruz and A. Santhanam, "Optimal routing, link scheduling and power control in multi-hop wireless networks," in *Proceedings of IEEE INFOCOM*, San Francisco, USA, March 2003, pp. 702– 711.

[4] T. ElBatt and A. Ephremides, "Joint scheduling and power control for wireless ad-hoc networks," in *Proceedings of IEEE INFOCOM*, New York, NY, June 2002, pp. 976–985.

[5] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer congestion control in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 302–315, April 2006.

[6] P.R.Kumar and P.Gupta, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, March 2000.

[7] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: Shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, Nov 1998.

[8] L. Massoulie and J. Roberts., "Bandwidth sharing: Objectives and algorithms." *IEEE/ACM Transactions on Networking*, no. 3, June 2002.

[9] X. Lin and S. Rasool, "Constant-time distributed scheduling policies for ad hoc wireless networks," 2006.

[10] A. Kako, T. One, T. Hirata, and M. M. Halldorsson, "Approximation algorithms for the weighted independent set problem," available at: http://www.hi.is/ mmh/papers/WIS_WG.pdf.

[11] S. Bohacek and P. Wang, "Toward tractable computation of the capacity multihop wireless networks. [extended version]," available at: http://udelmodels.eecis.udel.edu/publications1.php.

[12] S. Bohacek, V. Sridhara, and J. Kim, "UDel Models," available at: http://udelmodels.eecis.udel.edu/.

[13] S. Bohacek and P. Wang, "Toward tractable computation of the capacity multihop wireless networks. [implementation]," available at: http://udelmodels.eecis.udel.edu/lumnets1.php.

[14] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2003.

[15] F. P. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, January 1997.

[16] S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 525–536, August 2003.

[17] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*. Berlin: Springer-Verlag, 1985, p69.

[18] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proceedings of ACM MobiCom*, San Diego, CA, September 2003, pp. 66–80.

[19] M. M. Halldt'orsson, "Approximations of independent sets in graphs," in *The First International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, K. Jansen and J. Rolim, Eds., 1998, pp. 1–14.

[20] M. Grotschel, L. Lovasz, and A. Schrijver, *Geometrics Algorithms and Combinatorial Optimization*. Berlin: Springer-Verlag, 1993.