3-Approximation Algorithm for Joint Routing and Link Scheduling in Wireless Relay Networks

Chi-Yao Hong, *Student Member, IEEE*, and Ai-Chun Pang, *Member, IEEE*

Abstract—In emerging wireless relay networks (WRNs) such as *IEEE 802.16j*, efficient resource allocation is becoming a substantial issue for throughput optimization. In this paper, we propose an algorithm for joint routing and link scheduling in WRNs. The developed theoretical analysis indicates that the performance of the proposed algorithm is within a factor of three of that of any optimal algorithm in the worst case. Through simulation experiments, the numerical results show that our algorithm outperforms the previously proposed routing and link-scheduling algorithms. Furthermore, the proposed algorithm can effectively achieve near-optimal performance, and provide much better throughput than the theoretical worst-case bound in the average case.

Index Terms—Approximation algorithm, link scheduling, routing, wireless relay networks.

I. INTRODUCTION

RECENTLY, IEEE 802.16j task group has been devoted to the development of multi-hop relaying technology as an enhancement for IEEE 802.16 point-to-multipoint networks. While IEEE 802.16j subscribers are expected to experience performance improvement through low-cost relay stations (RSs), the degradation of system throughput would occur instead if multi-hop radio resources can not be well-utilized. Consequently, the *routing* and *link-scheduling* algorithms, which are of the important resource control functions in multi-hop networks, shall be specifically re-designed to facilitate the effective use of network resources for WRNs.

The link-scheduling issues have been extensively investigated in wireless multi-hop networks (WMNs) with or without quality of service (QoS) consideration. For instance, [1]–[3] target at QoS-sensitive services while the performance of best-effort traffic over WMNs is studied in [4]–[7]. For non-QoS traffic, system throughput is considered as an essential performance criterion. Unfortunately, the problem of link scheduling for system throughput maximization in WMNs was proven to be $\mathcal{NP}$-hard [7]. Due to the intractability of the problem, the previous works have focused on developing approximation algorithms for WMNs [4]–[6]. Besides, it was shown that cross-layer optimization takes the advantages over strictly layered design in mobile ad hoc networks [8]. The joint design

of link scheduling and routing for WMN poses a challenging research problem. Among the research related to join routing and link scheduling, the solution in [9] requires exponential-time complexity with respect to the number of stations, and is intractable for large-scale networks. In [10], Wei *et al.* proposed an interference-aware scheme for joint routing and link scheduling to maximize the system throughput of WiMAX-based WMNs. However, the proposed heuristic scheme can not guarantee the worst-case performance.

The previous schemes for WMNs could not properly applied to WRNs because of some inherent differences of routing and link scheduling for throughput maximization between WMNs and WRNs. The key discrepancies are substantially two-folds and described as follows. 1) In WRNs, RSs do not generate data traffic. This restriction may make the design of routing for throughput maximization in WRNs easier than that in WMNs. On the other hand, the routing problem in WRNs could be more difficult due to its limitation of transmission path of traffic flows. In a WMN, each node can exchange data packets with any other node in the network while a WRN traffic flow is either destined to a base station (BS) or delivered from a BS through one or more RSs. Such a limitation raises the routing complexity and reduces the routing flexibility for throughput maximization. 2) It is generally considered that WMN nodes are equipped with omni-directional antennas to provide a greater coverage area [4], [5], [9], [11]. However, WRN RSs are expected to have high-technology smart antennas with switchable/steerable beam to increase the spectral efficiency [12], [13]. In this case, the inter-station interference can be significantly reduced, and a higher degree of spatial reuse can be achieved. The decreased interference and increased degree of spatial reuse lead to a considerably large set of links that can be activated concurrently in WRNs. Thus the link scheduling problem for throughput maximization in WRNs would be more complex compared with that in WMNs.

Based on the above discussion, it is still an open and interesting question: whether there exists a time-efficient joint routing and link scheduling scheme that provides near-optimal system throughput for WRNs. In this study, we present an algorithm to support interference-aware routing and link scheduling to maximize the system throughput for WRNs. The algorithm accommodates a general workload model for data requests, and fits many other polling-based wireless systems as well as IEEE 802.16 [14]. The achieved throughput of our proposed algorithm is proven to be within a factor of three of that of any optimal solution in the worst case. To the best of our knowledge, the 3-approximation algorithm for joint routing

and link scheduling is one of the first works for WRNs that can guarantee the worst-case performance within a small-constant bound. Through our simulation experiments, the numerical results show that our proposed algorithm outperforms the previously proposed routing and link-scheduling algorithms. Furthermore, the algorithm effectively achieves near-optimal performance, and provides much better throughput than that of the theoretical worst-case bound in the average case.

The remainder of the paper is organized as follows. Section II formally presents our problem formulation. In Section III, we present our joint routing and link-scheduling algorithm. A time complexity analysis and a worst-case performance analysis for the proposed algorithm are also presented. Section IV presents some simulation experiments and numerical results of the proposed algorithm. Finally, the paper is concluded with a brief summary.

II. PROBLEM DEFINITION

This section presents the definition of our joint routing and link scheduling problem. We assume that the WRN under investigation adopts Time Division Multiple Access (TDMA) protocol [14], and accommodates best-effort services in IEEE 802.16j. The data transmission among stations in the WRN is synchronized on frame basis. A frame consists of several time slots which is defined as a basic time unit for transmission in the system. A centralized base station (BS) acts as a coordinator, and provides the contention resolution for data requests of best-effort service flows from serving subscriber stations (SSs). The BS also computes the corresponding transmission routes and schedules for these flows.

