Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 7985756, 11 pages
https://doi.org/10.1155/2018/7985756

Research Article

Optimal Multicommodity Spectrum-Efficient Routing in Multihop Wireless Networks

Mohamed Saad

Department of Electrical and Computer Engineering, University of Sharjah, Sharjah, UAE

Correspondence should be addressed to Mohamed Saad; msaad@sharjah.ac.ae

Received 16 April 2018; Accepted 13 June 2018; Published 9 July 2018

Academic Editor: Jose M. Gimenez-Guzman

Finding the route with maximum end-to-end spectral efficiency in multihop wireless networks has been subject to interest in the recent literature. All previous studies, however, focused on finding *one* route from a given source to a given destination under the constraint of equal bandwidth sharing. To the best of our knowledge, for the first time, this paper provides extensions to the multicommodity flow case, i.e., the case of multiple simultaneous source-destination (*s*-*d*) pairs. In particular, given an arbitrary number of *s*-*d* pairs, we address the problem of finding a route for every *s*-*d* pair such that the minimum spectral efficiency across all routes is maximized. We provide two alternative approaches, where one is based on fixed-sized time slots and the other is based on variable-sized time slots. For each approach, we derive the *provably* optimal routing algorithm. We also shed the light on the arising tradeoff between the complexity of network-layer route computation and the complexity of medium access control (MAC) layer scheduling of time slots, as well as the amenability to distributed implementation of our proposed algorithms. Our numerical results further illustrate the efficiency of the proposed approaches and their tradeoffs.

1. Introduction

Multihop wireless networks consist of a set of wireless devices that communicate with each other over multiple wireless hops, with participating nodes collaboratively relaying ongoing traffic. Wireless multihop relaying/routing is the foundation for the development and deployment of emerging technologies such as

(i) *client mesh networks*: a set of client devices (tablets, phones, and/or laptops) form a multihop network with *peer-to-peer* relaying;

(ii) *infrastructure wireless mesh networks*: wireless routers/access points are interconnected to provide an infrastructure/backbone for clients;

(iii) *millimeter-wave-based 5G networks*: future 5G networks are envisioned to depend (among others) on ultradense small-cell base stations and the use of millimeter- (mm-) wave spectrum for transmission [1]. The large bandwidth of mm-wave is also accompanied by a high path loss, which necessities the use of multihop relaying across the small-cell base stations [2]. Intelligent routing methods will also be needed for the underlying applications of 5G, e.g., the Internet of Things (IoT) [3].

The end-to-end *spectral efficiency* (in bps/Hz) of a communication route is defined as the rate at which data can be transmitted over the route per unit bandwidth. Therefore, it is an indication of how efficient the channel bandwidth is utilized. Since the bandwidth is a scarce resource in wireless systems, this paper focuses on finding communication routes with maximum spectral efficiencies. In particular, given a multihop wireless network consisting of set of wireless devices and interconnecting wireless links and a set of source-destination (*s*-*d*) pairs of nodes, this paper addresses the problem of finding a path for each *s*-*d* pair such that the minimum spectral efficiency of all paths is maximized.

1.1. Related Work. To the best of our knowledge, this is the first systematic, comprehensive study to address wireless spectrum-efficient routing in the case of multiple *s*-*d* pairs. Related work is presented in two categories:

(1) wireless spectrum-efficient routing,

(2) routing for multiple simultaneous *s-d* pairs.

In what follows we summarize the relevant previous work belonging to both groups.

Spectrum-efficient routing: the recent study in [4] has introduced the following spectrum-efficient routing problem. Given a multihop wireless network that employs time division multiple access (TDMA) and *one s-d* pair, it finds the route with *maximum spectral efficiency* under the constraint of equal bandwidth sharing. On the one hand, the authors of [4] noted that simple shortest path algorithms cannot be used to solve the problem because the resulting routing metric is not isotonic [5]. On the other hand, exhaustive search has an exponential computational complexity because it involves precomputing *all* paths joining a given node pair. Therefore, the study in [4] proposed two efficient, yet *suboptimal* spectrum-efficient routing heuristics. In [6], we have closed the algorithmic gap by introducing the *first* polynomial-time algorithm that solves the spectrum-efficient routing problem (originally introduced in [4]) to exact optimality. In particular, the algorithm presented in [6] has a worst-case computational complexity of $O(N^4)$, where N is the number of nodes. Moreover, we have introduced in [7, 8] an improved algorithm for the same problem that runs in $O(N^3)$-time. Furthermore, we have addressed in [9] the problem of joint routing and power allocation such that a desired spectral efficiency is achieved.

All the studies above focused on finding a route from a given source to a given destination and did not address the interaction between multiple *s-d* pairs; i.e., they did not address the multicommodity flow case. The study in [9] pointed very briefly to the extension to multiple *s-d* pairs for joint routing and power allocation. In contrast, this current paper takes a systematic and comprehensive approach to introduce the multicommodity spectrum-efficient routing problem, which is far from being fully explored.

Multicommodity flow: finding routes for multiple *s-d* pairs simultaneously, such that a network-wide objective is optimized and the link capacities are not exceeded, is known in the computing literature as the multicommodity flow problem. In particular, given a network with capacities on the links, and a set of *s-d* pairs of nodes with associated traffic demands, the problem is to route the demand of each *s-d* pair along *exactly one* route from s to d without violating/exceeding the link capacities. This problem is known as the integer multicommodity flow problem, or the *unsplittable* flow problem, and is known to be NP-hard. Readers can refer to [10, 11] for more information. If the limitation of routing each demand along exactly one route is relaxed (i.e., each demand can be split across arbitrarily many routes), the resulting *splittable* flow problem can be solved in polynomial-time using linear programming. Readers can refer to [12] for more information. The problem addressed in this paper falls in the category of unsplittable flow, which is in general NP-hard to solve. It is worth noting that the above-mentioned unsplittable/multicommodity flow literature is devised for generic network settings, mostly applicable to wired networks. In this paper, we do not borrow any general-purpose multicommodity flow algorithm from the computing literature. However, we devise new polynomial-time algorithms that harness the special structure of the spectrum-efficient routing problem and provide *provably* optimal solutions.

1.2. Contribution and Paper Outline. In light of the above, the contribution of this paper can be summarized as follows.

(i) To the best of our knowledge, for the first time, we address the problem of spectrum-efficient routing in the case of multiple *s-d* pairs. In particular, given a multihop wireless network and an arbitrary set of *s-d* pairs, we address the problem of finding a route for every *s-d* pair such that the minimum spectral efficiency across all routes is maximized.

(ii) For the problem above, we provide two alternative approaches, where one is based on fixed-sized time slots and the other is based on variable-sized time slots. For each approach, we derive the *provably* optimal routing algorithm. En route, our study sheds the light on the arising tradeoff between the complexity of network-layer route computation versus the complexity of medium access control (MAC) layer scheduling of time slots. Our numerical results further illustrate the efficiency of the proposed approaches, and their tradeoffs.

The remainder of this paper is organized as follows. Section 2 discusses the basics and preliminaries. Section 3 presents the problem formulation and provably optimal algorithm for the fixed time slots approach. The variable-time-slots-based approach, its problem formulation, and provably optimal algorithm are presented in Section 4. Section 5 discusses the tradeoff between the two approaches. Numerical examples and results are presented in Section 6. Section 7 concludes the paper.

2. Preliminaries

To avoid the computational intractability of joint optimal routing and medium access control (MAC) layer scheduling, and following [4, 6–9], it is assumed that a common channel is shared among all nodes using TDMA without spatial reuse, i.e., each node transmits in its own unique time slot. It is demonstrated in [4] that, even though a path is selected assuming no spatial reuse/interference, applying a scheduling technique (separately) that allows some spatial reuse to the selected path can further improve the spectral efficiency. In other words, our framework can still benefit from spatial reuse. It is worth noticing that the MAC layer of the IEEE 802.16 mesh protocol, for example, is based on TDMA (see, e.g., [13]).

