Syracuse University
SURFACE

Electrical Engineering and Computer Science College of Engineering and Computer Science

2011

Leveraging Multi-User Diversity, Channel Diversity and Spatial Reuse for Efficient Scheduling in Wireless Relay Networks

Shen Wan

Jian Tang
Syracuse University, jtang02@syr.edu

Brendan Mumey
Montana State University

Richard S. Wolff

Weiyi Zhang

Follow this and additional works at: https://surface.syr.edu/eecs

Part of the Electrical and Computer Engineering Commons

Recommended Citation

S. Wan, J. Tang, B. Mumey, R. S. Wolff, and W. Zhang, "Leveraging multi-user diversity, channel diversity and spatial reuse for efficient scheduling in wireless relay networks," in 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS 2011, October 17, 2011 - October 22, 2011, Valencia, Spain, 2011, pp. 401-410.

This Article is brought to you for free and open access by the College of Engineering and Computer Science at SURFACE. It has been accepted for inclusion in Electrical Engineering and Computer Science by an authorized administrator of SURFACE. For more information, please contact surface@syr.edu.

Leveraging Multi-User Diversity, Channel Diversity and Spatial Reuse for Efficient Scheduling in Wireless Relay Networks

Shen Wan, Jian Tang, Brendan Mumey, Richard S. Wolff and Weiyi Zhang

***Abstract*—Relay stations can be deployed in a wireless network to extend its coverage and improve its capacity. In this paper, we study a scheduling problem in OFDMA-based wireless relay networks with consideration for multi-user diversity, channel diversity and spatial reuse. First, we present a Mixed Integer Linear Programming (MILP) formulation to provide optimum solutions. It has been shown by previous research that performance of a wireless scheduling algorithm is usually related to the interference degree δ, which is the maximum number of links that interfere with a common link but do not interfere with each other. Therefore, we then show that the interference degree δ is at most 4 for any 2-hop relay network and 14 for any general h-hop ($h \geq 2$) relay network. Furthermore, we present a simple greedy algorithm for the scheduling problem and show it has an approximation ratio of $\frac{1}{1+\delta}$, which leads to an approximation ratio of $\frac{1}{5}$ for the 2-hop case and $\frac{1}{15}$ for the general case. In addition, we present three heuristic algorithms, namely, the weighted degree greedy algorithm, the Maximum Weighted Independent Set (MWIS) algorithm and the Linear Programming (LP) rounding algorithm, to solve the scheduling problem. Extensive simulation results have showed that the LP rounding algorithm performs best and always provides close-to-optimum solutions. The performance of the simple greedy algorithm is comparable to that of the other algorithms.**

***Keywords*—Wireless relay networks, WiMAX, scheduling, OFDMA.**

I. INTRODUCTION

A wireless relay network is composed of a Base Station (BS), a few Relay Stations (RSs) and a large number of Mobile Stations (MSs). The BS serves as a gateway connecting the network to external networks such as the Internet. A spanning tree rooted at the BS is formed for routing, in which all MSs are leaf nodes. If an MS is out of the transmission range of the BS, it can communicate with the BS via one or multiple RSs in a multihop manner. This kind of network architecture has been adopted by emerging wireless networking standards such as IEEE $802.16j$. The IEEE $802.16j$ task group [2] was formed to extend the scope of IEEE $802.16e$ to support Mobile Multihop Relay (MMR). Compared to a single-hop Point-to-MultiPoint (PMP) wireless network where each MS directly communicates with the BS, a relay network can significantly extend the coverage range, reduce dead spots and improve network capacity [2].

The emerging standards, such as the WiMAX standard [1], adopt a TDMA-based MAC layer in which the time domain is divided into timeslots and multiple timeslots are grouped together to form a scheduling frame. The WiMAX standard also provides a highly flexible MAC layer that can be implemented over a wide range of frequencies and different physical layer technologies such as Orthogonal Frequency Division Multiplexing(OFDM), Multiple Input Multiple Output (MIMO) and so on. Orthogonal Frequency Division Multiplexing Access (OFDMA) is an emerging OFDM-based multiple access technology. With OFDMA, the operating spectrum is divided into multiple sub-channels, each of which consists of multiple narrow frequency bands (a.k.a. sub-carriers). The multiple access is achieved by assigning different sub-channels to different users in the network for simultaneous transmissions. The available resources in an OFDMA-based wireless network can be viewed as *transmission blocks* (or simply *blocks*) in a two-dimensional structure with timeslots in one dimension and sub-channels in another [3], as illustrated in Fig. 1. Hence, the scheduling problem in such a network is the problem of assigning available transmission blocks to each link to optimize a certain objective function.

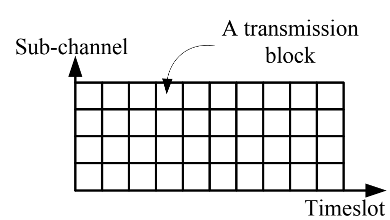


Fig. 1. Transmission blocks

Multipath fading and user mobility lead to independent fading across users for a sub-channel. Therefore, the gains of a channel for different users vary, which is referred to as *multi-user diversity* [15]. Moreover, different channels may experience different channel gains

Shen Wan was a Ph.D student working with Jian Tang when this work was done. He is currently with Amazon.com. Jian Tang is with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244. Email: jtang02@syr.edu Brendan Mumey and Richard Wolff are with the College of Engineering, Montana State University, Bozeman, MT 59717. Weiyi Zhang is with the AT&T Labs Research. This research was supported by NSF grant CNS-1113398. They would like to thank Prof. Li Xiao for her valuable suggestions.

for an MS due to frequency selective fading, which is referred to as *channel diversity* [15]. In addition, multiple users that are far apart can share a set of common subchannels, which is referred to as *spatial reuse*. To design an efficient scheduling algorithm, we need to exploit the benefits of multi-user diversity, channel diversity and spatial reuse.

In this paper, we study a scheduling problem in OFDMA-based wireless relay networks with consideration for multi-user and channel diversities, and spatial reuse. Similar to a closely related work [3], the objective is to keep the system stable, i.e., keep the length of every queue finite. We summarize our contributions as follows:

1) We present a Mixed Integer Linear Programming (MILP) formulation to provide optimum solutions, which can serve as a benchmark for evaluation.

2) The *interference degree* [11], [13] of a multihop wireless network is defined as the maximum number of non-interfering links that interfere with a common link. It has been shown by previous research [11], [13] that the performance of a wireless scheduling algorithms is usually related to the interference degree. *However, to the best of our knowledge, none of existing works provided a tight bound for the interference degree. In this paper, we show that the interference degree is at most* 4 *for any* 2*-hop relay network in which every MS can reach the BS within* 2 *hops and this bound is tight. Furthermore, we show that the interference degree is at most* 14 *for any general* h*-hop (*$h \geq 2$*) relay network.* These results are very important since they could be used to show that some simple or previously known heuristic scheduling algorithms are actually constant factor approximation algorithms in wireless relay networks.

3) We present a simple greedy algorithm and show it has an approximation ratio of $\frac{1}{1+\delta}$, i.e., it always find a solution with an objective value at least $\frac{1}{1+\delta}opt$, where opt is the corresponding optimum objective value. This gives an approximation ratio of $\frac{1}{5}$ for the 2-hop case where $\delta \leq 4$ and $\frac{1}{15}$ for the general h-hop ($h \geq 2$) case where $\delta \leq 14$.