In each computation, the WRN is modeled as a simple directed graph $G = (\mathbb{V}, \mathbb{E})$, where the vertices represent the stations including a BS, several RSs and SSs, and the edges denote the wireless links between two stations. The vertex set $\mathbb{V}$ is divided into three disjoint subsets $\mathbb{V}_B, \mathbb{V}_R, \mathbb{V}_S$, and they respectively include a BS, RSs, and SSs. Each SS $n_s \in \mathbb{V}_S$ has a traffic demand p_s. p_s signifies the aggregate uplink traffic request for n_s. For simplicity, only uplink traffic is considered in this study. The downlink traffic can be computed in a similar way, and the details are omitted. A wireless link (i, j) is an element of $\mathbb{E}$ if and only if station n_j is within the maximum transmission range of station n_i. Each edge (i, j) in $\mathbb{E}$ is associated with capacity $r_{i,j}$.

Given the graph G, a *conflict graph* $G^c = (\mathbb{V}^c, \mathbb{E}^c)$ is defined as a simple undirected graph to indicate wireless interference among the stations. Any vertex of G^c bijectively corresponds to an edge in G. Two vertices in $\mathbb{V}^c$ are adjacent if and only if the corresponding two edges in $\mathbb{E}$ are adjacent. This model reflects the *primary interference*, which captures the interference when a station 1) transmits packets to two different stations simultaneously, 2) receives packets from two different stations concurrently, or 3) transmits and receives packets at the same time. Another type of interference, namely, the *secondary interference*, occurs when a receiving station is interfered by other transmissions which are not intended for this station. We assume that the secondary interference can be avoided for the WRNs under investigation through the beam and null steering capability of smart antennas.

Based on the above definition of the problem, our proposed solution for joint routing and link scheduling is developed to produce a transmission schedule such that the schedule is interference-free and the system throughput is maximized. Note that the throughput maximization problem in our study is equivalent to the minimization of the schedule length in terms of time-slots because the achieved system throughput is inversely proportional to the schedule length [4], [5], [7], [11].

III. A JOINT ROUTING AND LINK-SCHEDULING ALGORITHM

This section elaborates on our routing and link-scheduling algorithm to achieve the maximal system throughput for WRNs.

A. Linear Programming Based Routing

For any edge $(i, j) \in \mathbb{E}$, let $f_{i,j}$ be the aggregate traffic transmitted from station n_i to station n_j. The total time of station n_i for transmitting and receiving the data is denoted by $\rho_i = \left(\sum_{(i,j) \in \mathbb{E}} \frac{f_{i,j}}{r_{i,j}} \right) + \left(\sum_{(j,i) \in \mathbb{E}} \frac{f_{j,i}}{r_{j,i}} \right)$. Due to the primary interference, any two transmissions that are originated/destined from/to a station can not be overlapped, and the schedule length would be limited by ρ_i. Consider a routing with a large ρ_i. The schedule length is inevitably large, regardless of which link scheduler is adopted. In order to minimize the schedule length, we shall find a routing such that $\max_{n_i \in \mathbb{V}} \{ \rho_i \}$ is minimized. To achieve this, the linear programming (LP) technique is used, and its objective function is

$$\textit{minimize} \quad \beta \quad (1a)$$

where

$$\beta \geq \rho_i, \quad \forall n_i \in \mathbb{V}. \quad (1b)$$

For any RS n_j, the flow conservation constraint is

$$\sum_{(r,j) \in E} f_{r,j} = \sum_{(j,r) \in E} f_{j,r}, \quad \forall n_r \in \mathbb{V}_r. \quad (1c)$$

Each SS has no incoming traffic, *i.e.*,

$$f_{i,s} = 0, \quad \forall n_s \in \mathbb{V}_s, \forall n_i \in \mathbb{V}, \quad (1d)$$

and each BS has no outgoing traffic, *i.e.*,

$$f_{b,i} = 0, \quad \forall n_b \in \mathbb{V}_b, \forall n_i \in \mathbb{V}. \quad (1e)$$

Furthermore, the nonnegative flow constraint is

$$f_{i,j} \geq 0, \quad \forall (i, j) \in \mathbb{E}. \quad (1f)$$

The traffic demand constraint p_s for each SS n_s

$$\sum_{(s,i) \in \mathbb{E}} f_{s,i} \geq p_s, \quad \forall n_i \in \mathbb{V}_s \quad (1g)$$

Note that we make no assumption on the number of routing paths for any subscriber. That is, the traffic load from each SS could be distributed to multiple routes to alleviate the network congestion.

B. Makespan Link Scheduling

Once the routing is determined, this section shows how we efficiently schedule the link traffic so that the schedule length can be minimized and the system throughput can be accordingly increased. Before presenting our scheduling algorithm, we define several terms used in this section. For each edge $(i, j) \in \mathbb{E}$, $t_{i,j}$ represents the number of time-slots for transmitting $f_{i,j}$, i.e., $t_{i,j} = f_{i,j}/r_{i,j}$. Since each link in $\mathbb{E}$ corresponds to a vertex in $\mathbb{V}^c$, $t_{i,j}$ can be bijective-mapped to $\ell^c(i)$ for any vertex n_i^c in $\mathbb{V}^c$. The definition of $\ell^c(i)$ corresponding to vertex n_i^c in $\mathbb{V}^c$ is equivalent to that of $t_{i,j}$ for edge $(i, j) \in \mathbb{E}$.