A multihop wireless network is modeled as a graph $G = (V, E)$, where V represents the set of nodes (vertices) and E represents the set of links (edges). We let $l \in E$ signify a link in the network. We also let $N = |V|$ and $M = |E|$ denote the number of nodes and links, respectively.

Following [4, 6, 8], we consider the setting in which all transmit devices are constrained by the same symbol-wise average transmit power P and assume that all devices transmit with power P when transmitting. A possible justification for this assumption is that nodes in *infrastructure* wireless mesh networks are mostly immobile and connected with abundant power supplies. Therefore, for a link $l \in E$, the signal-to-noise ratio (SNR) is given by

$$SNR_l = \frac{PG_l}{N_0 B}, \quad (1)$$

where G_l is the path gain from the sender of link l to the receiver of link l, N_0 is the normalized one-sided power spectral density of the additive white Gaussian noise (at any receiver in the network), and B is the finite bandwidth of the wireless channel.

Now, assume K simultaneous s-d pairs are using the network. Each s-d pair $i = 1, 2, \ldots, K$ has a source node $s_i \in V$ and a destination node $d_i \in V$. We also let $\mathscr{L}_i$ denote the set of all routes from s_i to d_i. Moreover, we let $L_i \in \mathscr{L}_i$ signify a route from s_i to d_i.

Finally, the spectral efficiency of an arbitrary path L in the network is defined as the bandwidth-normalized end-to-end data rate [4]. In other words,

$$R(L) = \frac{C(L)}{B}, \quad (2)$$

where $R(L)$ is the spectral efficiency (in bps/Hz) of path L, $C(L)$ is the end-to-end achievable rate (in bps) for path L, and B is the channel bandwidth (in Hz).

3. Fixed-Sized Time Slots

The studies in [4, 6–9] focused on a *single* s-d route and assumed the bandwidth is shared equally among its links via TDMA. In other words, each link transmits in its own unique time slot, where the time slots are of fixed size. One way to extend this equal bandwidth sharing to the multicommodity case is to maintain the assumption that the time frame is divided equally among all links of the different s-d routes. In particular, if s-d pairs $i = 1, 2, \ldots, K$ are served by routes $L_1, L_2, \ldots, L_K$, respectively, then any link on any of the K routes will transmit for a fraction of $1/\sum_{i=1}^{K} |L_i|$ of the time, where $|L_i|$ is the number of hops/links in route L_i. In other words, the time frame will be divided into $\sum_{i=1}^{K} |L_i|$ *fixed-sized* slots, where each slot has a length $1/\sum_{i=1}^{K} |L_i|$ of the frame length.

3.1. Problem Formulation. In light of the above discussion, the TDMA end-to-end achievable data rate on path L_i can be expressed using the well-known Shannon capacity formula as

$$C(L_i) = \frac{B}{\sum_{i=1}^{K} |L_i|} \min_{l \in L_i} \log\left(1 + \frac{PG_l}{N_0 B}\right). \quad (3)$$

Note that the factor $1/\sum_{i=1}^{K} |L_i|$ comes from the sharing of the bandwidth equally among all links (of all routes), i.e., each link on any route is allocated a time fraction of $1/\sum_{i=1}^{K} |L_i|$ for transmission. Note also that the minimum function in (3) results from the fact that the end-to-end data rate of any path L_i is equal to the data rate achieved by its bottleneck link. Using (2), the spectral efficiency of path L_i can, thus, be expressed as follows:

$$R(L_i) = \frac{1}{\sum_{i=1}^{K} |L_i|} \min_{l \in L_i} \log\left(1 + \frac{PG_l}{N_0 B}\right). \quad (4)$$

Consequently, the minimum spectral efficiency, R_{min}, across all active routes $(L_1, L_2, \ldots, L_K)$ can be expressed as

$$R_{min}(L_1, L_2, \ldots, L_K) = \frac{1}{\sum_{i=1}^{K} |L_i|} \min_{l \in \{L_i : i=1,2,\ldots,K\}} \log\left(1 + \frac{PG_l}{N_0 B}\right) \quad (5)$$

Note that $\log(1 + PG_l/N_0 B)$ can be viewed as the width of any link l. Consequently, $\min_{l \in \{L_i : i=1,2,\ldots,K\}} \log(1 + PG_l/N_0 B)$ is the smallest link width used by the set of routes $L_1, L_2, \ldots, L_K$. In other words, the latter represents the width of the narrowest route among $L_1, L_2, \ldots, L_K$. To simplify our notation and algorithm development, we use the following substitution:

$$w(L_1, L_2, \ldots, L_K) = \min_{l \in \{L_i : i=1,2,\ldots,K\}} \log\left(1 + \frac{PG_l}{N_0 B}\right). \quad (6)$$

In other words, $w(L_1, L_2, \ldots, L_K)$ is the smallest link width used by the set of routes $L_1, L_2, \ldots, L_K$; i.e., it represents the width of the narrowest route among $L_1, L_2, \ldots, L_K$ using $\log(1 + PG_l/N_0 B)$ as link widths. Consequently, the minimum spectral efficiency, R_{min}, across all active routes $(L_1, L_2, \ldots, L_K)$ can be rewritten as

$$R_{min}(L_1, L_2, \ldots, L_K) = \frac{w(L_1, L_2, \ldots, L_K)}{\sum_{i=1}^{K} |L_i|}. \quad (7)$$

Therefore, the problem of jointly finding routes for s-d pairs $1, 2, \ldots, K$ such that the minimum spectral efficiency across all routes is maximized can be cast as the following optimization problem:

$$\max_{L_i \in \mathscr{L}_i \, \forall 1 \le i \le K} \frac{w(L_1, L_2, \ldots, L_K)}{\sum_{i=1}^{K} |L_i|}. \quad (8)$$

It is worth noting that problem (8) cannot be solved using standard shortest path methods as the resulting routing metric is not isotonic [5]. In particular, even with one s-d pair, the routing metric of (8) is not isotonic. See, e.g., [4, 6]. In what follows, we develop a polynomial-time algorithm that provides provably optimal solutions to (8).

3.2. Algorithm. The main idea of the proposed algorithm is an extension of the single s-d pair case [6]. In fact, even for multiple paths $L_1, L_2, \ldots, L_K$, the value of $w(L_1, L_2, \ldots, L_K)$ takes one of finite possible values. It is readily seen from (6) that $w(L_1, L_2, \ldots, L_K) \in W$, where W is the set of all link widths in the network; i.e.,

$$W = \left\{\log\left(1 + \frac{PG_l}{N_0 B}\right) : l \in E\right\}. \quad (9)$$

Recall that E is the set of links in the network. Since $|W| = M = O(N^2)$, $w(L_1, L_2, \ldots, L_K)$ can take at most $M = O(N^2)$ values.

The main result follows.