4) We present three heuristic algorithms, namely, the weighted degree greedy algorithm, the Maximum Weighted Independent Set (MWIS) algorithm, and the Linear Programming (LP) rounding algorithm, to solve the problem, whose efficiency has been justified by extensive simulation results.

II. Related Work

Transmission scheduling is a fundamental problem in OFDMA-based wireless networks, which has been studied by several recent works. In [3], Andrews and Zhang considered several scheduling problems in single-hop OFDMA-based WiMAX networks. They analyzed the hardness of these problems and present several simple constant factor approximation algorithms to solve them. Furthermore, they considered the problem of creating template-based schedules for such networks in [4]. They presented a general framework to study the delay performance of a multi-carrier template. They then described several deterministic and randomized scheduling algorithms for template creation and studied their delay performance via analysis and simulation. In [15], Sundaresan *et al.* showed that the scheduling problem to exploit diversity gains alone in 2-hop WiMAX relay networks is NP-hard, and provided polynomial-time approximation algorithms to solve it. They also proposed a heuristic to exploit both spatial reuse and diversity gains. In [8], the authors studied a similar scheduling problem in OFDMA-based WiMAX relay networks. They provided an easy-to-compute upper bound. They also presented three fast heuristics and showed that they provide close-to-optimum solutions and outperform other existing algorithms by simulation results. In a very recent paper [16], the authors presented the NP-hardness proof and approximation algorithms for a problem of scheduling users with backlogged and finite buffers in OFDMA-based 2-hop wireless relay networks.

Transmission scheduling has also been studied for single-carrier or OFDM-based WiMAX mesh networks. Different centralized heuristics have been proposed for scheduling and/or routing in [9], [12], [18] with the objective of maximizing spatial reuse. In [14], the authors proposed routing and scheduling algorithms to provide per-flow QoS guarantees.

We summarize the novelty of this work and its differences from these related works as follows: 1) Unlike the closely related works [15], [16] that only presented scheduling algorithms, the major contribution of this work is that it shows that the well-known interference degree has constant bounds in both 2-hop and general multihop wireless relay networks, which has never been shown before. These results are expected to have a broader impact because they may be leveraged to show some simple or previously known heuristic algorithms can provide certain worst-case performance guarantees in wireless relay networks. We demonstrate that a simple greedy scheduling algorithm is a constant factor approximation algorithm based on these results. Moreover, [15], [16] only considered 2-hop wireless relay networks. In this work, we address not only 2-hop relay networks but also general h-hop relay networks with $h > 2$. 2) As pointed out by [3], [8], [15], the scheduling problem considered in this work is different from the scheduling problems in single-carrier or OFDM-based wireless mesh networks [9], [12], [14], [18], in which every link can only be assigned a single channel in a timeslot. 3) Spatial reuse is one of the main concerns of this work, which was not addressed by [3], [4], [8].

III. SYSTEM MODEL

We consider a TDMA-based MAC protocol [1], in which the time domain is divided into minislots with fixed durations, and multiple minislots are grouped together to form a scheduling frame. Each frame consists of a control subframe and a data subframe. Control subframes are used for exchanging control messages. Data transmissions occur in data subframes, each of which is further partitioned into a downlink subframe and an uplink subframe with T and T' minislots respectively. Note that T is not necessarily equal to T'. Since uplink and downlink transmission schedules can be computed separately, we only focus on downlink scheduling in the following. The proposed algorithms can be directly used for uplink scheduling as well. In an OFDMA-based wireless system, all the nodes are allowed to operate on C OFDM *sub-channels*, each of which consists of multiple sub-carriers. As mentioned before, MAC layer resources can be viewed as a total of $K = T \times C$ blocks. A link e_i can be assigned one or multiple blocks in a minislot for data transmissions. The scheduling problem is to determine how to allocate these blocks to links in the network [3], [8], [15], which will be formally defined in the next section.

Similar to [8], [15], we consider an OFDMA-based wireless relay network with n nodes including a BS, RSs and MSs. These nodes form a spanning tree rooted at the BS for routing. Since we only focus on downlink scheduling here, we consider $m = n-1$ directional links, each of which connects a node to one of its children. The index of the receiving node of each link in the tree is used as the index for that link, i.e., $e_i = (v_{p_i}, v_i)$, where p_i is the index of the parent node of v_i. Every non-leaf node is assumed to maintain a queue for each of its children. Therefore, we use q_i to denote the length of queue at node v_{p_i} for v_i, or simply the queue length of link e_i. Each MS is equipped with a single radio but each RS has two radios with one for its parent and another for its children in the routing tree. An MS can directly communicate with the BS if it is within the transmission range. All nodes are assumed to transmit at the same fixed power level. Therefore, each node has a fixed transmission range of R_T and a fixed interference range of R_I that is usually 2–3 times R_T.

We adopt the Fixed Power Protocol Interference Model (fPrIM) [17] to model interference. In this model, two links $e_i = (v_{p_i}, v_i)$ and $e_j = (v_{p_j}, v_j)$ are said to interfere with each other if $||v_{p_i}, v_j|| \le R_I$ or $||v_{p_j}, v_i|| \le R_I$, where $||\cdot||$ gives the distance between two nodes. An $m \times m$ matrix $\mathbf{I}$ is used to represent link interference relationships. $I_i^j = 1$ if link e_i interferes with link e_j; $I_i^j = 0$, otherwise. In an OFDMA-based multi-channel relay network, two interfering links e_i and e_j must be assigned different blocks.

We focus on centralized scheduling as specified by the WiMAX MAC protocol [1]. The BS gathers traffic information from all the MSs and RSs periodically, and controls transmission scheduling.

IV. PROBLEM FORMULATION

First, we summarize the major notations in Table I.

TABLE I
MAJOR NOTATIONS

$\mathbf{I}$	The interference matrix
I_i^j	An entry in matrix $\mathbf{I}$, $I_i^j = 1$ if link e_i interferes with link e_j; $I_i^j = 0$, otherwise. Let $I_i^i = 0$.
K	The number of blocks
$\mathbf{L}$	The set of $m \times K$ link-block pairs
n/m	The number of nodes/links in the network, $m = n - 1$
p_i	The index of the parent node of node v_i
q_i	The queue length of link e_i at the beginning of a frame
$\mathbf{Q}$	The queue length vector, $\mathbf{Q} = [q_1, \dots, q_i, \dots, q_m]$
$\mathbf{R}$	The data rate matrix
r_i^k	An entry in matrix $\mathbf{R}$, the data rate of link e_i that can be supported by block k
R_T/R_I	The transmission/interference range
$\mathbf{X}$	The block assignment matrix
x_i^k	An entry in $\mathbf{X}$, $x_i^k = 1$ if link e_i is assigned block k; $x_i^k = 0$, otherwise.
δ	The interference degree of the network

Definition 1: Given m links, K blocks, the queue length vector $\mathbf{Q}$, the interference matrix $\mathbf{I}$, and the data rate matrix $\mathbf{R}$, the scheduling problem seeks an interference-free block assignment that assigns a subset B_i of blocks to each link e_i such that the utility function $\sum_{i=1}^m q_i \min\{q_i, \sum_{k \in B_i} r_i^k\}$ is maximized.