For any vertex n_i^c in $\mathbb{V}^c$, we define that $\xi^c(i)$ is the total number of time-slots required by n_i^c and its neighbors to fulfill their traffic demands, i.e.,

$$\xi^c(i) = \ell^c(i) + \left(\sum_{(i,j) \in \mathbb{E}^c} \ell^c(j) \right). \tag{2}$$

For any time-slot τ, $\mathbb{I}_\tau$ denotes a set of the links that are scheduled to transmit the data in time-slot τ. In order to avoid the collision resulting from the primary interference, $\mathbb{I}_\tau$ shall be an independent set of vertices in G^c.

Given a flow requirement $f_{i,j}$ of each link in G, we present an Interference-aware Makespan Scheduling (IMS) algorithm to minimize the schedule length for the traffic flows. As shown in Algorithm 1, the inputs include the conflict graph G^c, and the corresponding $\ell^c(i)$ and $\xi^c(i)$ for each vertex n_i^c in V^c. Initially, G' is set to G^c for the operations of the following loop. For the τth time slot (see the outer **repeat** loop), some proper vertices in G^c will be selected into $\mathbb{I}_\tau$. The selection is based on their up-to-date values of ξ', and the process is shown in the inner **repeat** loop of lines 6-15. $G^{(\tau)}$ in line 4 will be used for the worst-case analysis in the following section, and is not discussed here. Let $\wp'(j)$ be a flag to indicate whether vertex $n_j \in \mathbb{V}'$ is able to be scheduled into $\mathbb{I}_\tau$ due to the primary interference. Initially, all vertices in $\mathbb{V}'$ could be scheduled, and their $\wp'$ is set to 0. In each iteration of the inner **repeat** loop, a vertex in V' with minimal ξ' and its $\wp' = 0$ is selected into $\mathbb{I}_\tau$. Then the $\wp'$ values of the vertex and its neighbors are set by 1 for collision avoidance. Also, if vertex $n_i' \in \mathbb{V}'$ has been satisfied with its scheduled time slots, i.e., $\ell'(i) = 0$, it will be removed from V'. The inner loop will be performed until $\wp' = 1$ for all vertices in $\mathbb{V}'$. Then the link-scheduling operation for the following time slot will be activated. The outer **repeat** loop terminates when all link flows are properly scheduled (i.e., $|V'| = 0$). Through our Algorithm IMS, a collision-free makespan schedule $\mathbb{I}_1, \mathbb{I}_2, \ldots, \mathbb{I}_\kappa$ will be produced, where κ is the schedule length in the unit of time-slots.

C. Time Complexity Analysis

Then we analyze the time complexity of our proposed joint routing and link-scheduling algorithm. The proposed LP-based routing algorithm can be solved in polynomial time by the well-known projective method [15]. Let $\triangle(G^c)$ be the maximal degree of vertices in G^c. The time complexity of scheduling each time-slot in Algorithm IMS is $O(|\mathbb{E}| \times \max\{\lg |\mathbb{E}|, \triangle(G^c)\})$, because choosing a vertex $n_i' \in \mathbb{V}'$ takes $O(\max\{\lg |\mathbb{E}|, \triangle(G^c)\})$
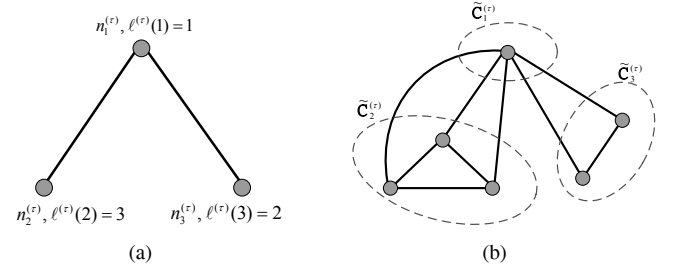

Fig. 1. An example of graph reduction from (a) $G^{(\tau)}$ to (b) $\tilde{G}^{(\tau)}$

time and at most $|\mathbb{E}|$ links could be scheduled to the slot. Thus Algorithm IMS requires $O(\kappa |\mathbb{E}| \times \max\{\lg |\mathbb{E}|, \triangle(G^c)\})$ computations. Since $\triangle(G^c) \leq |\mathbb{E}|$, the worst-case running time of Algorithm IMS will be $O(\kappa |\mathbb{E}|^2)$, which is acceptable in a WMN/WRN network [4], [5], [10].

Algorithm 1 IMS

Require: $G^c = (\mathbb{V}^c, \mathbb{E}^c)$
Ensure: $\mathbb{I}_1, \mathbb{I}_2, \ldots$
1: $G' \leftarrow G^c$ ($\mathbb{V}' \leftarrow \mathbb{V}^c, \mathbb{E}' \leftarrow \mathbb{E}^c, \ell'(i) \leftarrow \ell^c(i)$, and $\xi'(i) \leftarrow \xi^c(i)$ for all vertex n_i^c in $\mathbb{V}^c$)
2: $\tau \leftarrow 1$
3: **repeat**
4:      $G^{(\tau)} \leftarrow G'$
5:      $\wp'(j) \leftarrow 0$ for all vertices n_j in $\mathbb{V}'$
6:      **repeat**
7:          $n_i \leftarrow \arg\min_{n_j \in \mathbb{V}', \wp'(j)=0} \xi'(j)$
8:          $\mathbb{I}_\tau \leftarrow \mathbb{I}_\tau \cup \{n_i\}$
9:          $\wp'(j) \leftarrow 1$ for all vertices $n_j \in \{n_i\} \cup \{n_k | (i, k) \in \mathbb{E}'\}$
10:         $\ell'(i) \leftarrow \ell'(i) - 1$
11:         **if** $\ell'(i) = 0$ **then**
12:             $\mathbb{V}' \leftarrow \mathbb{V}' \setminus \{n_i\}$
13:         **end if**
14:         update $\xi'(j)$ for all vertices $n_j \in \{n_i\} \cup \{n_k | (i, k) \in \mathbb{E}'\}$ according to (2)
15:      **until** $\wp'(j) = 1$ for all vertices $n_j \in \mathbb{V}'$
16:      $\tau \leftarrow \tau + 1$
17: **until** $|\mathbb{V}'| = 0$
18: $\kappa \leftarrow \tau$
19: **return** $\mathbb{I}_1, \mathbb{I}_2, \ldots, \mathbb{I}_\kappa$