Theorem 1. *Let the set of routes $(L_1^*, L_2^*, \ldots, L_K^*)$ denote the optimal solution to the original multicommodity spectrum-efficient routing problem (8). Let also the set of routes $(L_1^a, L_2^a, \ldots, L_K^a)$ denote the optimal solution to the following modified problem:*

$$\max_{\substack{L_i \in \mathscr{L}_i \, \forall 1 \le i \le K \\ w(L_1, L_2, \ldots, L_K) \ge a}} \frac{a}{\sum_{i=1}^{K} |L_i|}. \tag{10}$$

Then

$$R_{min}(L_1^*, L_2^*, \ldots, L_K^*) = \max_{a \in W} R_{min}(L_1^a, L_2^a, \ldots, L_K^a). \tag{11}$$

Proof. First, let the set of routes $(\widehat{L}_1^a, \widehat{L}_2^a, \ldots, \widehat{L}_K^a)$ be an optimal solution to the following subproblem:

$$\max_{\substack{L_i \in \mathscr{L}_i \, \forall 1 \le i \le K \\ w(L_1, L_2, \ldots, L_K) = a}} \frac{w(L_1, L_2, \ldots, L_K)}{\sum_{i=1}^{K} |L_i|}. \tag{12}$$

Note that (12) is the same as (8) with the additional constraint that $w(L_1, L_2, \ldots, L_K) = a$. By the divide-and-conquer principle [14], and since the union of the sets $\{L_i \in \mathscr{L}_i : 1 \le i \le K, w(L_1, L_2, \ldots, L_K) = a\}$, over all possible values of $a \in W$, covers the route set $\{L_i \in \mathscr{L}_i : 1 \le i \le K\}$, the following is true:

$$R_{min}(L_1^*, L_2^*, \ldots, L_K^*) = \max_{a \in W} R_{min}\left(\widehat{L}_1^a, \widehat{L}_2^a, \ldots, \widehat{L}_K^a\right). \tag{13}$$

Note that $R_{min}(L_1^*, L_2^*, \ldots, L_K^*)$ and $R_{min}(\widehat{L}_1^a, \widehat{L}_2^a, \ldots, \widehat{L}_K^a)$ represent the optimal objective function values of (8) and (12), respectively.

Moreover, by substituting the equality constraint $w(L_1, L_2, \ldots, L_K) = a$ in its objective function, (12) is equivalent to

$$\max_{\substack{L_i \in \mathscr{L}_i \, \forall 1 \le i \le K \\ w(L_1, L_2, \ldots, L_K) = a}} \frac{a}{\sum_{i=1}^{K} |L_i|}. \tag{14}$$

Now, it is readily seen that (10) is a relaxation of (14). Therefore, if $(L_1^a, L_2^a, \ldots, L_K^a)$ and $(\widehat{L}_1^a, \widehat{L}_2^a, \ldots, \widehat{L}_K^a)$ are the optimal solutions to (10) and (14), respectively, then $a/\sum_{i=1}^{K} |L_i^a| \ge a/\sum_{i=1}^{K} |\widehat{L}_i^a|$. The latter implies that

$$\sum_{i=1}^{K} \left|L_i^a\right| \le \sum_{i=1}^{K} \left|\widehat{L}_i^a\right|. \tag{15}$$

Now, the following is true:

$$\begin{aligned} R_{min}(L_1^*, \ldots, L_K^*) &= \max_{a \in W} \frac{w\left(\widehat{L}_1^a, \widehat{L}_2^a, \ldots, \widehat{L}_K^a\right)}{\sum_{i=1}^{K} \left|\widehat{L}_i^a\right|} \\ &= \max_{a \in W} \frac{a}{\sum_{i=1}^{K} \left|\widehat{L}_i^a\right|} \\ &\le \max_{a \in W} \frac{w\left(L_1^a, L_2^a, \ldots, L_K^a\right)}{\sum_{i=1}^{K} \left|L_i^a\right|} \end{aligned} \tag{16}$$

Note that the first equality comes directly from (13). The second equality comes from the fact that the route set $(\widehat{L}_1^a, \widehat{L}_2^a, \ldots, \widehat{L}_K^a)$ is feasible for (12). The inequality comes from (15) and from the fact that the route set $(L_1^a, L_2^a, \ldots, L_K^a)$ is feasible for (10). Consequently, (16) implies that

$$R_{min}(L_1^*, L_2^*, \ldots, L_K^*) \le \max_{a \in W} R_{min}(L_1^a, L_2^a, \ldots, L_K^a). \tag{17}$$

Since, among all route sets $\{L_i \in \mathscr{L}_i : 1 \le i \le K\}$, $(L_1^*, L_2^*, \ldots, L_K^*)$ is the route set which maximizes the minimum spectral efficiency, (17) must hold with strict equality. This completes the proof. □

In light of Theorem 1, the multicommodity spectrum-efficient routing problem (8) can be solved using the following procedure:

(i) `For every` $a \in W$, `find the route set` $(L_1^a, L_2^a, \ldots, L_K^a)$ `by solving (10).`

(ii) `Return the route set` $(L_1^*, L_2^*, \ldots, L_K^*) = \operatorname{argmax}_{a \in W} R_{min}(L_1^a, L_2^a, \ldots, L_K^a)$.

Recall that, in the above procedure, W is the set of link widths given by (9), and R_{min} is given by (7). The only remaining issue to show is how to solve (10). Note that, for a given $a \in W$, maximizing $a/\sum_{i=1}^{K} |L_i|$ is equivalent to minimizing $\sum_{i=1}^{K} |L_i|$. Moreover, the latter is minimized if every *s-d* pair i minimizes $|L_i|$. Consequently, problem (10) is equivalent to finding the minimum-hop path for every *s-d* pair $1 \le i \le K$, such that the minimum link width across all paths is not less than a. Therefore, for a given value of a, (10) can be solved as follows:

(i) Remove all links $l \in E$ for which $\log(1 + \mathrm{PG}_l/N_0 B) < a$. In the remaining graph, obtain the minimum-hop path for every *s-d* pair $1 \le i \le K$.

In light of the above discussion, problem (8) can be solved by the following algorithm.

`Algorithm Equal-Time-Slots`

(1) `Let` $W = \{\log(1 + PG_l/N_0 B) : l \in E\}$. `For every` $a \in W$, `do:`

  (a) `For every` $s - d$ `pair` $i = 1, 2, \ldots, K$, `do:`

    (i) `Remove all links` $l \in E$ `for which` $\log(1 + PG_l/N_0 B) < a$.

    (ii) `In the remaining graph, find` L_i^a, `the minimum-hop path from source` s_i `to destination` d_i.

  (b) `Let` $R_{min}(L_1^a, L_2^a, \ldots, L_K^a) = w(L_1^a, L_2^a, \ldots, L_K^a)/\sum_{i=1}^{K} |L_i^a|$.

(2) `Return the path set with largest` $R_{min}(L_1^a, L_2^a, \ldots, L_K^a)$.

3.3. Observations. The following observations are in order regarding algorithm *Equal-Time-Slots*.

(i) Step (1a) of algorithm *Equal-Time-Slots* can be implemented by each $s-d$ pair independently, and without any coordination with the other $s-d$ pairs. In particular, $s-d$ pair i (or more precisely source node s_i) obtains its path L_i^a independently.

(ii) Step (1b), however, requires knowledge about all $s-d$ pairs. Therefore, it can be implemented by a centralized entity which knows the hop-count $|L_i^a|$ of every path L_i^a. Alternatively, it requires that all $s-d$ pairs (or source nodes) exchange their information about $|L_i^a|$ with all other nodes using flooding, or any other means of all-to-all communication.

(iii) At the MAC layer, the resulting set of paths will require dividing the time frame into $\sum_{i=1}^{K}|L_i^*|$ *equal-sized* time slots. This simplicity of MAC layer scheduling comes at the expense of the necessity of coordination between $s-d$ pairs during network-layer path computation.

(iv) The computational complexity of algorithm *Equal-Time-Slots* is dominated by the complexity of invoking a shortest path procedure $|W| = M$ times, as in step (1a). Note that step (1a) can be implemented by the different s-d pairs in parallel. Since the number of links M is of $O(N^2)$, where N is the number of nodes, and if the Dijkstra shortest path algorithm is used in every iteration, the overall complexity of algorithm *Equal-Time-Slots* is $O(N^2 \cdot N^2) = O(N^4)$.