We choose the above objective function because it is known that if a scheduling algorithm can achieve the above objective in each frame or timeslot, then it can keep the system stable, i.e., keep the length of each queue finite [3]. As mentioned before, such a stable scheduling algorithm is also considered to achieve 100% throughput [6].

Next, we present an MILP formulation for the scheduling problem. The decision variables are described as follows.

1) $x_i^k = 1$ if block k is assigned to link e_i, $x_i^k = 0$ otherwise. (mK such variables)
2) y_i, the effective utility value obtained on link e_i according to the assignment. (m such variables)

MILP:

$$\max \sum_{i=1}^{m} q_i y_i \tag{1}$$

subject to:

$$y_i \le q_i, \qquad i \in \{1,\dots,m\}; \tag{2}$$

$$y_i \le \sum_{k=1}^{K} r_i^k x_i^k, \qquad i \in \{1,\dots,m\}; \tag{3}$$

$$x_i^k + x_j^k \le 1, \quad i,j \in \{1,\dots,m\} \text{ and } I_i^j = 1, \quad k \in \{1,\dots,K\}; \tag{4}$$

$$x_i^k \in \{0,1\}, i \in \{1,\dots,m\}, k \in \{1,\dots,K\}. \tag{5}$$

In this formulation, constraints (2) and (3) make sure that $y_i = \min\{q_i, \sum_{k\in B_i} r_i^k\}$. Constraint (4) is the interference constraint, which ensures that if two links interfere with each other, they are not assigned the same block. The objective is to maximize the aforementioned utility function.

As mentioned before, a similar scheduling problem in single-hop PMP OFDMA-based wireless networks has been shown to be NP-hard [3]. It is a special case of our scheduling problem since in a single-hop wireless network, all links interfere with each other and no spatial reuse is possible. Therefore, our scheduling problem is more general and harder.

V. Interference Degree

It has been shown by previous research [11], [13] that the performance of a wireless scheduling algorithm is usually highly related to the interference degree. So we first show the bounds of interference degree in wireless relay networks, which will be used to analyze the performance of the proposed algorithms later.

The interference degree $\delta(e)$ of link e is the maximum number of links that interfere with e but do not interfere with each other. It characterizes the potential loss of system capacity if link e is scheduled. The interference degree δ of the whole network is the maximum interference degree among all links. First of all, we focus on the interference degree in a 2-hop relay network. Without loss of generality, let the transmission range $R_T = 1$ and the interference range $R_I = 2$. All the following claims still hold if $R_I > 2$. We denote the BS as O, the transmitting and receiving nodes of each link e_i on layer 2 as T_i and R_i, respectively. Then we have $\|OT_i\| \le 1, \|T_iR_i\| \le 1, 1 < \|OR_i\| \le 2$.

Lemma 1: For any pair of non-interfering links (T_i, R_i) and (T_j, R_j) on layer 2, $\angle T_iOT_j > \frac{\pi}{3}$.

Proof: Since $\|T_iR_j\| > 2$ and $\|T_jR_j\| \le 1$, in $\triangle T_iT_jR_j$, $\|T_iT_j\| > \|T_iR_j\| - \|T_jR_j\| > 1$. In $\triangle OT_iT_j$, $\|OT_i\| \le 1$, $\|OT_j\| \le 1$, and $\|T_iT_j\| > 1$, therefore $\angle T_iOT_j > \frac{\pi}{3}$. ■

Lemma 2: For any link (T_i, R_i) on layer 2, $\angle T_iOR_i < \frac{\pi}{3}$.

Proof: In $\triangle OT_iR_i$, $\|OT_i\| \le 1$, $\|OR_i\| > 1$, and $\|T_iR_i\| \le 1$, therefore $\angle T_iOR_i < \frac{\pi}{3}$. ■

Lemma 3: For any pair of non-interfering links (T_i, R_i) and (T_j, R_j) on layer 2, $\angle T_iOR_j > \frac{5\pi}{12}$.

Proof: In $\triangle OT_iR_j$, $\|OT_i\| \le 1$, $\|OR_j\| \le 2$, $\|T_iR_j\| > 2$, hence $\cos\angle T_iOR_j < \frac{\|OT_i\|^2+\|OR_j\|^2-\|T_iR_j\|^2}{2\|OT_i\|\|OR_j\|} < \frac{1}{4}$. So $\angle T_iOR_j > \arccos\frac{1}{4} > \frac{5\pi}{12}$. ■

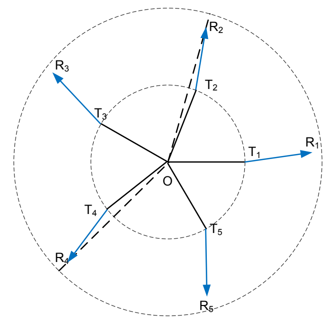


Fig. 2. Impossible to have a 2-hop relay network with $\delta = 5$

Theorem 1: The interference degree of any 2-hop relay network is at most 4.

Proof: We prove this by showing that it is impossible to have 5 non-interfering links. First, we notice that for any two links on layer 1, they interfere with each other. Second, for a link (O, T_1) on layer 1 and another link (T_2, R_2) on layer 2, they interfere with each other since $\|OR_2\| \le 2$. Therefore, any two non-interfering links must be both on layer 2. Assume there exist 5 non-interfering links $(T_1, R_1), \dots, (T_5, R_5)$ on layer 2. We denote $\theta(x)$ as the angle of the point x in the polar coordinate system originated at the BS O. Without loss of generality, let $\theta(T_1) \le \cdots \le \theta(T_5)$, as shown in Fig. 2. We have $\theta(R_2) \ge \theta(T_1) + \angle T_1OT_2 - \angle T_2OR_2 > \theta(T_1) + \frac{\pi}{3} - \frac{\pi}{3} = \theta(T_1)$, $\theta(T_3) = \theta(T_2) + \angle T_2OT_3 > \theta(R_2) - \angle T_2OR_2 + \angle T_2OT_3 > \theta(R_2) - \frac{\pi}{3} + \frac{\pi}{3} = \theta(R_2)$, etc. Therefore, we have $\theta(T_1) < \theta(R_2) < \theta(T_3) < \theta(R_4) < \theta(T_5)$. According to Lemmas 1 and 3, we have $\angle T_1OR_2 + \angle R_2OT_3 + \angle T_3OR_4 + \angle R_4OT_5 + \angle T_5OT_1 > \frac{5\pi}{12} + \frac{5\pi}{12} + \frac{5\pi}{12} + \frac{5\pi}{12} + \frac{\pi}{3} = 2\pi$, which is impossible. This completes the proof. ■

We show the interference degree bound 4 is tight by constructing an example in Fig. 3. The coordinates of the BS O are $(0,0)$, T_0 are $(0,-\epsilon')$, and R_0 are $(0,-1-\epsilon')$ (out of the transmission range of the BS). Therefore, (T_0, R_0) is a link on layer 2. The coordinates of T_1, T_2, T_3, T_4 are $(0,-1+\epsilon)$, $(-1+\epsilon,0)$, $(0,1-\epsilon)$ and $(1-\epsilon,0)$ respectively. The coordinates of R_1 to R_4 are $(0,-2+\epsilon)$, $(-2+\epsilon,0)$, $(0,2-\epsilon)$ and $(2-\epsilon,0)$ respectively. Here ϵ and ϵ' are two arbitrarily small positive numbers and $\epsilon' < \epsilon$. It is easy to verify that links $(T_1, R_1), \dots, (T_4, R_4)$ are all on layer 2. They do not interfere with each other but they all interfere with the link (T_0, R_0). Therefore, $\delta(T_0, R_0) = 4$ and $\delta = 4$

for this network.