D. Worst-Case Performance Analysis

In this section, a worst-case analysis for system throughput of our routing and link scheduling algorithm against any of optimal solutions is presented. For any $G^{(\tau)}$ ($1 \leq \tau \leq \kappa$), we construct a simple undirected graph $\tilde{G}^{(\tau)} = (\tilde{\mathbb{V}}^{(\tau)}, \tilde{\mathbb{E}}^{(\tau)})$ such that for each vertex $n_i^{(\tau)}$ in $\mathbb{V}^{(\tau)}$, a *clique* $\tilde{\mathbf{C}}_i^{(\tau)}$ with $\ell^{(\tau)}(i)$ fully-connected vertices is added into $\tilde{G}^{(\tau)}$. For any two different cliques, say $\tilde{\mathbf{C}}_i^{(\tau)}$ and $\tilde{\mathbf{C}}_j^{(\tau)}$, $\tilde{\mathbf{C}}_i^{(\tau)}$ is fully connected to $\tilde{\mathbf{C}}_j^{(\tau)}$ if (i, j) is in $\mathbb{E}^{(\tau)}$. If (i, j) is not in $\mathbb{E}^{(\tau)}$, $\tilde{\mathbf{C}}_i^{(\tau)}$ and $\tilde{\mathbf{C}}_j^{(\tau)}$ are disconnected. An example of the construction is shown in Figure 1, and the following facts are observed for the construction.

Fact 1: $|\tilde{\mathbb{V}}^{(\tau)}| = \sum_{n_i^{(\tau)} \in \mathbb{V}^{(\tau)}} \ell^{(\tau)}(i) = |\mathbb{V}^{(\tau)}| \times \overline{\ell^{(\tau)}}$, where $\overline{\ell^{(\tau)}} = \left(\sum_{n_i^{(\tau)} \in \mathbb{V}^{(\tau)}} \ell^{(\tau)}(i) \right) / |\mathbb{V}^{(\tau)}|$ is the average traffic demand of $\ell^{(\tau)}(i)$.

Fact 2: $\forall \tilde{n}_i^{(\tau)} \in \tilde{\mathbf{C}}_j^{(\tau)} \in \tilde{\mathbb{V}}^{(\tau)}, \tilde{d}_i^{(\tau)} = \ell^{(\tau)}(j) - 1 + \sum_{(j,k)\in\mathbb{E}^{(\tau)}} \ell^{(\tau)}(k)$, where $\tilde{d}_i^{(\tau)}$ represents the vertex degree of $\tilde{n}_i^{(\tau)}$.

Fact 3: $\tilde{\delta}^{(\tau)} = \left(\sum_{\tilde{n}_i^{(\tau)}\in\tilde{\mathbb{V}}^{(\tau)}} \tilde{d}_i^{(\tau)}\right)/|\tilde{\mathbb{V}}^{(\tau)}| = 2|\tilde{\mathbb{E}}^{(\tau)}|/|\tilde{\mathbb{V}}^{(\tau)}|$, where $\tilde{\delta}^{(\tau)}$ represents the average vertex degree in $\tilde{\mathbb{V}}^{(\tau)}$.

Based on the facts, the upper bound of $\tilde{\delta}^{(\tau)}$ in terms of $\ell^{(\tau)}(j)$ and $\ell^{(\tau)}$ will be derived in the following theorem.

Theorem 1: $\tilde{\delta}^{(\tau)} \le \max_{n_i^{(\tau)}\in\mathbb{V}^{(\tau)}}\left\{\sum_{(i,k)\in\mathbb{E}^{(\tau)}} \ell^{(\tau)}(k)\right\} + \ell^{(\tau)} - 1$.

Proof:

$$
\begin{aligned}
\tilde{\delta}^{(\tau)} &= \frac{\sum_{\tilde{n}_i^{(\tau)}\in\mathbf{C}_j\in\tilde{\mathbb{V}}^{(\tau)}} \tilde{d}_i^{(\tau)}}{|\tilde{\mathbb{V}}^{(\tau)}|} \quad \text{(by Fact 3)} \\
&= \left[\frac{\sum_{\tilde{n}_i^{(\tau)}\in\mathbf{C}_j\in\tilde{\mathbb{V}}^{(\tau)}} \sum_{(j,k)\in\mathbb{E}^{(\tau)}} \ell^{(\tau)}(k)}{|\tilde{\mathbb{V}}^{(\tau)}|}\right] \\
&\quad + \left[\frac{\sum_{\tilde{n}_i^{(\tau)}\in\mathbf{C}_j\in\tilde{\mathbb{V}}^{(\tau)}} [\ell^{(\tau)}(j) - 1]}{|\tilde{\mathbb{V}}^{(\tau)}|}\right] \quad \text{(by Fact 2)} \\
&= \left\{\frac{\left[\sum_{n_j^{(\tau)}\in\mathbb{V}^{(\tau)}} \ell^{(\tau)}(j) \times \sum_{(j,k)\in\mathbb{E}^{(\tau)}} \ell^{(\tau)}(k)\right]}{\ell^{(\tau)}|\mathbb{V}^{(\tau)}|}\right\} + \ell^{(\tau)} - 1 \quad \text{(by Fact 1)} \\
&\le \left\{\frac{\left[\sum_{n_j^{(\tau)}\in\mathbb{V}^{(\tau)}} \ell^{(\tau)}(j) \times \max_{n_i^{(\tau)}\in\mathbb{V}^{(\tau)}}\left\{\sum_{(i,k)\in\mathbb{E}^{(\tau)}} \ell(k)\right\}\right]}{\ell^{(\tau)}|\mathbb{V}^{(\tau)}|}\right\} + \ell^{(\tau)} - 1 \\
&= \max_{n_i^{(\tau)}\in\mathbb{V}^{(\tau)}}\left\{\sum_{(i,k)\in\mathbb{E}^{(\tau)}} \ell^{(\tau)}(k)\right\} + \ell^{(\tau)} - 1 \quad \text{(by Fact 1)}
\end{aligned}
$$
∎