4. Variable-Sized Time Slots

An alternative approach to accommodating multiple s-d pairs under the condition of equal bandwidth sharing is to divide the time frame equally among *s-d pairs* (as opposed to dividing the time equally among the *links*). In other words, every s-d pair/path will transmit for a fraction of $1/K$ of the time (assuming K s-d pairs). Consequently, if s-d pair i uses path L_i, then every link along this path will transmit for a fraction of $1/K|L_i|$ of the time. Since different s-d pairs may use paths with different hop-counts, their respective links may use time slots of different sizes.

In this case, the end-to-end spectral efficiency of route L_i serving s-d pair i can be expressed as

$$R(L_i) = \frac{1}{K|L_i|}\min_{l\in L_i}\log\left(1+\frac{PG_l}{N_0 B}\right). \tag{18}$$

Note that the factor $1/K|L_i|$ comes from the fact that every link $l \in L_i$ transmits for a fraction of $1/K|L_i|$ of the time. Note also that the minimum function in (18) results from the fact that the end-to-end data rate of any path L_i is equal to the data rate achieved by its bottleneck link. It is worth noting that the spectral efficiency for s-d pair i depends on the hop-count $|L_i|$ of its own path L_i only. This is in contrast to the case of equal time slots, where the spectral efficiency for any s-d pair i depends on the hop-counts of all s-d paths $L_1, L_2, \ldots, L_K$.

The problem of maximizing the minimum spectral efficiency across all s-d pairs can, thus, be expressed as

$$\max_{L_i\in\mathscr{L}_i\,\forall 1\le i\le K}\ \min_{1\le i\le K} R(L_i), \tag{19}$$

where $R(L_i)$ is given by (18). It is not hard to see that the minimum spectrum efficiency will be maximized if every s-d pair i maximizes its individual spectral efficiency $R(L_i)$. In other words, every s-d pair i solves the following optimization problem:

$$\max_{L_i\in\mathscr{L}_i}\frac{1}{K|L_i|}\min_{l\in L_i}\log\left(1+\frac{PG_l}{N_0 B}\right). \tag{20}$$

Moreover, since, for any number of s-d pairs, K is a constant, solving (20) is equivalent to solving the single s-d pair problem. The best known algorithm for solving (20) has been introduced in [8], and can be summarized as follows.

`Algorithm Variable-Time-Slots`
`For every` s-d `pair` $i = 1, 2, \ldots, K$, `do:`

(1) `For` $h = 1, 2, \ldots, N-1$, `do:`

  (a) `Find` L_i^h, `the widest path with at most` h `hops connecting` s_i `to` d_i, `using` $\log(1 + PG_l/N_0B)$ `as link labels.`

  (b) `Let` $R(L_i^h) = (1/K|L_i^h|)\min_{l\in L_i^h}\log(1 + PG_l/N_0B)$.

(2) `Return the path with largest` $R(L_i^h)$.

The following observations are in order regarding algorithm *Variable-Time-Slots*.

(i) The algorithm can be implemented by each s-d pair in a completely independent manner. In other words, s-d pairs are completely isolated and there is no need for coordination and/or a centralized component.

(ii) The number of s-d pairs K does not affect the computation of the optimal paths. In particular, for any number of s-d pairs, K is a constant, and thus

$$\begin{aligned}\operatorname*{argmax}_{L_i\in\mathscr{L}_i}&\frac{1}{K|L_i|}\min_{l\in L_i}\log\left(1+\frac{PG_l}{N_0 B}\right)\\ &=\operatorname*{argmax}_{L_i\in\mathscr{L}_i}\frac{1}{|L_i|}\min_{l\in L_i}\log\left(1+\frac{PG_l}{N_0 B}\right).\end{aligned} \tag{21}$$

In other words, every s-d pair computes its optimal path regardless of how many other s-d pairs exist, and the optimal paths do not change with the change of the number of s-d pairs.

(iii) The number of s-d pairs K, however, is needed for MAC layer scheduling. In particular, every s-d pair transmits for $1/K$ of the time, and every link $l \in L_i$ used by s-d pair i transmits for a fraction of $1/K|L_i|$ of the time.

(iv) Since the paths used by different s-d pairs may have different hop-counts, the resulting time slots may be of different durations.

(v) The algorithm can be implemented by the different s-d pairs in parallel. Note also that shortest path algorithms can be modified to compute the widest path. See, e.g., [15]. Moreover, it is worth noticing that the Bellman-Ford shortest path algorithm, in its h^{th} iteration, computes the shortest (or widest) path with at most h hops. Consequently, algorithm *Variable-Time-Slots* can be implemented by invoking the Bellman-Ford procedure only *once*. The overall complexity of algorithm *Variable-Time-Slots* is, thus, $O(N^3)$.

Note that *dynamically* partitioning the TDMA frame into a set of *variable-length* transmission slots is possible. See, e.g., [9, 16]. As an example of the resulting MAC strategy, let the number of s-d pairs be 3. Assume also that applying algorithm *Variable-Time-Slots* results in paths L_1, L_2, and L_3 for the 3 s-d pairs, where $|L_1| = 3$, $|L_2| = 2$, and $|L_3| = 4$, respectively. The time frame will, thus, be divided into $3 + 2 + 4 = 9$ slots. The normalized slot size is $1/(3 \times 3) = 1/9$ for the first 3 slots (used by path L_1), $1/(3 \times 2) = 1/6$ for the following 2 slots (used by path L_2), and $1/(3 \times 4) = 1/12$ for the last 4 slots (used by path L_3). In other words, the normalized sizes of the 9 slots are 1/9, 1/9, 1/9, 1/6, 1/6, 1/12, 1/12, 1/12, and 1/12, respectively.

In contrast, however, assume that algorithm *Equal-Time-Slots* results in paths L_1, L_2, and L_3 for the 3 s-d pairs, where $|L_1| = 3$, $|L_2| = 2$, and $|L_3| = 4$, respectively. In this case, the time frame will be divided into $3 + 2 + 4 = 9$ slots, where the normalized size of each slot is precisely 1/9.

5. Tradeoffs

In this section we summarize the main differences and tradeoffs between algorithms *Equal-Time-Slots* and *Variable-Time-Slots*.

(i) **Distributed implementation**: algorithm *Equal-Time-Slots* has a component that requires centralized knowledge of the paths computed by *all* s-d pairs (in every iteration of the algorithm). Alternatively, coordination and exchange of information is required among s-d pairs. Algorithm *Variable-Time-Slots*, however, can be implemented in a fully distributed fashion among s-d pairs. No coordination is required among s-d pairs in their path computation.

(ii) **Synchronization**: as described above, algorithm *Equal-Time-Slots* is based on the centralized (or coordinated) knowledge of the paths L_i^a for all s-d pairs $i = 1, 2, \ldots, K$. This implicitly implies that all existing s-d pairs must make their path computation decisions (i.e., invoke their algorithms) in a synchronized way; any s-d pair cannot obtain its optimal path without the results of the other s-d pairs. Algorithm *Variable-Time-Slots*, however, does not require this synchronization. Any s-d pair can compute its optimal path regardless of the other s-d pairs. In fact, optimal path computation does not even require the knowledge of the number of existing s-d pairs K. The optimal path in case of only *one* s-d pair would remain optimal in the presence of *any* number of s-d pairs K. Knowing K is necessary in the transmission (MAC) scheduling phase only.

(iii) **MAC scheduling**: algorithm *Equal-Time-Slots* results in TDMA transmission frames with equal time slots, while algorithm *Variable-Time-Slots* results in TDMA transmission frames with variable time slots.