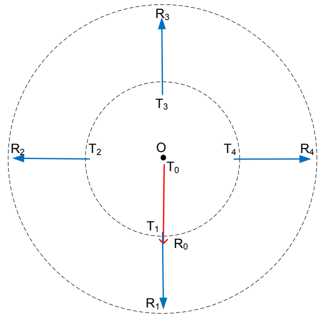


Fig. 3. An example with $\delta = 4$

Next we consider the interference degree in general h-hop relay networks.

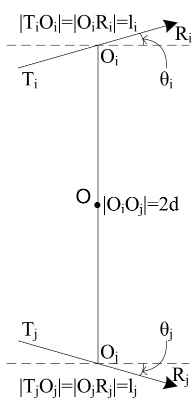


Fig. 4. Proof of Lemma 4

Lemma 4: For any pair of non-interfering links (T_i, R_i) and (T_j, R_j), let their center points be O_i and O_j respectively. Then $\|O_iO_j\| > \sqrt{3}$.

Proof: See Fig. 4. Let the center point of O_i and O_j be the origin point and let the coordinates of O_i and O_j be $(0, d)$ and $(0, -d)$, respectively. Let $l_i = \frac{1}{2}\|T_iR_i\|$, $l_j = \frac{1}{2}\|T_jR_j\|$, $\theta_i = \frac{\pi}{2} - \angle T_iO_iO_j$, and $\theta_j = \frac{\pi}{2} - \angle T_jO_jO_i$. Then $T_i = (-l_i \cos\theta_i, d - l_i \sin\theta_i)$, $R_i = (l_i \cos\theta_i, d + l_i \sin\theta_i)$, $T_j = (-l_j \cos\theta_j, -d + l_j \sin\theta_j)$ and $R_j = (l_j \cos\theta_j, -d - l_j \sin\theta_j)$. We have $\|T_iR_j\|^2 = l_i^2 + l_j^2 + 2l_il_j \cos(\theta_i + \theta_j) + 4d^2 + 4d(l_j \sin\theta_j - l_i \sin\theta_i) > 2^2 = 4$. Since $l_i \le \frac{1}{2}$, $l_j \le \frac{1}{2}$, and $\cos(\theta_i + \theta_j) \le 1$, we can deduce that $4d^2 + 4d(l_j \sin\theta_j - l_i \sin\theta_i) > 3$. Similarly, we can deduce that $4d^2 + 4d(l_i \sin\theta_i - l_j \sin\theta_j) > 3$ from $\|T_jR_i\| > 2$. Therefore, $8d^2 > 6$, and $\|O_iO_j\| = 2d > \sqrt{3}$. ■

Lemma 5: For any pair of interfering links (T_i, R_i) and (T_j, R_j), let their center points be O_i and O_j respectively. Then $\|O_iO_j\| \le 3$.

Proof: Without loss of generality, we assume that the interference is caused by $\|T_iR_j\| \le 2$. Then we have $\|O_iO_j\| \le \|O_iT_i\| + \|T_iR_j\| + \|R_jO_j\| \le \frac{1}{2} + 2 + \frac{1}{2} = 3$. ■

Theorem 2: The interference degree of any general h-hop ($h \ge 2$) relay network is at most 14.

Proof: Consider a link with a center point of O. Suppose there are N non-interfering links which interfere with this link. Denote the center points of these N links as $O_1, \ldots, O_N$. Then we have:

$$\|O_iO_j\| > \sqrt{3}, \quad i, j \in \{1, \ldots, N\} \quad \text{by Lemma 4(1)}$$
$$\|OO_i\| \le 3, \quad i \in \{1, \ldots, N\} \quad \text{by Lemma 5(2)}$$

If we place circles with a diameter of $\sqrt{3}$ centered at each O_i, we can see from (1) that these circles do not overlap. Next, we place a big circle with a diameter of $6 + \sqrt{3}$ centered at O, we can see from (2) that all the small circles are contained in this big circle. Kravitz has shown in [20] that in order to pack 15 circles with a radius of r in a larger circle with a radius of R, $\frac{R}{r} > 4.52$. However, $\frac{6+\sqrt{3}}{\sqrt{3}} < 4.52$. Therefore, it is impossible to pack 15 such small circles in the big circle, which means it is impossible to have 15 non-interfering links that all interfere with a common link. This completes the proof. ■

VI. Proposed Scheduling Algorithms

In this section, we present a simple greedy approximation algorithm for the problem defined above and analyze its performance. We also present three heuristic algorithms, namely, the weighted degree greedy algorithm, the MWIS algorithm and the LP rounding algorithm.

A. The Simple Greedy Algorithm

The basic idea of this algorithm is to check all the link-block combinations and select the one which can lead to the maximum utility gain in each step.

Algorithm 1 The Simple Greedy Algorithm

Step 1 $\mathbf{X} \leftarrow \mathbf{0}$; $\mathbf{Q}' \leftarrow \mathbf{Q}$;
Step 2 $(i_{\text{best}}, k_{\text{best}}) \leftarrow \text{argmax}_{(i,k)}\, q_i \min\{q'_i, r_i^k\}$;
 if $q_{i_{\text{best}}} \min\{q'_{i_{\text{best}}}, r_{i_{\text{best}}}^{k_{\text{best}}}\} = 0$ **return** $\mathbf{X}$;
Step 3 $x_{i_{\text{best}}}^{k_{\text{best}}} \leftarrow 1$;
 $q'_{i_{\text{best}}} \leftarrow \max\{0, q'_{i_{\text{best}}} - r_{i_{\text{best}}}^k\}$;
 $\mathbf{L} \leftarrow \mathbf{L} \setminus \{(i_{\text{best}}, k_{\text{best}})\}$;
 forall $j : I_{i_{\text{best}}}^j = 1$ **do** $\mathbf{L} \leftarrow \mathbf{L} \setminus \{(j, k_{\text{best}})\}$;
 if $\mathbf{L} = \emptyset$ **return** $\mathbf{X}$;
Step 4 **goto** Step 2;

In Algorithm 1, q'_i is the remaining queue length of link e_i after some blocks are allocated to link e_i. After initialization in Step 1, the greedy algorithm always selects a link-block pair $(i_{\text{best}}, k_{\text{best}})$, i.e., assign block k_{best} to link $e_{i_{\text{best}}}$, such that it achieves the maximum utility gain in Step 2. Step 3 updates the remaining queue length $q'_{i_{\text{best}}}$ and the block assignment matrix $\mathbf{X}$. The selected link-block pair $(i_{\text{best}}, k_{\text{best}})$ is then removed from the list $\mathbf{L}$. In addition, the algorithm ensures that those links interfering with link $e_{i_{\text{best}}}$ will not be assigned block k_{best} by removing the corresponding link-block pairs from $\mathbf{L}$. The algorithm terminates when no more utility gain can be obtained or $\mathbf{L}$ becomes empty, which indicates either all blocks have been assigned or all queues are empty.