In addition to the observations from the $\tilde{G}^{(\tau)}$ construction, the following facts are observed for our Algorithm IMS.

Fact 4: $\tilde{d}_i^{(\tau)} \le \tilde{d}_i^{(\tau-1)} - 1$ for $\tau = 2,\ldots,\kappa$, where $d_i^{(\tau)}$ represents the vertex degree of $n_i^{(\tau)}$ in $\mathbb{V}^{(\tau)}$.

Corollary 1: $\tilde{\delta}^{(\tau)} \le \tilde{\delta}^{(\tau-1)} - 1$ for $\tau = 2,\ldots,\kappa$. (by Fact 3 and Fact 4)

Fact 5: $\sum_{n_i^{(\tau)}\in\mathbb{V}^{(\tau)}} \ell^{(\tau)}(i) = \sum_{n_i^{(\tau+1)}\in\mathbb{V}^{(\tau+1)}} \ell^{(\tau+1)}(i) + |\mathbb{I}_\tau|$ for $\tau = 1,\ldots,\kappa - 1$.

Corollary 2: $|\tilde{\mathbb{V}}^{(\tau)}| = |\tilde{\mathbb{V}}^{(\tau+1)}| + |\mathbb{I}_\tau|$ for $\tau = 1,\ldots,\kappa - 1$. (by Fact 1 and Fact 5)

Since Algorithm IMS selects an independent set $\mathbb{I}_\tau$ of vertices in $G^{(\tau)}$ for each time slot τ, by [16], (2) and Fact 2, the cardinality $|\mathbb{I}_\tau|$ satisfies

$$
|\mathbb{I}_\tau| \ge \sum_{\tilde{n}_i^{(\tau)}\in\tilde{\mathbb{V}}^{(\tau)}} \frac{1}{1 + \tilde{d}_i^{(\tau)}} \tag{3}
$$

For a sequence of positive numbers, their arithmetic mean can not be less than the harmonic mean. Thus we have

$$
\frac{\sum_{\tilde{n}_i^{(\tau)}\in\tilde{\mathbb{V}}^{(\tau)}} 1 + \tilde{d}_i^{(\tau)}}{|\tilde{\mathbb{V}}^{(\tau)}|} \ge \frac{|\tilde{\mathbb{V}}^{(\tau)}|}{\sum_{\tilde{n}_i^{(\tau)}\in\tilde{\mathbb{V}}^{(\tau)}} \frac{1}{1+\tilde{d}_i^{(\tau)}}}. \tag{4}
$$

Substituting (4) into (3), we have

$$
\begin{aligned}
|\mathbb{I}_\tau| &\ge \frac{|\tilde{\mathbb{V}}^{(\tau)}|^2}{\sum_{\tilde{n}_i^{(\tau)}\in\tilde{\mathbb{V}}^{(\tau)}} 1 + \tilde{d}_i^{(\tau)}} \\
&= \frac{|\tilde{\mathbb{V}}^{(\tau)}|}{1 + \tilde{\delta}^{(\tau)}} \quad \text{(by Fact 3)}
\end{aligned} \tag{5}
$$

Let $\Psi_\tau = \sum_{1\le i\le\tau} |\mathbb{I}_i|$ be the scheduled traffic demand (in the unit of time-slots) in the first τ slots for $1 \le \tau \le \kappa$ and $\Psi_0 = 0$. Then we obtain

$$
\begin{aligned}
\Psi_\tau &= \Psi_{\tau-1} + |\mathbb{I}_\tau| \\
&\ge \Psi_{\tau-1} + \frac{|\tilde{\mathbb{V}}^{(\tau)}|}{1 + \tilde{\delta}^{(\tau)}} \quad \text{(by (5))} \\
&\ge \Psi_{\tau-1} + \frac{|\tilde{\mathbb{V}}^{(\tau)}|}{\tilde{\delta}^{(1)} - \tau + 2} \quad \text{(by Corollary 1)} \\
&= \Psi_{\tau-1} + \frac{|\tilde{\mathbb{V}}^{(1)}| - \Psi_{\tau-1}}{\tilde{\delta}^{(1)} - \tau + 2} \quad \text{(by Corollary 2)}
\end{aligned} \tag{6}
$$

From (6), we observe that $\Psi_1 \ge |\tilde{\mathbb{V}}^{(1)}|/(\tilde{\delta}^{(1)} + 1)$. We will prove that $\Psi_\tau \ge \tau\left[\frac{|\tilde{\mathbb{V}}^{(1)}|}{(\tilde{\delta}^{(1)}+1)}\right]$ by induction on τ.