(iv) **Complexity**: the complexity of *Equal-Time-Slots* is $O(N^4)$, while the complexity of *Variable-Time-Slots* is $O(N^3)$.

It is worth noting that both algorithms (*Equal-Time-Slots* and *Variable-Time-Slots*) require the value of the link SNRs (PG_l/N_0B) to be known at the source nodes computing their respective paths. In practice, the link SNR can be directly measured by received signal strength indicators available on most devices [4], and fed back to the transmitters. Nodes can then exchange their knowledge about the values of PG_l/N_0B for their outgoing links using a *distributed* link-state protocol. Please refer to [9] for further elaboration.

In the following section we provide a numerical study on the performance of both proposed approaches and their tradeoffs.

6. Numerical Results

We consider multihop wireless networks, in which the nodes are located at random positions in a 100×100 two-dimensional area. Without loss of generality, it is assumed that any two nodes can directly communicate; i.e., the network is fully connected. Note that, from an information theoretic point of view, two nodes can always communicate at a sufficiently low rate [4, 17]. The path gain G_l of each link l is assumed to be given by

$$G_l = c \cdot A_l \cdot \left(\max\left\{d_l, d_0\right\}\right)^{-4}, \tag{22}$$

where d_l is the length of link l, d_0 is the reference distance for the far-field, A_l is a log-normally distributed random variable (with 0-dB mean and 8-dB log-variance) that reflects shadowing, and c is a constant. Without loss of generality, we set $d_0 = 0.1$ and $c = 0.01$. We test our proposed algorithms on *random* and *independent* network realizations, where in each realization the horizontal and vertical coordinates of each node are chosen randomly (and independently) according to a uniform distribution between 0 and 100, and the path gains are generated randomly (and independently) according to (22). Among the randomly generated nodes, a set of s-d pairs is chosen at random. Furthermore, for each tested scenario, we average our results over 10^4 random network realizations. In other words, *every point in each of the following result figures is averaged over* 10^4 *random network realizations*.

The simulation parameters are summarized in Table 1 (note that $E(\cdot)$ and $V(\cdot)$ denote the expected value and variance of a random variable, respectively).

6.1. Effect of the Network Size. First, we vary the number of nodes in the network (N) from 5 to 30. For every value of N,

Table 1: Simulation parameters.

Parameter	Value
Path gain (G_l)	Equation (22)
Path gain constant (c)	0.01
Far-field reference distance (d_0)	0.1
Shadowing (A_l)	log-normal distribution
$E(10\log A_l)$	0 dB
$V(10\log A_l)$	8 dB
Node positions	random in 100×100 area
Number of nodes (N)	5 to 30
Number of s-d pairs (K)	5 to 20
Network SNR (P/N_0B)	-20 dB to 80 dB
Simulation software	MATLAB R2017a

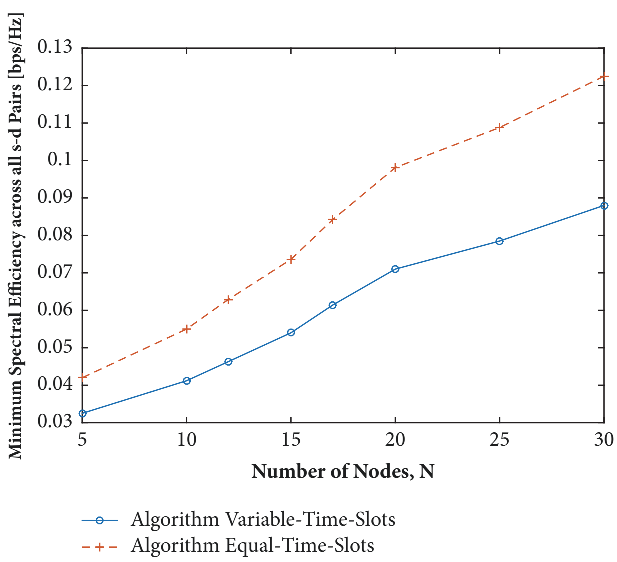

Figure 1: Minimum spectral efficiency among all s-d routes obtained using algorithms *Equal-Time-Slots* and *Variable-Time-Slots*, respectively, versus the number of network nodes.

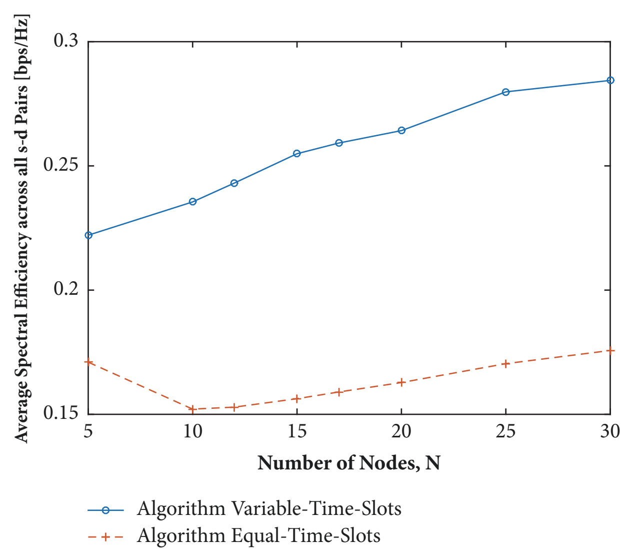

Figure 2: Average spectral efficiency among all s-d routes obtained using algorithms *Equal-Time-Slots* and *Variable-Time-Slots*, respectively, versus the number of network nodes.

we let the number of s-d pairs (K) be 5, and we set the network SNR (P/N_0B) to 80 dB. Figure 1 depicts the *minimum* spectral efficiency among all s-d routes obtained using algorithms *Equal-Time-Slots* and *Variable-Time-Slots*, respectively. It is clearly seen that algorithm *Equal-Time-Slots* results in higher worst-case spectral efficiencies. In particular, the minimum source-destination (s-d) spectral efficiency resulting from algorithm *Equal-Time-Slots* is from 29.23% to 39.2% higher than that of algorithm *Variable-Time-Slots*. Averaged over all experiments for different values of N, the minimum s-d spectral efficiency resulting from algorithm *Equal-Time-Slots* is 36% higher than that of algorithm *Variable-Time-Slots*.

Moreover, Figure 2 depicts the *average* spectral efficiency across all s-d routes obtained using algorithms *Equal-Time-Slots* and *Variable-Time-Slots*, respectively. It is straightforward to see that algorithm *Variable-Time-Slots* results in higher average spectral efficiencies. In particular, the average s-d spectral efficiency resulting from algorithm *Variable-Time-Slots* is from 29.88% to 64.2% higher than that of algorithm *Equal-Time-Slots*. Averaged over all experiments for different values of N, the average s-d spectral efficiency resulting from algorithm *Variable-Time-Slots* is 57.27% higher than that of algorithm *Equal-Time-Slots*. In short, although algorithm *Equal-Time-Slots* has a better worst-case performance (as seen in Figure 1), algorithm *Variable-Time-Slots* has a significantly better average performance (as seen in Figure 2).