The algorithm takes $O(mK)$ time to compare all link-block pairs and select one in Step 2. The running time of Step 2 dominates the running time of the loop from Step 2 to Step 4. The loop will be executed for at most $|\mathbf{L}|$ times. Therefore, the time complexity of the simple greedy algorithm is $O(mK|\mathbf{L}|) = O(m^2K^2)$.

Next, we analyze the performance of the simple greedy algorithm.

Theorem 3: For a network with an interference degree of $\delta \geq 1$, the simple greedy algorithm has an approximation ratio of $\frac{1}{1+\delta}$.

Proof: Each available link-block (l-b) pair (i,k) in L is evaluated in Step 2 of the greedy algorithm and is ranked by its current utility gain, defined as follows: Let S be the set of l-b pairs already allocated to v_i. Then

$$gain((i,k'),S) = q_i \min(q_i, \sum_{(i,k)\in S\cup\{(i,k')\}} r_i^k) - q_i \min(q_i, \sum_{(i,k)\in S} r_i^k).$$

The l-b pair $(i,k) \in L$ with the highest utility gain is selected and removed, block k is allocated to link i, and interfering l-b pairs (j,k) are removed from L; we say that (j,k) was *blocked* by (i,k). For simplicity we will assume that all l-b pairs are either allocated or blocked (in practice the algorithm stops allocating l-b pairs when none have positive gain). Let X be the list of l-b pairs selected by the greedy algorithm (in the order that they were selected) and let X^* be an optimal list of l-b pairs (in some fixed order). We also define the sublists $X_i = \{(i,k) \in X\}$ and $X_i^* = \{(i,k) \in X^*\}$ (with the same orderings as their parent lists). For any l-b pair $(i,k) \in X_i$, let $X_{i,<(i,k)}$ be the l-b pairs from X_i that precede (i,k) in the list X. Define $X^*_{i,<(i,k)}$ similarly. For each l-b pair $(i,k) \in X$, let $g_i^k = gain((i,k), X_{i,<(i,k)})$. Likewise, for each $(i,k) \in X^*$, let $g_i^{*k} = gain((i,k), X^*_{i,<(i,k)})$. The value of the objective function (1) achieved by the greedy algorithm is $V = \sum_{(i,k)\in X} g_i^k$ and the optimal value is $V^* = \sum_{(i,k)\in X^*} g_i^{*k}$. Any l-b pair $(i,k) \in X^* \setminus X$ was blocked by some other l-b pair (i',k) during the execution of the greedy algorithm. If $g_i^{*k} > g_{i'}^k$, we say that (i,k) was *blocked to a loss* and if $g_i^{*k} \leq g_{i'}^k$, we say that (i,k) was *blocked to a gain*. Any l-b pair $(i,k) \in X^* \cap X$ was allocated by the greedy algorithm; if $g_i^{*k} > g_i^k$ we say (i,k) was *allocated to a loss* and if $g_i^{*k} \leq g_i^k$, we say (i,k) was *allocated to a gain*. Let $X_i^{*\text{al}} = \{(i,k) \in X_i^* : (i,k)$ was allocated to a loss$\}$, $X_i^{*\text{ag}} = \{(i,k) \in X_i^* : (i,k)$ was allocated to a gain$\}$, $X_i^{*\text{bl}} = \{(i,k) \in X_i^* : (i,k)$ was blocked to a loss$\}$, and $X_i^{*\text{bg}} = \{(i,k) \in X_i^* : (i,k)$ was blocked to a gain$\}$. Let $A = \{i : X_i^{*\text{al}} \neq \emptyset\}$. Note that if $i \in A$, there is an l-b pair (i,k) that provided strictly less gain to v_i than it did in the optimal solution; this implies that $\sum_{(i,k)\in X_i} g_i^k = q_i^2$. Let $B = \{i : X_i^{*\text{al}} = \emptyset \wedge X_i^{*\text{bl}} \neq \emptyset\}$. If $i \in B$, let (i,k) be some element of $X_i^{*\text{bl}}$ and suppose (i,k) was blocked by some $(i',k) \in X$, thus $gain((i,k), X_{i,<(i',k)}) \leq g_{i'}^k < g_i^{*k}$. So (i,k) provided strictly less gain to v_i at the time it was blocked than it did in the optimal solution; this implies that $\sum_{(i,k')\in X_i} g_i^{k'} + g_{i'}^k \geq q_i^2$. For each $i \in B$, we add the l-b pair (i',k) to a multiset $\mathcal{M}_B$. Let $C = \{i : X_i^{*\text{al}} = \emptyset \wedge X_i^{*\text{bl}} = \emptyset\}$. For each $i \in C$ and $(i,k) \in X^{*\text{bg}}$, (i,k) was blocked by some $(i',k) \in X$ with $g_i^{*k} \leq g_{i'}^k$; we add these l-b pairs (i',k) to a multiset $\mathcal{M}_C$. Let $\mathcal{M} = \mathcal{M}_B \cup \mathcal{M}_C$. Observe that the multiplicity of any l-b pair $(i,k) \in \mathcal{M}$ is at most δ; and $A \cup B \cup C = \{1, \ldots, m\}$. Then

$$\begin{aligned}
V^* = & \sum_{\substack{i\in A,\\(i,k)\in X_i^*}} g_i^{*k} + \sum_{\substack{i\in B,\\(i,k)\in X_i^*}} g_i^{*k} + \sum_{\substack{i\in C,\\(i,k)\in X_i^*}} g_i^{*k} \\
\leq & \sum_{i\in A} q_i^2 + \sum_{i\in B} q_i^2 + \sum_{\substack{i\in C,\\(i,k)\in X_i^{*\text{ag}}}} g_i^{*k} + \sum_{\substack{i\in C,\\(i,k)\in X_i^{*\text{bg}}}} g_i^{*k} \\
\leq & \sum_{\substack{i\in A,\\(i,k)\in X_i}} g_i^k + \Big(\sum_{\substack{i\in B,\\(i,k)\in X_i}} g_i^k + \sum_{(i,k)\in\mathcal{M}_B} g_i^k\Big) \\
& + \sum_{\substack{i\in C,\\(i,k)\in X_i}} g_i^k + \sum_{(i,k)\in\mathcal{M}_C} g_i^k \\
= & \sum_{i\in\{1,\ldots,m\}} \sum_{(i,k)\in X_i} g_i^k + \sum_{(i,k)\in\mathcal{M}} g_i^k \\
\leq & V + \delta V = (1+\delta)V
\end{aligned}$$

Thus $V \geq \frac{1}{1+\delta} V^*$ as claimed. ■

Note that our result is consistent with but more general than that in [3] which considers the same scheduling problem in a single-hop OFDMA-based network where all MSs interfere with each other (no spatial reuse), i.e., the interference degree is 1. In [3], the authors showed a simple greedy algorithm has an approximation ratio of $\frac{1}{2}$. By applying our theorem, we can also obtain the same approximation ratio for the simple greedy algorithm. However, our result is more general since it can be used for the case where any pair of links may or may not interfere with each other.

Combining Theorem 3 with Theorems 1&2, we have:

Corollary 1: The simple greedy algorithm has an approximation ratio of $\frac{1}{5}$ for any 2-hop relay network where $\delta \leq 4$ and $\frac{1}{15}$ for any general h-hop relay network where $\delta \leq 14$.