$$
\begin{aligned}
\Psi_\tau &\ge \Psi_{\tau-1} + \frac{|\tilde{\mathbb{V}}^{(1)}| - \Psi_{\tau-1}}{\tilde{\delta}^{(1)} - \tau + 2} \\
&\ge \frac{|\tilde{\mathbb{V}}^{(1)}| \times (\tau - 1)}{\tilde{\delta}^{(1)} + 1} + \frac{|\tilde{\mathbb{V}}^{(1)}| - \frac{|\tilde{\mathbb{V}}^{(1)}|\times(\tau-1)}{\tilde{\delta}^{(1)}+1}}{\tilde{\delta}^{(1)} - \tau + 2} \\
&= \frac{|\tilde{\mathbb{V}}^{(1)}| \times \tau}{\tilde{\delta}^{(1)} + 1}, \quad \text{for } \tau \ge 1
\end{aligned} \tag{7}
$$

Substituting $\Psi_\kappa = |\tilde{\mathbb{V}}^{(1)}|$ into (7), we find that the schedule length κ of Algorithm IMS can be

$$
\begin{aligned}
\kappa &\le \tilde{\delta}^{(1)} + 1 \tag{8} \\
&\le \max_{n_i^{(1)}\in\mathbb{V}^{(1)}}\left\{\sum_{(i,j)\in\mathbb{E}^{(1)}} \ell^{(1)}(j)\right\} + \ell^{(1)} \quad \text{(By Theorem 1)} \\
&= \max_{n_i^c\in\mathbb{V}^c}\left\{\sum_{(i,j)\in\mathbb{E}^c} \ell^c(j)\right\} + \ell^c \tag{9}
\end{aligned}
$$

slots.

Theorem 2: Our joint routing and link-scheduling algorithm is a 3-approximation algorithm.

Proof: Assume that an "optimum" algorithm for routing and link scheduling outputs the smallest schedule length Φ, and Φ can be derived as

$$
\Phi \ge \max_{n_i\in\mathbb{V}}\left\{\left(\sum_{(i,j)\in\mathbb{B}} \frac{f_{i,j}^*}{r_{i,j}}\right) + \left(\sum_{(j,i)\in\mathbb{B}} \frac{f_{j,i}^*}{r_{j,i}}\right)\right\} \tag{10}
$$

where $f_{i,j}^*$ is the aggregate traffic from n_i to n_j produced by the optimum algorithm. Since our LP-based routing provides an optimal solution to minimize $\max_{n_i\in\mathbb{V}}\left\{\left(\sum_{(i,j)\in\mathbb{B}} \frac{f_{i,j}}{r_{i,j}}\right) + \left(\sum_{(j,i)\in\mathbb{B}} \frac{f_{j,i}}{r_{j,i}}\right)\right\}$, (10) can be rewritten as

$$
\begin{aligned}
\Phi &\ge \max_{n_i\in\mathbb{V}}\left\{\left(\sum_{(i,j)\in\mathbb{B}} \frac{f_{i,j}}{r_{i,j}}\right) + \left(\sum_{(j,i)\in\mathbb{B}} \frac{f_{j,i}}{r_{j,i}}\right)\right\} \\
&= \max_{n_i\in\mathbb{V}}\left\{\left(\sum_{(i,j)\in\mathbb{B}} t_{i,j}\right) + \left(\sum_{(j,i)\in\mathbb{B}} t_{j,i}\right)\right\} \\
&\ge \frac{1}{2}\max_{(i,j)\in\mathbb{B}}\left\{\left(\sum_{(i,i')\in\mathbb{B}} t_{i,i'}\right) + \left(\sum_{(i',i)\in\mathbb{B}} t_{i',i}\right) + \left(\sum_{(j,j')\in\mathbb{B}} t_{j,j'}\right) + \left(\sum_{(j',j)\in\mathbb{B}} t_{j',j}\right)\right\} \\
&\ge \frac{1}{2}\max_{n_i^c\in\mathbb{V}^c}\left\{2\ell^c(i) + \sum_{(i,j)\in\mathbb{E}^c} \ell^c(j)\right\} \tag{11}
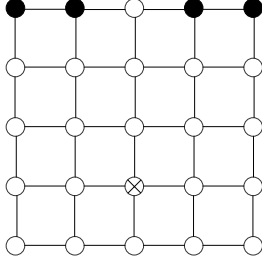\end{aligned}
$$

Fig. 2.   Topology used in the experiment

By (9) and (11), the approximation factor ϵ of our proposed algorithm can be derived as

$$\epsilon \;\leq\; 2 \times \frac{\max_{n_i^c \in \mathbb{V}^c}\left\{\sum_{(i,j)\in\mathbb{E}^c} \ell^c(j)\right\} + \ell^c}{\max_{n_i^c \in \mathbb{V}^c}\left\{2\ell^c(i) + \sum_{(i,j)\in\mathbb{E}^c} \ell^c(j)\right\}} \qquad (12)$$

$$\leq\; 2 \times \frac{\max_{n_i^c \in \mathbb{V}^c}\left\{\sum_{(i,j)\in\mathbb{E}^c} \ell^c(j)\right\} + \ell^c}{\max_{n_i^c \in \mathbb{V}^c}\left\{\sum_{(i,j)\in\mathbb{E}^c} \ell^c(j)\right\} + 2\ell^c_{\min}} \qquad (13)$$

where $\ell^c_{\min} = \min_{n_i^c \in \mathbb{V}^c}\{\ell^c(i)\}$. Note that when $\ell^c \leq 2\ell^c_{\min}$, our algorithm has an approximation factor of 2. In a general case, (12) can be rewritten as

$$\begin{aligned}\epsilon &\leq 2 \times \left(1 + \frac{\ell^c}{2\ell^c_{\max} + \ell^c_{\min}}\right) \\ &= 2 \times \left[1.5 - \frac{\ell^c_{\max} + (\ell^c_{\min}/2) - \ell^c}{2\ell^c_{\max} + \ell^c_{\min}}\right] \\ &= 3 - \left[2 \times \frac{\ell^c_{\max} + (\ell^c_{\min}/2) - \ell^c}{2\ell^c_{\max} + \ell^c_{\min}}\right] \qquad (14)\end{aligned}$$

where $\ell^c_{\max} = \max_{n_i^c \in \mathbb{V}^c}\{\ell^c(i)\}$. Observe that $\frac{\ell^c_{\max} + (\ell^c_{\min}/2) - \ell^c}{2\ell^c_{\max} + \ell^c_{\min}} > 0$ completes the proof.   ∎