Finally, we provide a comparison between the running times of algorithms *Equal-Time-Slots* and *Variable-Time-Slots* per s-d pair. For fairness of comparison, we compare the overall running time of algorithm *Variable-Time-Slots* with the running time of the distributed component of algorithm *Equal-Time-Slots* (i.e., Step (1a), which can be implemented by each s-d pair in isolation). In other words, the running time of the centralized component of algorithm *Equal-Time-Slots* is excluded from the comparison. The running times of both algorithms are depicted in Figure 3. In fact, the (per

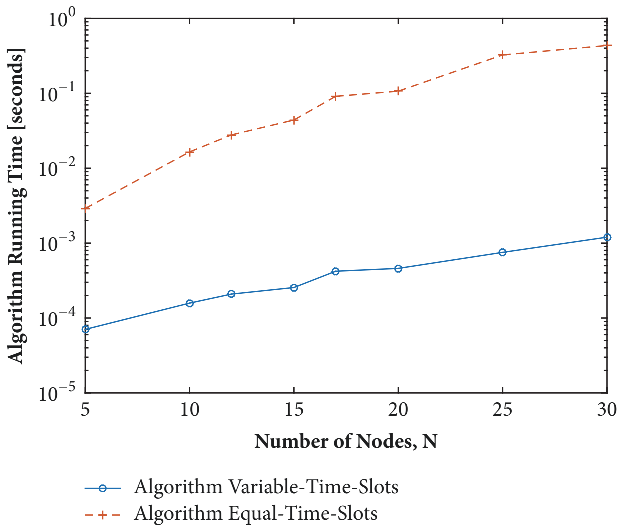

Figure 3: Running times of algorithms *Equal-Time-Slots* and *Variable-Time-Slots*.

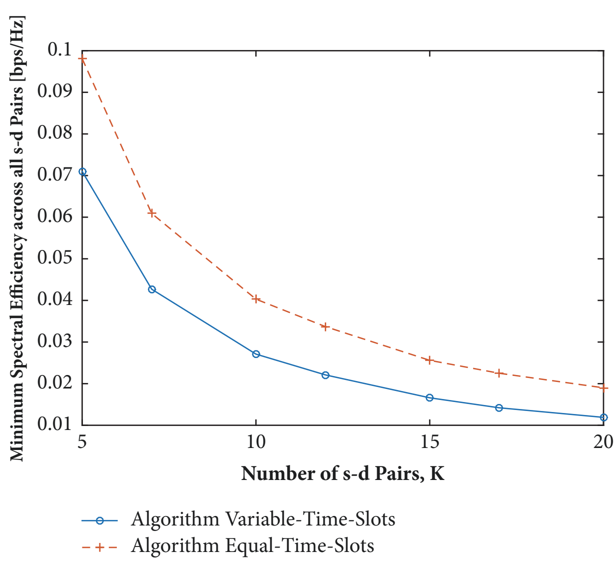

Figure 4: Minimum spectral efficiency among all *s-d* routes obtained using algorithms *Equal-Time-Slots* and *Variable-Time-Slots*, respectively, versus the number of *s-d* pairs.

s-d pair) average running time of algorithm *Variable-Time-Slots* is 0.44 milliseconds, while that of algorithm *Equal-Time-Slots* is 0.13 seconds. In other words, although the centralized component of algorithm *Equal-Time-Slots* was not considered in this comparison, the running time of algorithm *Variable-Time-Slots* is on average 99.23% lower than that of algorithm *Equal-Time-Slots*.

6.2. Effect of Number of s-d Pairs. Now, we vary the number of *s-d* pairs (K) from 5 to 20, while the number of nodes is fixed at $N = 20$ and the network SNR is fixed at 80 dB. The results for minimum and average spectral efficiencies across all *s-d* routes are depicted in Figures 4 and 5, respectively. Again, it can be easily seen that algorithm *Equal-Time-Slots* has a better worst-case performance (as seen in Figure 4),

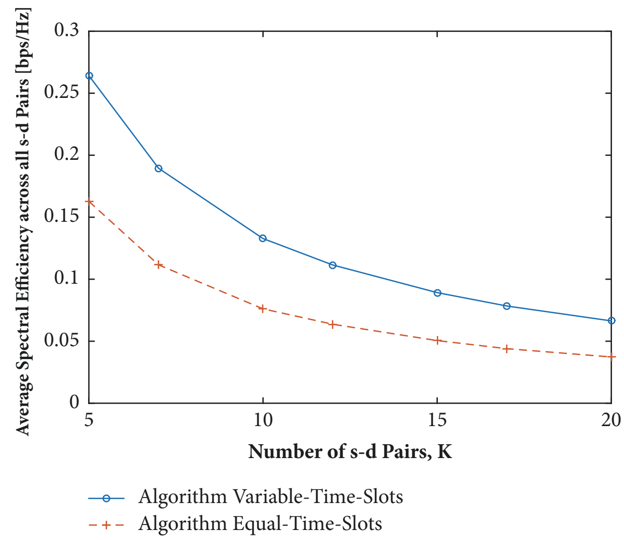

Figure 5: Average spectral efficiency among all *s-d* routes obtained using algorithms *Equal-Time-Slots* and *Variable-Time-Slots*, respectively, versus the number of *s-d* pairs.

while algorithm *Variable-Time-Slots* has a significantly better average performance (as seen in Figure 5). In particular, the minimum spectral efficiency across all *s-d* routes obtained by algorithm *Equal-Time-Slots* is from 38.17% to 59.66% higher than that obtained by algorithm *Variable-Time-Slots*. Averaged over all experiments, the minimum spectral efficiency resulting from algorithm *Equal-Time-Slots* is 50.62% higher than that resulting from algorithm *Variable-Time-Slots*. On the other hand, the average spectral efficiency across all *s-d* routes resulting from algorithm *Variable-Time-Slots* is from 62.29% to 78.19% higher than that resulting from algorithm *Variable-Time-Slots*. Averaged over all experiments, algorithm *Variable-Time-Slots* results in 73.50% higher average spectral efficiencies than algorithm *Equal-Time-Slots*.

6.3. Effect of the Network SNR. Now, we vary the network SNR from -20 dB to 80 dB, while the number of nodes is fixed at $N = 20$ and the number of *s-d* pairs is fixed at $K = 5$. The results for minimum and average spectral efficiencies across all *s-d* routes are depicted in Figures 6 and 7, respectively. In consistency with all other results, algorithm *Equal-Time-Slots* consistently shows a better worst-case performance (as seen in Figure 6), while algorithm *Variable-Time-Slots* shows a consistently and significantly better average performance (as seen in Figure 7). In particular, the improvement in *worst-case* spectral efficiencies due to algorithm *Equal-Time-Slots* is between 37.96% and 38.24% (with an average improvement of 37.93% across all experiments). However, the improvement in *average* spectral efficiencies due to algorithm *Variable-Time-Slots* is between 65.18% and 136.37% (with an average improvement of 129.20% across all experiments).

6.4. Comparison against Benchmarks. Finally, we assess the performance of our proposed algorithms in comparison to existing techniques. To this end, we use the following two benchmarks.

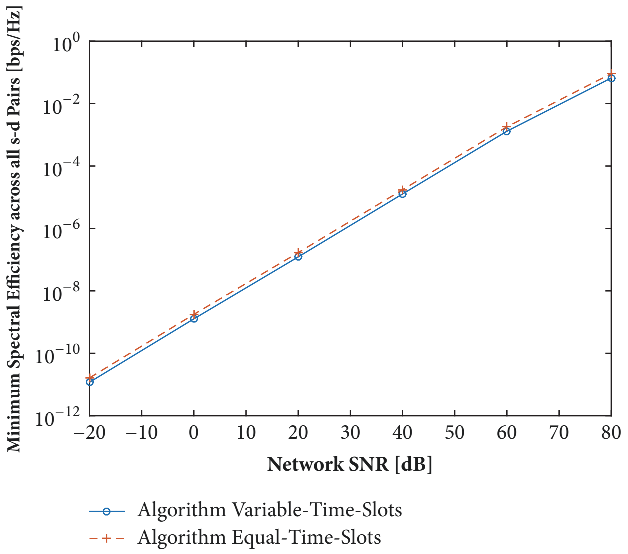

Figure 6: Minimum spectral efficiency among all *s-d* routes obtained using algorithms *Equal-Time-Slots* and *Variable-Time-Slots*, respectively, versus the network SNR.