B. The Weighted Degree Greedy Algorithm

The simple greedy algorithm always selects the link-block pair with the maximum utility gain even if this may make the block unavailable to serve a set of non-interfering links, which could contribute more utility gain. To improve it, we present another greedy algorithm that makes the greedy choice based on the ratio of the possible loss to the gain (similar to the price-quality ratio). Simulation results showed that such greedy selection does make some improvements on average cases.

We first define the *contention graph* $G_C = (V_C, E_C)$, where each vertex in V_C corresponds to a link and there exists an undirected edge connecting two vertices if the corresponding links interfere with each other. The weight of each vertex (link) for block k is set to $w_i^k \leftarrow q_i \min\{q_i', r_i^k\}$, which is the utility gain achieved by assigning block k to link e_i. We then define the *weighted interference degree* of link e_i in G_C for block k as $d_w^k(i, G_C) = \frac{\max_{E_i} \sum_{e_j \in E_i} w_j^k}{w_i^k}$, where E_i is a set of non-interfering links that interfere with link e_i. Usually, there are multiple such non-interfering link sets for a link e_i. We select the set with the maximum total utility gain to calculate the weighted interference degree. Note that G_C is block dependent and it changes during the execution of the algorithm.

Algorithm 2 The Weighted Degree Greedy Algorithm

Step 1 $\mathbf{X} \leftarrow \mathbf{0}$; $\mathbf{Q}' \leftarrow \mathbf{Q}$;
Step 2 $(i_{\text{best}}, k_{\text{best}}) \leftarrow \text{argmin}_{(i,k)}\, d_w^k(i, G_C)$;
  if $w_{i_{\text{best}}}^{k_{\text{best}}} = 0$ **return** $\mathbf{X}$;
Step 3 $x_{i_{\text{best}}}^{k_{\text{best}}} \leftarrow 1$;
  $q'_{i_{\text{best}}} \leftarrow \max\{0, q'_{i_{\text{best}}} - r^k_{i_{\text{best}}}\}$;
  $\mathbf{L} \leftarrow \mathbf{L} \setminus \{(i_{\text{best}}, k_{\text{best}})\}$;
  forall $j : I^j_{i_{\text{best}}} = 1$ **do** $\mathbf{L} \leftarrow \mathbf{L} \setminus \{(j, k_{\text{best}})\}$;
  if $\mathbf{L} = \emptyset$ **return** $\mathbf{X}$;
Step 4 **goto** Step 2;

In Step 2, the weighted degree of each link can be computed in polynomial time since according to Theorems 1 and 2, the size of any set of non-interfering links is bounded by a constant (4 for the 2-hop case and 14 for the h-hop case). Hence, all such sets can be enumerated in polynomial time. The other parts of the algorithm are the same as the simple greedy algorithm. Therefore, this greedy algorithm is a polynomial time algorithm.

C. The Maximum Weighted Independent Set (MWIS) Algorithm

The basic idea of the MWIS algorithm is to assign a block to a maximal set of non-interfering links in each step and keep doing it until all blocks are assigned.

Algorithm 3 The MWIS Algorithm

Step 1 $\mathbf{X} \leftarrow \mathbf{0}$; $\mathbf{Q}' \leftarrow \mathbf{Q}$; $k \leftarrow 1$;
Step 2 Construct a weighted contention graph G_C;
Step 3 Use an MWIS algorithm to find an independent set E_{IS} in G_C;
  forall $e_i \in E_{IS}$ **do**
    $x_i^k \leftarrow 1$; $q_i' \leftarrow \max\{0, q_i' - r_i^k\}$;
  endforall
Step 4 **if** $k = K$ **return** $\mathbf{X}$;
  $k \leftarrow k + 1$;
  goto Step 2;

Algorithm 3 assigns K blocks one by one. Essentially, a block can be allocated to a maximal set of non-interfering links (an independent set in G_C) to maximize spatial utilization. Steps 2 and 3 determine which subset of non-interfering links the current block k should be assigned to. The objective is to maximize the utility gain. We also use a contention graph $G_C = (V_C, E_C)$ described above to assist the computation. Similarly, the weight of each vertex (link) in G_C for block k is set to $w_i^k \leftarrow q_i \min\{q_i', r_i^k\}$, which is the utility gain achieved by assigning block k to link e_i. Note this weight changes during the execution of the algorithm. Therefore, we construct a G_C for every block k. In each step, the maximum utility gain can be achieved by finding an MWIS on G_C. However, the MWIS problem is a well-known NP-hard problem [21]. Any MWIS algorithm in the literature [21] can be applied here. Note that the MWIS subproblem can be solved exactly in polynomial time for any 2-hop relay network because the proof of Theorem 1 implies that the size of any independent set in this case is at most 4. So we can enumerate all of them in polynomial time.

In the simulation, we used the greedy approximation algorithm described in [21] to compute a maximal independent set E_{IS} in G_C. This algorithm repeatedly selects a vertex (link) with the minimum weighted degree, puts it into the result set, and removes this vertex and all its neighbors from G_C, until G_C becomes empty. The weighted degree of a vertex v in G_C is defined as $d_w(v) = \frac{\sum_{v' \in N_v} w_{v'}}{w_v}$, where N_v denotes the set of neighbors of v in G_C. This algorithm has been shown to have an approximation ratio of $\frac{1}{1+\eta}$, where $\eta = \max_{v \in V_C} d_w(v)$ [21]. The running time for computing an MWIS in G_C is $O(m^2\Delta^2)$, where Δ is the maximum vertex degree in G_C. In Algorithm 3, the loop from Step 2 to Step 4 will be executed K times and Step 3 dominates its running time. Therefore, the time complexity of the MWIS algorithm is $O(m^2\Delta^2 K)$.

D. The LP Rounding Algorithm

LP rounding is a common approach for solving ILP problems. The critical part of LP rounding is the rounding scheme, where the rounding algorithm decides which variable(s) should be rounded to 1. Basically, our algorithm determines the values of a subset of variables by solving the relaxed LP and then updates the LP in each step. It keeps doing this until the values of all variables are determined.

For our scheduling problem, we found out that a trivial rounding scheme usually results in poor performance, since after rounding, it is very likely that the rate summation of all the blocks allocated to a link outnumbers its queue length. In this case, part of channel capacity is wasted and the values of variables x_i^k are not tightly fixed in the LP solution. In other words, it is possible to increase the value of some x_i^k and decrease the value of another $x_i^{k'}$ while keeping the LP solution's feasibility and utility value. However, such a randomness in the LP solution may result in different MILP solutions and often causes a low quality MILP solution. In Algorithm 4, we propose a novel rounding scheme. Our idea is to figure out the maximum possible value of each variable in the LP solution to eliminate the aforementioned randomness. Unlike other rounding methods, we consider two factors for making a rounding decision: the value of each variable in the LP solution and the maximum possible utility gain that can be provided by this variable (its *potential*).