IV. PERFORMANCE EVALUATION

This section investigates the performance of the proposed algorithm through our developed Monte Carlo simulation. We use LINGO to solve the LP-based routing problem [17], and Dijkstra's shortest-path algorithm with respect to the number of hop counts is adopted as the compared routing algorithm in the experiments. The centralized scheduling algorithm (Cent) proposed in [5] is used for comparison against our IMS. For the joint design of routing and link scheduling, the heuristic (HEU) in [10] is used for comparison with our joint algorithm.

We employ the Gamma distribution to approximate the traffic demand p_s for each subscriber $n_s \in \mathbb{V}_S$ in G, where the mean and variance of p_s are denoted by p_{mean} (Mbit) and p_{var} (Mbit²), respectively. As shown in Figure 2, a 5×5 grid topology is adopted to simulate a Manhattan-like urban environment [18], where the filled, blank and "⊗" circles represent the SSs, RSs and BS, respectively. The data rates of relay links (RS-to-BS or RS-to-RS) and access links (SS-to-BS or SS-to-RS) are set by 18.36 Mbps and 6 Mbps, which are the raw data rates of adopting the modulation scheme of quadrature phase-shift keying (QPSK) and 64-quadrature amplitude modulation (64-QAM) in IEEE 802.16 orthogonal frequency-division multiplexing (OFDM) [19].
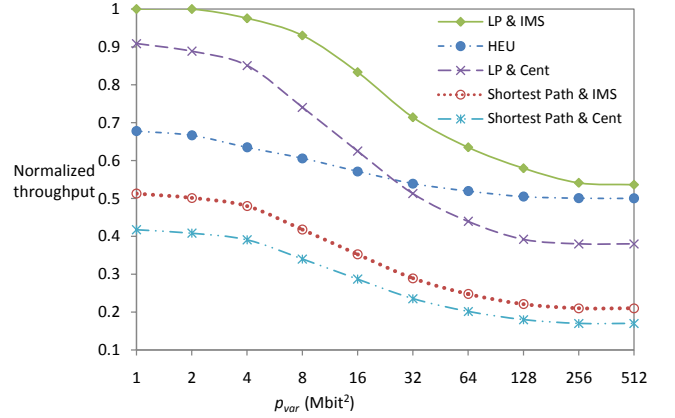

Fig. 3.   Effects of p_{shape} on the normalized throughput

By setting $p_{mean} = 1$ Mbit, Figure 3 shows the system throughput for our joint routing and link-scheduling algorithm as well as the compared algorithms. The throughput is normalized to an upper bound, and the upper bound can be obtained as follows. Due to the primary interference, any two links incident to the base station can not be scheduled simultaneously. Thus the system throughput is bounded to the data rate of relay links, i.e., 18.36 Mbps. From Figure 3, we observe that the normalized throughput of our joint routing and link-scheduling algorithm (LP & IMS) is much better than those compared algorithms for all p_{var} under investigation. Specifically, the curves indicate that the normalized throughput of Algorithm Cent (with LP) is less than about 10% of that of our IMS (with LP). They also indicate that our LP-based routing provides at least a double of the throughput of "Shortest Path" (with IMS). For the joint design of routing and link scheduling, Figure 3 shows that our LP&IMS outperforms HEU especially for small p_{var}. As p_{var} is small, another important phenomenon is observed that the throughput of our algorithm is very close to the theoretical upper bound.

Note that the derived upper bound could be much higher than the optimum solution since the bound might not be tight. It is suffice to show that the proposed algorithm yields the throughput that would be pretty close to the optimal solution for most of p_{var} values under investigation. Although our LP&IMS is proven to be a 3-approximation algorithm in the worst case, the simulation results show that the throughput in the average case is much higher than that derived in the worst-case analysis.

As to the effects of p_{var}, we observe that the normalized throughput of all algorithms under investigation decreases as p_{var} increases. The reason is that as the variance of traffic demand of each SS increases, the load of each link is difficult to balance. To avoid collision, the unbalanced links cannot be scheduled simultaneously, and thus the lower spatial reuse leads to a lower throughput. However, when p_{var} becomes large, our IMS provides more effective link-scheduling capability compared to Cent. For example, the normalized throughput of Algorithm Cent (with LP) is about 90% and 68% of that of our IMS (with LP) as $p_{var} = 1$ and 128, respectively. Furthermore, we observe that the performance of HEU is relatively insensitive to p_{var}, and its throughput

becomes close to that of our proposed algorithm for large p_{var}.

V. CONCLUSION

This paper proposed a joint routing and link-scheduling algorithm for TDMA-based wireless relay networks. The performance of the proposed algorithm was investigated through our developed theoretical analysis and simulation experiments. The theoretical analysis indicated that the performance of the proposed algorithm is within a factor of three of that of any optimal algorithm even in the worst case. Through simulation experiments, the numerical results demonstrated that our algorithm outperforms the previously proposed routing and link-scheduling algorithms under investigation. Furthermore, we observed that our algorithm can effectively achieve near-optimal performance, and provide much better throughput than that of the theoretical worst-case bound in the average case.