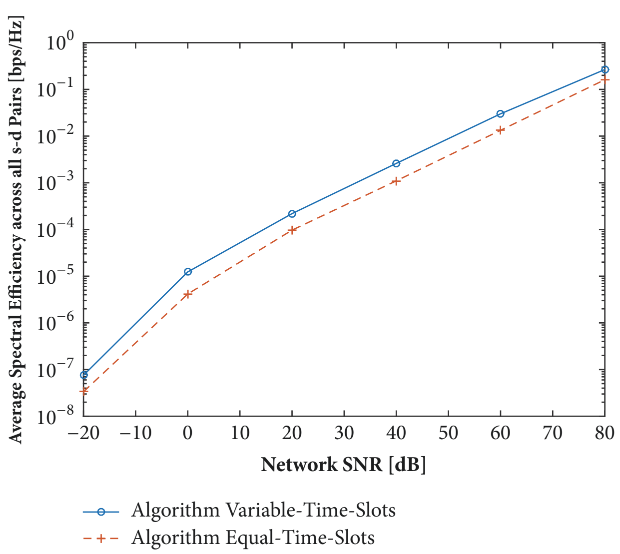

Figure 7: Average spectral efficiency among all *s-d* routes obtained using algorithms *Equal-Time-Slots* and *Variable-Time-Slots*, respectively, versus the network SNR.

(i) We compare our algorithms against the spectral efficiency resulting from routing every *s-d* pair along the direct link from *s* to *d*. Since all resulting routes are one hop long, dividing the time frame equally between *s-d* pairs or between transmission links is equivalent. Therefore, direct link routing is compared against both proposed algorithms *Equal-Time-Slots* and *Variable-Time-Slots*.

(ii) We also compare our algorithms against the spectral efficiency resulting from routing every *s-d* pair *independently* using the distributed spectrum-efficient routing (DSER) algorithm introduced in [4]. DSER [4] operates by simply finding a shortest path from the source to the destination using $(1 + 2^{\gamma}/SNR_l)$ as the link metric, where γ is the path-loss exponent. Here, $\gamma = 4$. It is assumed that the time frame is divided equally among *s-d* pairs, resulting in variable-length time slots for individual link transmissions. Therefore, DSER is compared against our proposed algorithm *Variable-Time-Slots*.

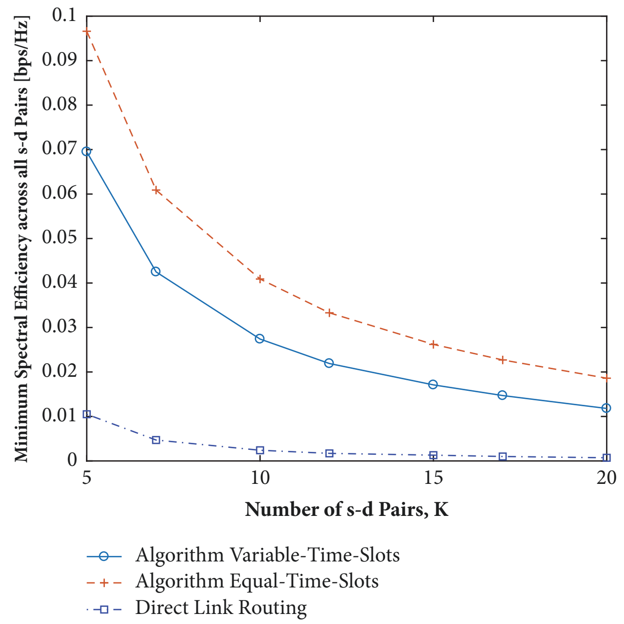

Figure 8: Minimum spectral efficiency among all *s-d* routes obtained using algorithms *Equal-Time-Slots* and *Variable-Time-Slots* compared against direct link routing.

To compare them against direct link routing, we vary the number of *s-d* pairs K from 5 to 20, while the number of nodes is fixed at $N = 20$ and the network SNR is fixed at 80 dB. The results for minimum and average spectral efficiencies across all *s-d* routes are depicted in Figures 8 and 9, respectively. Following the same trend, algorithm *Equal-Time-Slots* shows a better worst-case performance, while algorithm *Variable-Time-Slots* shows a better average performance. In particular, the worst-case spectral efficiency resulting from algorithm *Equal-Time-Slots* is 820% to 2500% higher than that resulting from direct routing (with an average improvement of 1724% across all experiments). Moreover, the average spectral efficiency resulting from algorithm *Variable-Time-Slots* is about 30% higher than that resulting from direct routing in all experiments.

To compare against DSER from [4], we vary the number of nodes in the network (N) from 5 to 30, while the number of *s-d* pairs (K) is set to 5 and the network SNR is set to 80 dB. The results for minimum and average spectral efficiencies across all *s-d* routes are depicted in Figures 10 and 11, respectively. The superior performance of our proposed algorithm *Variable-Time-Slots* is clearly seen. In particular, our algorithm results in 93% to 743% higher worst-case spectral efficiencies as compared to direct link routing (with an average improvement of 446% across all experiments) and results in 16% to 68% higher worst-case spectral efficiencies

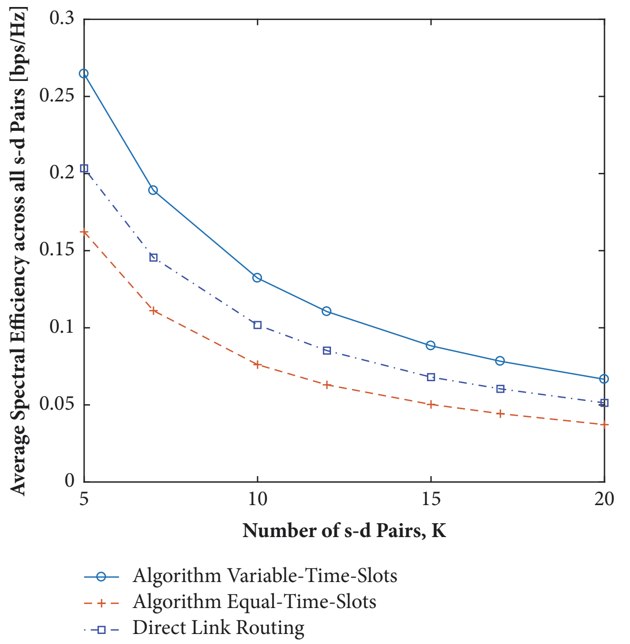

FIGURE 9: Average spectral efficiency among all *s-d* routes obtained using algorithms *Equal-Time-Slots* and *Variable-Time-Slots* compared against direct link routing.

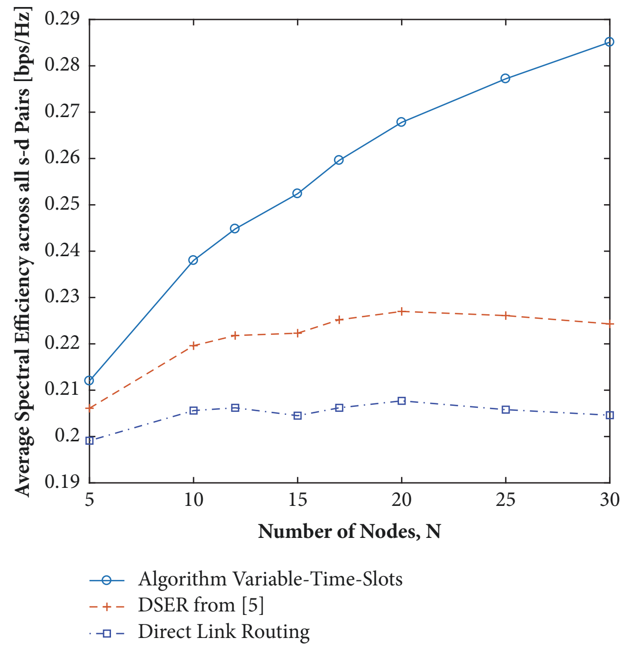

FIGURE 11: Average spectral efficiency among all *s-d* routes obtained using algorithm *Variable-Time-Slots* compared against DSER from [4] and direct link routing.