Our LP rounding algorithm starts by sorting link-block pairs according to their maximum achievable utility gains. Next, we relax the original MILP problem $\mathcal{P}$ to an LP problem $\mathcal{P}'$ that can be solved by any existing LP solving algorithm [5]. In Step 3, we select those link-block pairs that not only are selected by the LP solution but also appear on the top of the link-block pair list $\mathbf{L}$. So we round the corresponding variables x_i^k to 1 and all variables $x_j^k : I_i^j = 1$ to 0 (variables corresponding to interfering link-block pairs), remove all the corresponding link-block pairs from $\mathbf{L}$, and update $\mathcal{P}'$. The rounding subroutine, *Round(·)*, performs the above steps, which guarantee that the interference constraints in the original MILP are not violated. Step 3 terminates when we find that a channel-block pair with the largest utility gain among those not selected by the LP solution. Notice that the failure to select this link-block pair in the LP solution may be due to the randomness. We eliminate the randomness in Step 4 by finding the maximum possible x_i^k value for each link-block pair that has not been selected so far under the constraint that the utility

---

Algorithm 4 The LP Rounding Algorithm

Step 1 Sort all the link-block pairs in $\mathbf{L}$ in the descending order of $q_i r_i^k$;
Step 2 Solve the LP relaxation $\mathcal{P}'$ of the MILP problem $\mathcal{P}$ defined in Section IV;
Step 3 **forall** $(i, k) \in \mathbf{L}$ **do**
  if $x_i^k = 1$ in the solution of $\mathcal{P}'$ **then** Round(x_i^k);
  else break;
  endforall
Step 4 **forall** $(i, k) \in \mathbf{L}$ **do**
  Solve a new LP problem $\mathcal{P}''$ which is the same as $\mathcal{P}'$ except that its objective is to maximize x_i^k and it has one more constraint fixing the value of the utility function (1) to that given by the solution of $\mathcal{P}'$;
  if $x_i^k = 1$ in the solution of $\mathcal{P}''$ **then** Round(x_i^k);
  endforall
Step 5 **if** ($\mathbf{L} \neq \emptyset$)
  then Select $(i, k) \in \mathbf{L}$ with the maximum value of x_i^k in the solution of $\mathcal{P}''$ from Step 4 and Round(x_i^k);
Step 6 **if** $\mathbf{L} = \emptyset$ **return** $\mathbf{X}$;
  goto Step 2;

---

function value is conserved, i.e., finding their potentials. This can be done by solving a series of LPs. In effect, Step 4 looks for more variables and confirms their value assignments. So far our algorithm has not rounded any variable from *a fractional value* to 1 yet. Consequently, the utility value remains equal to that given by the LP $\mathcal{P}'$. After Step 4, none of the remaining undecided variables can be rounded to 1 without reducing the utility value. Hence, we select a link-block pair with the largest value of x_i^k and round the corresponding variable to 1 in Step 5. It is possible to select multiple non-conflicting variables in Step 5 and round all of them to 1 at once. This will speed up the algorithm at the likely cost of reducing the solution quality. It terminates when the values of all the variables are decided.

The LP rounding algorithm is obviously a polynomial time algorithm since all the LP problems can be solved in polynomial time. In the worst case, we may need to solve many LPs. However, the algorithm is time efficient in practice because it can often round many variables in each loop, and once a variable has been rounded to 1 (a link-block pair is selected), all variables corresponding to the interfering link-block pairs will be rounded to 0.

VII. Numerical Results

In this section, we present simulation results to show the performance of the proposed algorithms. The ILOG

CPLEX 10.1 [7] was used to solve all the LP and MILP problems. Optimal solutions obtained by the CPLEX were used as the benchmark for performance comparison. Note that we were not able to compare our algorithms with the algorithms in closely related works such as [15], [16] because we used a different objective function (See the problem definition in Section IV) and it is not trivial to modify their algorithms to achieve the objective considered here.

We set the transmission range to 1km and the interference range to 2km. For scenarios regarding h-hop networks, we uniformly placed all nodes in a 5km$\times$5km square region. For scenarios regarding 2-hop networks, we uniformly placed all nodes in a circle with a radius of 2km. The BS was always placed at the center of the region. We used the Breadth First Search (BFS) algorithm to construct routing trees rooted at the BS. All non-leaf nodes served as RSs in the simulation. We calculated path losses for all links using the model described in [19]. The effective propagation exponent calculated from the model is 3.3. To reflect the heterogeneous channel fading, we let the path loss fluctuate around the calculated mean value according to the Rayleigh distribution. According to the WiMAX standard [1], there are 7 adaptive modulation schemes: BPSK$\frac{1}{2}$, QPSK$\frac{1}{2}$, QPSK$\frac{3}{4}$, 16QAM$\frac{1}{2}$, 16QAM$\frac{3}{4}$, 64QAM$\frac{2}{3}$, and 64QAM$\frac{3}{4}$ which can support 0.5, 1, 1.5, 2, 3, 4, and 4.5 bits per symbol respectively. Since the symbol rate is fixed, we normalized the link data rates to $\{1, 2, 3, 4, 6, 8, 9\}$. Since we assume that all nodes transmit at the same fixed power level, the data rate of a link is computed by applying a stair function to the path loss, resulting in one of the 7 possible data rates. Traffic demands (queue lengths) were randomly generated according to the binomial distribution. The mean value was fixed for all nodes in one scenario but varied in different scenarios. The results are shown in Figs. 5–6.

In the scenarios corresponding to Fig. 5, we fixed the network size to 30 and varied the mean queue length. We compared the performance of the four proposed algorithms against the optimum solutions obtained by solving the MILP problem. The quality of a solution given by a proposed algorithm was measured by dividing its utility value by the corresponding optimum value. We make the following observations:

1) All the proposed algorithms offer decent performance. Specifically, they all achieve at least 91% of the optimum utility values. Their performance are also stable with regards to different traffic loads. Note that the longer the mean queue length is, the heavier the traffic load becomes.

2) The LP rounding algorithm performs best. On average, it outperforms the simple greedy algorithm by 5.4%, the weighted degree greedy algorithm (labeled as "WD-Greedy") by 2.3%, and the MWIS algorithm by 5.2%. In most cases, the LP rounding algorithm can find a solution with a utility difference of 1% to the optimum value. It can often find the optimum solutions.

In summary, on average cases, the LP rounding algorithm performs best. However, it has a high time complexity and is hard to implement. In most cases, the weighted degree greedy algorithm performs better than the simple greedy algorithm as expected at the cost of longer running time. In relatively sparse networks (i.e., the maximum vertex degree in the contention graph, Δ, is a small constant), the MWIS algorithm has the lowest time complexity. It offers quite similar performance as the simple greedy algorithm. The simple greedy algorithm is easy to implement and provides comparable performance as the other three algorithms on average cases. It has a constant approximation ratio and thus can provide performance guarantees in the worst case.

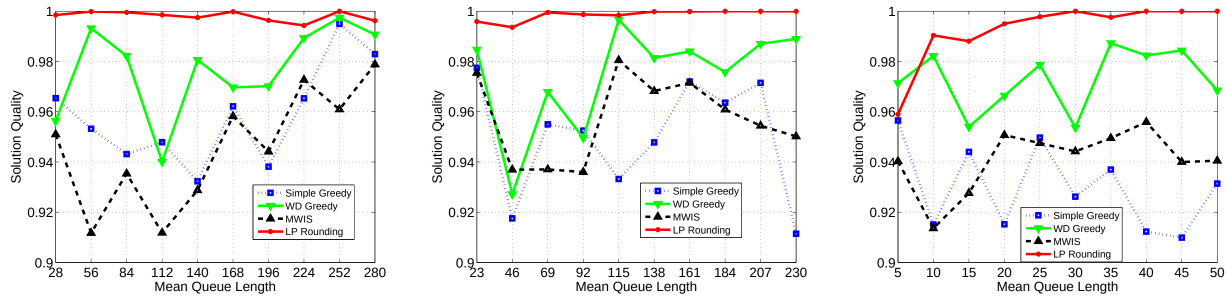
In the scenarios related to Fig. 6, we fixed the mean queue length and varied the network size. In each of such scenarios, we do not show the results acquired by solving the MILP problem because it took a long time for the ILP solver to find optimum solutions in large networks. We found for those small size networks, the utility values given by the LP rounding are very close to the corresponding optimum values. The performance of the proposed algorithms was measured in terms of the utility value. We make the following observations:

1) As in the previous scenarios, the LP rounding algorithm performs the best and all other algorithms have comparable performance. In these scenarios, on average, it improves the utility values by 8.7% compared to the simple greedy algorithm, by 2.8% compared to the weighted degree greedy algorithm, and by 8.2% compared to the MWIS algorithm.

2) The utility value increases with network size because a larger network includes more links and has heavier traffic load.

VIII. Conclusions

We studied a scheduling problem in OFDMA-based wireless relay networks with consideration for multiuser diversity, channel diversity and spatial reuse. First, we presented an MILP formulation to provide optimum solutions. We also showed that the interference degree is at most 4 for any 2-hop relay network and 14 for any general h-hop ($h \geq 2$) relay network. Furthermore, we presented a greedy algorithms for the scheduling problem and showed it has a constant approximation ratio. In addition, we presented three other algorithms, namely, the weighted degree greedy algorithm, the MWIS algorithm and the LP rounding algorithm, to solve the problem. Extensive simulation results showed that the LP rounding algorithm performs best and always

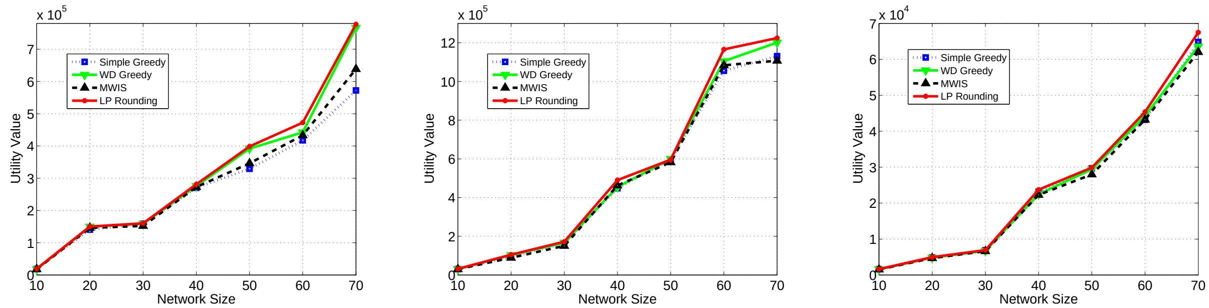


(a) 2-hop networks with 30 nodes and 128 blocks

(b) h-hop networks with 30 nodes and 128 blocks

(c) h-hop networks with 30 nodes and 24 blocks

Fig. 5. Solution quality VS. queue length



(a) 2-hop networks with 128 blocks

(b) h-hop networks with 128 blocks

(c) h-hop networks with 24 blocks

Fig. 6. Utility value VS. network size

provides close-to-optimum solutions. The performance of the simple greedy algorithms is comparable to that of the other algorithms.

REFERENCES

[1] IEEE 802.16e Working Group, Part 16: Air interface for fixed and mobile broadband wireless access systems amendment 2: Physical and Medium Access Control layers for combined fixed and mobile operation in licensed bands, *IEEE Standard*, 2005.

[2] IEEE 802.16j Working Group, Part 16: Air interface for fixed and mobile broadband wireless access systems—multihop relay specification, *IEEE Draft Standard*, 2007.

[3] M. Andrews and L. Zhang, Scheduling algorithms for multi-carrier wireless data systems, *Proceedings of ACM MobiCom'07*, pp. 3–14.

[4] M. Andrews and L. Zhang, Creating templates to achieve low delay in multi-carrier frame-based wireless data systems, *Proceedings of IEEE Infocom'08*, pp. 861–869.

[5] M. S. Bazaraa, J. J. Jarvis and H. D. Sherali, *Linear Programming and Network Flows (3rd edition)*, John Wiley & Sons, 2005.

[6] A. Brzezinski, G. Zussman, and E. Modiano, Enabling distributed throughput maximization in wireless mesh networks—a partitioning approach, *Proceedings of ACM MobiCom'06*, pp. 26–37.

[7] ILOG Software Inc., CPLEX 10.1, http://www.ilog.com

[8] S. Deb, V. Mhatre and V. Ramaiyan, WiMAX relay networks: opportunistic scheduling to exploit multiuser diversity and frequency selectivity, *Proceedings of ACM MobiCom'08*, pp. 163–174.

[9] L. Fu, Z. Cao and P. Fan, Spatial reuse in IEEE 802.16 based wireless mesh networks, *Proceedings of IEEE International Symposium on Communnication & Information Technology*, 2005, pp. 1358–1361.

[10] P. Gupta and P. R. Kumar, The capacity of wireless networks, *IEEE Transactions on Information Theory*, Vol. 46, No. 2, 2000, pp. 388–404.

[11] C. Joo, X. Lin and N. B. Shroff, Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks, *Proceedings of IEEE Infocom'08*, pp. 1103–1111.

[12] D. Kim and A. Ganz, Fair and efficient multihop scheduling algorithm for IEEE 802.16 BWA systems, *Proceedings of IEEE Broadnets'05*, pp. 833–839.

[13] X. Lin and S. Rasool, A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad hoc wireless networks, *Proceedings of IEEE Infocom'07*, pp. 1118–1126.

[14] H. Shetiya and V. Sharma, Algorithms for routing and centralized scheduling to provide QoS in IEEE 802.16 mesh networks, *Proceedings of ACM Workshop on Wireless Multimedia Networking & Performance Modeling*, 2005, pp. 140–149.

[15] K. Sundaresan and S. Rangarajan, On exploiting diversity and spatial reuse in relay-enabled wireless networks, *Proceedings of ACM MobiHoc'08*, pp. 13–22.

[16] K. Sundaresan and S. Rangarajan, Efficient algorithms for leveraging spatial reuse in OFDMA relay networks, Proceedings of Infocom'09, pp. 1539–1547.

[17] W. Wa ng, Y. Wa ng, X-Y. Li, W-Z. Song and O. Frieder, Efficient interference-aware TDMA link scheduling for static wireless networks, *Proceedings of ACM MobiCom'06*, pp. 262–273.

[18] H. Y. Wei, S. Ganguly, R. Izmailov, and Z. Haas, Interference-aware IEEE 802.16 WiMax mesh networks, *Proceedings of IEEE VTC'05*, pp. 3102–3106.

[19] H. Carneiro, J. Oliveira, A. Rodrigues, and P. Sebastiao, Software planning tool for WiMAX networks, *Proceedings of International Conference on Telecommunications*, 2007.

[20] S. Kravitz, Packing cylinders into cylindrical containers, *Mathematics Magazine*, Vol. 40, 1967, pp.65–71.

[21] A. Kako, T. Ono, T. Hirata, and M. M. Halldórsson, Approximation algorithms for the weighted independent set problem in sparse graphs, *Discrete Applied Mathematics*, Vol. 157, No. 4, 2009, pp. 617–626.