REFERENCES

[1] P. Djukic and S. Valaee, "Link scheduling for minimum delay in spatial re-use tdma," in *Proc. IEEE INFOCOM'07*, May 2007.

[2] S. Harish and S. Vinod, "Algorithms for routing and centralized scheduling to provide qos in ieee 802.16 mesh networks," in *Proc. 1st ACM workshop on Wireless multimedia networking and performance modeling*, Oct. 2005, pp. 140-149.

[3] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and QoS routing in multi-channel wireless mesh networks," in *Proc. ACM International Symposium Mobile Ad Hoc Networking Computing (MobiHoc'05)*, May 2005, pp. 68-77.

[4] G. Brar, D. M. Blough, and P. Santi, "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks," in *Proc. ACM MOBICOM'06*, Sept. 2006, pp. 2-13.

[5] W. Wang, Y. Wang, X.-Y. Li, W.-Z. Song, and O. Frieder, "Efficient interference-aware TDMA link scheduling for static wireless networks," in *Proc. ACM MOBICOM'06*, Sept. 2006, pp. 262-273.

[6] X. Lin and S. Rasool, "A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad hoc wireless networks," in *Proc. IEEE INFOCOM'07*, May 2007.

[7] E. Arikan, "Some complexity results about packet radio networks," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 910-918, July 1984.

[8] M. Conti *et al*, "Cross-layering in mobile ad hoc network design," *IEEE Commun. Mag.*, vol. 37, no. 2, pp. 48-51, Feb. 2004.

[9] H. Viswanathan and S. Mukherjee, "Throughput-range tradeoff of wireless mesh backhaul networks," *IEEE J. Select. Areas Commun.*, vol. 24, pp. 593-602, Mar. 2006.

[10] H. Y. Wei, S. Ganguly, R. Izmailov, and Z. J. Haas, "Interference-aware IEEE 802.16 WiMax mesh networks," in *Proc. IEEE Veh. Technol. Conf. (VTC Spring'04)*, June 2005, pp. 3102-3106.

[11] M. Alicherry, R. Bhatia, and L. E. Li, "Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks," *IEEE J. Select. Areas Commun.*, vol. 24, pp. 1960-1971, Nov. 2006.

[12] R. Pabst *et. al*, "Relay-based deployment concepts for wireless and mobile broadband radio," *IEEE Commun. Mag.*, vol. 42, no. 9, pp. 80-89, Sept. 2004.

[13] Y. Fan and J. Thompson, "MIMO configurations for relay channels: theory and practice," *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, pp. 1774-1786, May 2007.

[14] Air Interface for Fixed Broadband Wireless Access Systems, IEEE Std. 802.16-2004, 2004.

[15] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, no. 4, pp. 373-395, 1984.

[16] S. Sakai, M. Togasaki, and K. Yamazaki, "A note on greedy algorithms for the maximum weighted independent set problem," *Discrete Applied Mathematics*, vol. 126, pp. 313-322, 2003.

[17] L. Systems. LINGO. [Online]. Available: http://www.lindo.com/

[18] J. Cho, M. Lee, H. Kang, S. Lee, J. Son, and P. Joo, "On the throughput enhancement of fixed relay concept in Manhattan-like urban environments," IEEE C802.16mmr-06/003, Jan. 2006.

[19] A. Ghosh, D. R. Wolter, J. G. Andrews, and R. Chen, "Broadband wireless access with WiMax/802.16: current performance benchmarks and future potential," *IEEE Commun. Mag.*, vol. 43, pp. 129-136, Feb. 2005.

Chi-Yao Hong received his junior B.S. degree in Electronic Engineering from National Ilan Institute of Technology, Taiwan, in 2004, and B.S. degree in Electronic Engineering from National Taiwan University of Science and Technology, Taiwan, in 2006. He is currently working toward the M.S. degree in Computer Science, National Taiwan University, Taiwan. His current research interests are focused on wireless multihop networks, especially in QoS provisioning issues.

Ai-Chun Pang received the B.S., M.S. and Ph.D. degrees in Computer Science and Information Engineering from National Chiao Tung University, Taiwan, in 1996, 1998 and 2002, respectively. She joined the Department of Computer Science and Information Engineering (CSIE), National Taiwan University (NTU), Taipei, Taiwan, as an Assistant Professor in 2002. From August 2004 to July 2005, Dr. Pang served as an Assistant Professor in Graduate Institute of Networking and Multimedia (INM), and an adjunct Assistant Professor in CSIE/NTU, Taipei, Taiwan. Currently, she is an Associate Professor in INM and CSIE of NTU, Taipei, Taiwan. Her research interests include design and analysis of personal communications services network, mobile computing, voice over IP and performance modeling.

Dr. Pang has served as the program co-chairs and committees of many international conferences/workshops. She was the guest editor of IEEE WIRELESS COMMUNICATIONS, and is currently an associate editor of INTERNATIONAL JOURNAL OF SENSOR NETWORKS and ACM WIRELESS NETWORKS. Also, Dr. Pang was a recipient of the Teaching Award at NTU in 2005, 2006 and 2007, received the Investigative Research Award from the Pan Wen Yuan Foundation in 2006, received Wu Ta You Memorial Award from National Science Council (NSC) in 2007, received Excellent Young Engineer Award from the Chinese Institute of Electrical Engineering, and received K. T. Li Award for Young Researchers from ACM Taipei/Taiwan Chapter in 2007.