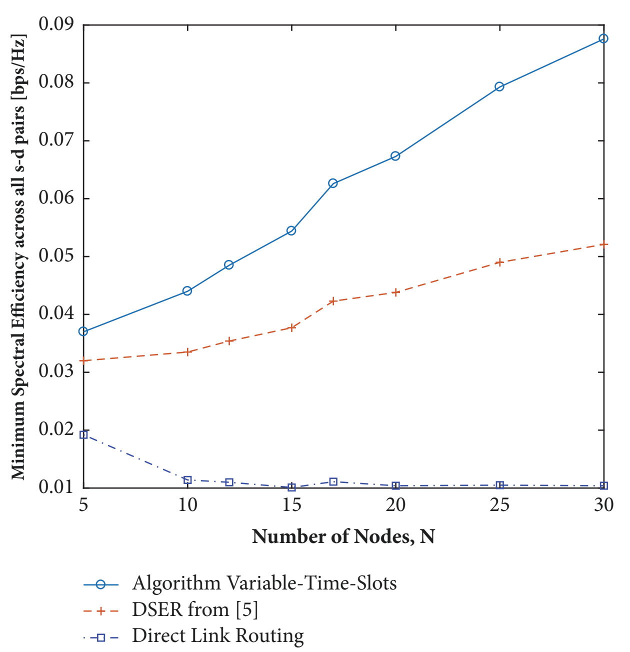

FIGURE 10: Minimum spectral efficiency among all *s-d* routes obtained using algorithm *Variable-Time-Slots* compared against DSER from [4] and direct link routing.

as compared to DSER (with an average improvement of 45% across all experiments). Moreover, our algorithm results in 6.5% to 39% higher average spectral efficiencies as compared to direct link routing (with an average improvement of 24% across all experiments) and results in 2.9% to 27% higher average spectral efficiencies as compared to DSER (with an average improvement of 15% across all experiments).

In summary, all our experiments (of varying the network size, number of *s-d* pairs, and network SNR) indicate a similar trend of a better worst-case performance for algorithm *Equal-Time-Slots* versus a better average/typical performance for algorithm *Variable-Time-Slots*. It is worth noting, however, that the improvement in average results due to algorithm *Variable-Time-Slots* is always more significant. Moreover, algorithm *Variable-Time-Slots* enjoys a significantly (more than 99%) lower running time. Moreover, our proposed algorithm *Variable-Time-Slots* has shown a significantly superior worst-case and average performance as compared to DSER from [4], and as compared to direct link routing.

7. Conclusion

To the best of our knowledge, previous work on finding the path with maximum end-to-end spectrum efficiency was restricted to a single *s-d* pair. This paper proposed two alternative approaches for the spectrum-efficient routing problem in the multicommodity flow regime, i.e., in the case of multiple active *s-d* pairs. The routing objective was to maximize the minimum spectrum efficiency achieved across all active *s-d* pairs. The first approach was based on dividing the time frame into equal-sized slots, while the second approach allows dividing the time frame into variable-sized slots. For each approach, we derived the *provably* optimal routing algorithm. We also shed the light on the arising tradeoff between the resulting routing algorithms. In summary, the routing algorithm induced by equal time slots enjoys a better worst-case performance (i.e., higher worst-case spectrum efficiencies), at the price of a higher computational complexity and the existence of a centralized component requiring coordination and synchronization among all *s*-*d* pairs in the route computation phase. However, the routing algorithm induced by variable time slots has the advantages of (1) a significantly lower computational complexity (more than 99% reduction in running time), (2) a significantly better average/typical performance (i.e., higher average achieved spectral efficiencies), (3) a significantly better worst-case and average spectral efficiency performance as compared to existing methods, and (4) being entirely distributed with no need for coordination or synchronization among *s*-*d* pairs. In fact the routing algorithm induced by variable time slots does not even require the knowledge of the number of active *s*-*d* pairs; the optimal path in case of the existence of a single *s*-*d* pair remains optimal in case of any number of *s*-*d* pairs. Knowledge of the number *s*-*d* pairs is required only in the phase of MAC layer scheduling of time slots. This concludes that algorithm *Variable-Time-Slots* might be preferred for practical implementation.

Data Availability

The details of how the numerical experiment scenarios were generated are clearly explained in Section 6 of the manuscript and can easily be regenerated.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

This research was supported in part by the Distributed and Networked Systems Research Group, University of Sharjah, Operating Grant no. 150410, and in part by a University of Sharjah Targeted Project no. 1602040336-P.

References

[1] J. G. Andrews, S. Buzzi, and W. Choi, "What will 5G be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[2] W. Feng, Y. Li, D. Jin, L. Su, and S. Chen, "Millimetre-Wave Backhaul for 5G Networks: Challenges and Solutions," *Sensors*, vol. 16, no. 6, p. 892, 2016.

[3] D. Airehrour, J. Gutierrez, and S. K. Ray, "Secure routing for internet of things: A survey," *Journal of Network and Computer Applications*, vol. 66, pp. 198–213, 2016.

[4] D. Chen, M. Haenggi, and J. Laneman, "Distributed spectrum-efficient routing algorithms in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 5297–5305, 2008.

[5] J. L. Sobrinho, "Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 541–550, 2002.

[6] M. Saad, "Optimal spectrum-efficient routing in multihop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 5822–5826, 2009.

[7] M. Saad, "Reduced-complexity spectrum-efficient routing in TDMA multihop wireless networks," in *Proceedings of the 2010 IEEE Symposium on Computers and Communications (ISCC)*, pp. 617–621, Riccione, Italy, June 2010.

[8] M. Saad, "On optimal spectrum-efficient routing in TDMA and FDMA multihop wireless networks," *Computer Communications*, vol. 35, no. 5, pp. 628–636, 2012.

[9] M. Saad, "Joint optimal routing and power allocation for spectral efficiency in multihop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 5, pp. 2530–2539, 2014.

[10] M. Martens and M. Skutella, "Flows on few paths: algorithms and lower bounds," *Networks. An International Journal*, vol. 48, no. 2, pp. 68–76, 2006.

[11] T. H. Szymanski, "Max-flow min-cost routing in a future-internet with improved QoS guarantees," *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1485–1497, 2013.

[12] F. Shahrokhi and D. W. Matula, "The maximum concurrent flow problem," *Journal of the ACM*, vol. 37, no. 2, pp. 318–334, 1990.

[13] P. Djukic and S. Valaee, "Delay aware link scheduling for multihop TDMA wireless networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, pp. 870–883, 2009.

[14] L. A. Wolsey, *Integer Programming*, John Wiley & Sons, New York, NY. USA, 1998.

[15] R. Guérin and A. Orda, "Computing shortest paths for any number of hops," *IEEE/ACM Transactions on Networking*, vol. 10, no. 5, pp. 613–620, 2002.

[16] H. Zhang and P. Gburzynski, "A variable slot length TDMA protocol for personal communication systems," *Wireless Personal Communications*, vol. 22, no. 3, pp. 409–432, 2002.

[17] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, NY, USA, 1991.


Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
Advances in Multimedia
Advances in Civil Engineering
Hindawi
Submit your manuscripts at
www.hindawi.com
Journal of Control Science and Engineering
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration