Spatial Reusability-Aware Routing in Multi-Hop Wireless Networks

Tong Meng, *Student Member, IEEE,* Fan Wu, *Member, IEEE,* Zheng Yang, *Member, IEEE,* Guihai Chen, *Member, IEEE,* Athanasios V. Vasilakos, *Senior Member, IEEE*

Abstract—In the problem of routing in multi-hop wireless networks, to achieve high end-to-end throughput, it is crucial to find the "best" path from the source node to the destination node. Although a large number of routing protocols have been proposed to find the path with minimum total transmission count/time for delivering a single packet, such transmission count/time minimizing protocols cannot be guaranteed to achieve maximum end-to-end throughput. In this paper, we argue that by carefully considering spatial reusability of the wireless communication media, we can tremendously improve the end-to-end throughput in multi-hop wireless networks. To support our argument, we propose spatial reusability-aware single-path routing (SASR) and anypath routing (SAAR) protocols, and compare them with existing single-path routing and anypath routing protocols, respectively. Our evaluation results show that our protocols significantly improve the end-to-end throughput compared with existing protocols. Specifically, for single-path routing, the median throughput gain is up to 60%, and for each source-destination pair, the throughput gain is as high as 5.3x; for anypath routing, the maximum per-flow throughput gain is 71.6%, while the median gain is up to 13.2%.

Index Terms—Routing, Wireless Network, Protocol Design

1 Introduction

Due to limited capacity of wireless communication media and lossy wireless links [30], it is extremely important to carefully select the route that can maximize the end-to-end throughput, especially in multi-hop wireless networks. In recent years, a large number of routing protocols (*e.g.*, [4], [14], [20], *etc.*) have been proposed for multi-hop wireless networks. However, a fundamental problem with existing wireless routing protocols is that minimizing the overall number (or time) of transmissions to deliver a single packet from a source node to a destination node does not necessarily maximize the end-to-end throughput. A detailed example will be presented in Section 3.2 to show this observation.

In this paper, we investigate two kinds of routing protocols, including single-path routing and anypath routing. The task of a single-path routing protocol is to select a cost minimizing path, along which the packets are delivered from the source node to the destination node. Recently, anypath routing (*e.g.*, [2], [4]) appears as a novel routing technique exploiting the broadcast nature of wireless communication media to improve the end-to-end throughput. It aggregates the power of multiple relatively weak paths to form a strong path, by welcoming any intermediate node who overhears the packet to participate in packet forwarding. Most of existing routing protocols, no matter single-path routing protocols or anypath routing protocols, rely on link-quality aware routing metrics, such as link transmission count-based metrics (*e.g.*, ETX [6] and EATX [32]) and link transmission time-based metrics (*e.g.*, ETT [7] and EATT [13]). They simply select the (any)path that minimizes the overall transmission counts or transmission time for delivering a packet.

However, an important property of the wireless communication media, which distinguishes it from traditional wired communication media, is the spatial reusability. Specifically, because wireless signals fade during propagation, two links are free of interference if they are far away enough, and thus can transmit at the same time on the same channel. To the best of our knowledge, most of the existing routing protocols do not take spatial reusability of the wireless communication media into account. Our example in Section 3.2 will show the improper usage of routing metrics by existing routing protocols, when spectrum spatial reusability is not considered. In this primer work, we argue that by carefully considering spatial reusability of the wireless communication media, we can tremendously improve the end-to-end throughput in multi-hop wireless networks (*i.e.*, up to $5.3\times$ throughput gain in single-path routing and up to 71.6% gain in anypath routing shown by our evaluation results).

This work was supported in part by the State Key Development Program for Basic Research of China (973 project 2014CB340303), in part by China NSF grant 61422208, 61472252, 61272443, 61171067, and 61133006, in part by CCF-Intel Young Faculty Researcher Program and CCF-Tencent Open Fund, in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, and in part by Jiangsu Future Network Research Project No. BY2013095-1-10. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

T. Meng, F. Wu, and G. Chen are with the Shanghai Key Laboratory of Scalable Computing and Systems, Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. E-mails: mengtong@sjtu.edu.cn, {fwu,gchen}@cs.sjtu.edu.cn.

Z. Yang is with the School of Software, Tsinghua University, China. E-mail: yangzheng@tsinghua.edu.cn.

A. V. Vasilakos is with the Department of Computer Science, Lulea University of Technology, Sweden. E-mail: athanasios.vasilakos@ltu.se.

F. Wu is the corresponding author.

The detailed contributions of our work are as follows.

- To the best of our knowledge, we are the first to explicitly consider spatial reusability of the wireless communication media in routing, and design practical spatial reusability-aware single-path routing (SASR) and anypath routing (SAAR) protocols.
- We formulate the problem of spatial reusability-aware single-path routing as a binary program, and propose two complementary categories of algorithms for path selection. While one category (SASR-MIN and SASR-FF) tends to exploit the best performance of the paths, the other category (SASR-MAX) evaluates the performance of the paths in the worst case.
- We further investigate the spectrum spatial reusability in anypath routing, and propose SAAR algorithm for participating node selection, cost calculation, and forwarding list determination.
- We have evaluated SASR algorithms and SAAR algorithm with different data rates in NS-2. The evaluation results show that our algorithms significantly improve the end-to-end throughput compared with existing ones. Specifically, for single-path routing, a throughput gain up to $5.3\times$ with a median of more than 60% is achieved in the case of single-flow, and an average gain of more than 20% is achieved with multiple flows; for anypath routing, a median gain of 13.2% and the maximum gain up to 71.6% can be realized.

The rest of the paper is organized as below. In Section 2, we briefly review related works. In Section 3, we introduce the preliminaries as well as a motivating example. In Section 4, we present our algorithms for reusability-aware single-path routing. In Section 5, we present the algorithm for reusability-aware anypath routing. In Section 6, we discuss the implementation issues of the proposed algorithms. In Section 7, we show the evaluation results. In Section 8, we conclude the paper and point out future work directions.

2 Related Work

In this section, we briefly review related works on metric design and protocol implementation. We also compare our work with those on joint routing problems, as well as other works considering reusability.

2.1 Routing Metrics

There are a number of works on wireless routing metrics. For single-path routing, several link-quality aware metrics [1], [6], [7], [9] were proposed. RTT [1] weighed the cost of single wireless link by the round trip delay of probe packets on it; ETX [6] assigned the link cost with its expected number of transmissions to successfully deliver a packet. Based on ETX, the authors in [9] designed ETOP metric considering links' actual position on the path. In addition, incorporating the multi-rate ability, ETT [7] took the expected transmission time of a link as its cost; and EMTT [31] extended the work to multicast. What's more, [27] provided some principles for routing metric design.

There're also metrics suitable for anypath routing [4], [13], [32]. Chachulski provided ETOX in [4] which considers opportunistic receptions at any forwarder. In [32], the EATX metric was defined to reflect overall transmissions in any-path forwarding. Laufer *et al.* [13] adopted EATX as the hyperlink cost, and defined the anypath cost composed of the hyperlink cost and the remaining cost.

However, existing routing metrics tend to calculate path cost using some mechanism of lossless combination of link costs. For example, the ETX value of a path is the addition of each link's ETX [6]. Similarly, Laufer calculated the anypath cost while considering all the forwarders' costs [13]. Besides, the guidelines in [27], such as consistency, ignored the effect of reusability. Such lossless mechanism thus misses the opportunity of exploiting spectrum spatial reusability in wireless media.

2.2 Routing Protocols

The earliest single-path routing protocols [3], [10], [17], [18] applied Dijkstra algorithm for route selection. When it comes to anypath routing, for example, ExOR [2] appeared as a coordination mechanism between forwarders; MORE [4] broke such coordination where all the forwarders worked according to their workload. Besides, MORE introduced network coding into anypath routing. On that basis, [13] proposed the shortest anypath first (SAF) algorithm to determine the forwarders' priorities, and proved its optimality; [19] incorporated rate control and used a notion called credit to realize flow control; CodeOR [14] enabled concurrent transmissions of a window of segments; SOAR [24] considered the problem of path divergence and rate limitation to efficiently support multiple flows; SourceSync [20] synchronized senders to achieve combined signals which lowers the packet error rate. Besides, [23] developed an optimization framework to exploit communication opportunities arising by chance; Hu *et al.* [8] proposed POR based on a per-packet feedback mechanism.

Because the above routing protocols were designed based on existing transmission cost minimizing routing metrics, they cannot guarantee maximum end-to-end throughput when spatial reusability cannot be ignored. In addition, different from works such as [2] and [20], which should to some degree rely on synchronization between nodes, the throughput improvements of our algorithms in this work do not need MAC-layer coordination.

2.3 Other Related Works

Some existing cross-layer approaches jointly consider routing and link scheduling (*e.g.*, [11], [16], [29]). Zhang *et al.* [29] formulated joint routing and scheduling into an optimization problem, and solved the problem with a column generation method. Pan *et al.* [16] dealt with the joint problem in cognitive radio networks considering the vacancy of licensed bands. Jones *et al.* [11] implemented k-tuple network coding and proved throughput optimality of their policy. Although these works can provide good performance theoretically, they need centralized control to realize MAC-layer scheduling, and to eliminate transmission contention. The algorithms proposed in this work do not require any scheduling, and the SASR algorithms can be implemented in a distributed manner.

Last but not least, there are also works aimed at exploiting spatial reusability. Specifically, the authors in [12] considered the trade-off between spatial reuse and data rate, and proposed a decentralized power and rate control algorithm for higher network capacity. Zhai *et al.* [28] investigated the optimum carrier sensing range for throughput maximization. However, none of these works deal with the problem of route selection.

3 Technical Preliminaries

In this section, we introduce the preliminary knowledge related to our work, and provide a motivating example to illustrate the importance of exploiting spectrum spatial reusability for routing in multi-hop wireless networks.

3.1 System Model

We consider a static multi-hop wireless network with a set of N nodes. For clarity, we assume that the nodes use the same transmission rate, and do not employ any power control scheme in this work.[1]

Let p_{ij} be the link delivery probability from node i to node j, *i.e.*, if a packet is transmitted from node i to node j, then with probability p_{ij} the packet can be decoded. That is to say, to deliver a packet from node i to node j, node i is expected to do

$$z_i = \frac{1}{p_{ij} \times p_{ji}}, \tag{1}$$

times of transmissions, when MAC-layer acknowledgment is required. We note that in practice, the probability of p_{ij} is related to packet size of data packet or MAC-layer ACK. This is commonly considered in the single path routing as Expected Transmission Count metric (ETX) [6]. Let T_{data} and T_{ack} denote the transmission time of a data packet and an acknowledgment, respectively. Then, the expected time to deliver a packet from node i to node j is

$$\begin{aligned} t_{ij} &= z_i \times T_{data} + z_i \times p_{ij} \times T_{ack} \\ &= \frac{T_{data}}{p_{ij} \times p_{ji}} + \frac{T_{ack}}{p_{ji}}. \end{aligned} \tag{2}$$

In the case of anypath routing (*e.g.*, [4], [13]), the hyperlink from a sender to a set of forwarders and the end-to-end acknowledgment are usually used instead of previous deterministic link and MAC-layer ACK, respectively. Let $F_i \subset N$ be the forwarding set of node i. Then, to deliver a packet from node i to at least one of the nodes in its forwarding set F_i, the expected number of transmissions needed to be done by node i is

$$z_{iF_i} = \frac{1}{1 - \prod_{j \in F_i}(1 - p_{ij})}. \tag{3}$$

This cost metric is called the expected number of anypath transmissions (EATX) [4], [32]. Since the packets are normally sent in batches and only an end-to-end ACK is needed for a whole batch in anypath routing, the cost of ACK is very small compared with the total size of the packets in the batch and can normally be ignored [4]. Therefore, the expected time to deliver a packet from node i to at least one of the nodes in its forwarding set F_i is

$$\begin{aligned} t_{iF_i} &= z_{iF_i} \times T_{data} \\ &= \frac{T_{data}}{1 - \prod_{j \in F_i}(1 - p_{ij})}. \end{aligned} \tag{4}$$

Since wireless signal fades in the process of propagation, two wireless (hyper-)links can work simultaneously, if they are spatially far away enough from each other. We define *non-interfering set* I, in which any pair of (hyper-)links are out of the interference range of each other, *i.e.*, the (hyper-)links in the same non-interfering set can work at the same time.

3.2 Improper Usage of ETX/EATX

Although ETX/EATX has been widely incorporated into many single-path/anypath routing protocols, it is still not properly used, especially in multi-hop wireless networks. There are two reasons:

1) ETX/EATX is designed to capture the quality of a single-hop wireless link/hyperlink. It does not naturally indicate the transmission capability of an end-to-end (any)path.
2) ETX/EATX-based routing protocols tend to choose the route that minimizes the sum of the ETXs/EATXs of the links/hyperlinks involved. Since the wireless communication media has the property of spatial reusability, minimizing the total number of transmissions to deliver a packet from a source node to a destination node does not necessarily maximize the end-to-end throughput.

Here, we use a toy example as shown in Fig. 1 to illustrate the importance of considering spatial reusability of the communication media in single-path routing in wireless networks. In the example, we have four intermediate nodes $\{A, B, C, D\}$ between source node Src and destination node Dst. The dashed circle centered at each of the nodes indicates the interference range of the

[1] However, our approach can be extended to adapt to multiple transmission rates, as long as the conflict graph of links can be calculated. We leave it to our future work.

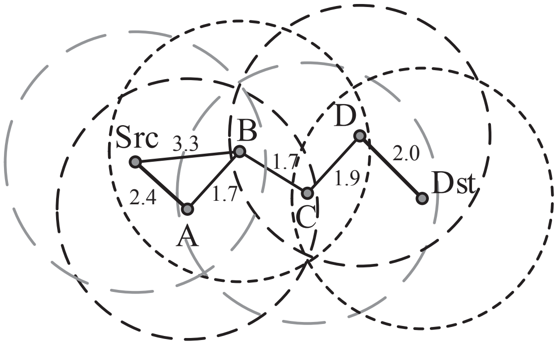


Fig. 1. Importance of Spatial Reusability

node; and the ETX cost is marked beside each of the wireless links.

There are two paths from node Src to node Dst:

$$\begin{aligned}\text{Path I} &: \quad Src - B - C - D - Dst,\\ \text{Path II} &: \quad Src - A - B - C - D - Dst.\end{aligned}$$

The ETX cost of path I and path II is $3.3+1.7+1.9+2.0 = 8.9$ and $2.4+1.7+1.7+1.9+2.0 = 9.7$, respectively. Since path I has a smaller ETX cost, it is normally selected by traditional ETX-based routing protocols, and is expected to have better performance. However, our simulation results show that path II achieves an average end-to-end throughput of 753 Kbps, which is 10.2% higher than 683 Kbps achieved by path I, when the transmission rate is 11 Mbps. This result indicates that the ETX minimizing path is not necessarily the throughput maximizing path in multi-hop wireless networks. If we look into the toy example, we can find that link (Src, A) and link (D, Dst) are out of the interference range of each other, and thus can work simultaneously. Therefore, it is necessary to "fuse" spatially non-interfering links' costs when doing path selection. By fusing costs, we mean that the costs of a set of non-interfering links should be considered as a whole, instead of directly summing them up. In the above example, if we fuse the costs of link (Src, A) and link (D, Dst), and pick the larger cost of the two as the fused cost, the cost of path II becomes 7.7, which is smaller than that of path I.[2] Thus, when the spatial reusability of wireless communication media is taken into account, the higher throughput path can be selected.

A similar example can be found for the case of EATX-based anypath routing. Due to limitations of space, we do not present the example in this paper.

Considering the improper usage of ETX/EATX routing metric in existing works, we propose to take into account the spatial reusability of wireless communication media during path selection in wireless networks, and will present our spatial reusability-aware routing protocols in the following sections.

4 Spatial Reusability-Aware Single-Path Routing (SASR)

We first consider the spatial reusability-aware path cost evaluation for single-path routing. Given each of the paths found by an existing source routing protocol (*e.g.*, DSR [10]), our SASR algorithm calculates the spatial reusability-aware path cost of it. Then, the path with the smallest cost can be selected.

As mentioned in Section 3.1, we can use a non-interfering set I to represent a group of wireless links that can work simultaneously. The fused cost of the non-interfering set I can be defined as the largest link delivery time in the set

$$c(I) = \max\{t_{ij} | (i,j) \in I\}. \tag{5}$$

Given the collection $\mathcal{I}$ of the non-interfering sets on a path P, the spatial reusability-aware path delivery time is

$$C = \sum_{I \in \mathcal{I}} c(I). \tag{6}$$

For ease of expression, we use link/path delivery time and cost interchangeably in the rest of the paper. Then, the key issue here is to calculate the collection $\mathcal{I}$ of the non-interfering sets[3], given the interference condition of the links on the path P. We note that interference among links on the path can be represented by a conflict graph $G = \{P, E\}$, in which the vertices and the edges represent the links and interferences, respectively. Here, $E = \{[(i,j),(i',j')] |$ links (i,j) and (i',j') have interference between each other$\}$. Like many works utilizing the conflict graph [21], we compute G with measurement-based techniques [15], [22] within $O(|P|)$ time. Then $\mathcal{I}$ must be a collection of maximal independent sets on the conflict graph.

In this section, we present two categories of algorithms to calculate the collection $\mathcal{I}$. One aims to find a collection $\mathcal{I}$ that minimizes the path cost, while the other one targets at finding the worst possible fused path cost and its corresponding collection $\mathcal{I}$. These two categories of algorithms are complementary to each other. While path cost minimizing collection reflects the best possible performance of the path, the path cost maximizing collection indicates how bad the path can be in the worst case.

4.1 Cost Minimizing Fusion

The problem of finding the collection of non-interfering sets that minimizes the path cost, can be formulated into a binary program as follows.
Objective:

$$Minimize \quad C = \sum_{I \in \mathcal{M}} x(I)c(I)$$

[2] It is sufficient to fuse the ETX costs to show the effect of spatial reusability. However, as shown in Section 4, for link cost fusion, we should consider the expected link delivery time $t_{Src,A}$ and $t_{D,Dst}$ instead of the ETX costs.

[3] The calculation of collection $\mathcal{I}$ requires no MAC-layer scheduling in the packet delivery process. Actually, the proposed algorithms are all MAC-independent, which is one of the advantages of this work.

Subject to:

$$\sum_{I:(i,j)\in I} x(I) = 1, \qquad (i,j) \in P, \tag{7}$$

$$x(I) \in \{0,1\}, \qquad I \in \mathcal{M}, \tag{8}$$

where, $\mathcal{M}$ is a collection of all the non-interfering sets on path P. Here, constraint (7) guarantees that each link is involved in exactly one non-interfering set. Constraint (8) indicates the possible values of $x(I)$. If non-interfering set I is selected to the collection, then $x(I) = 1$; otherwise, $x(I) = 0$.

We note that the above problem of finding the path cost minimizing collection of non-interfering sets can be reduced to the minimum set cover problem [26], which is NP-hard.

4.1.1 Approximation Algorithm for Min-Cost Fusion

Since scale-free networks with degree exponent $2 < \lambda < 3$ possess a diameter $D \sim \ln\ln|N|$ [5], the paths in the network are normally not long. So, we first present an approximation algorithm for finding the path delivery time minimizing collection of non-interfering sets, namely SASR-MIN algorithm, when the collection $\mathcal{M}^\star$ of all the *maximal* non-interfering sets on path P can be calculated efficiently. We note that a non-interfering set corresponds to an independent set in the conflict graph, or equivalently, a clique in the complementary conflict graph. Therefore, the collection $\mathcal{M}^\star$ can be computed in time $O(3^{|P|/3})$ [25]. Generally, SASR-MIN iteratively visits all the maximal non-interfering sets in $\mathcal{M}^\star$ to pick the most cost-effective set among the rest ones, until all the links on path P have been covered by the selected sets. Here, by cost-effectiveness, we mean the average cost, at which a maximal non-interfering set induces, to cover new links, *i.e.*, $c(I)/|I|$.

Algorithm 1 shows the pseudo-code of our SASR-MIN algorithm. We use set Q to store covered links. Then, we iteratively select maximal non-interfering sets to put into the collection $\mathcal{I}$ (Lines 3-15). In each of the iterations, we check every remaining maximal non-interfering set I in $\mathcal{M}^\star$. Since each link should be covered exactly once, if the set I contains already covered links, we need to remove the covered links from I (Lines 5-6). Then, if the set I is not empty, we compare it with the currently most efficient cost factor α in this iteration. If this is a more cost efficient set, we update the factor α, and record the corresponding set (Lines 7-9). At the end of the iteration, we add the cost of the currently most cost-effective maximal non-interfering set into the collection $\mathcal{I}$, and update the total path cost C and the covered set Q correspondingly (Lines 12-14).

Since at least one link is added into the covered set Q in each iteration, Algorithm 1 iterates at most $|P|$ times. In each iteration, at most $|\mathcal{M}^\star|$ cost-effectiveness factors are calculated. Thus, the total run time of Algorithm 1 is $O(|P||\mathcal{M}^\star|)$.

Next, we show the approximation ratio of Algorithm 1.

Algorithm 1: SASR-MIN Algorithm

```
Input: A path $P$, a profile of link delivery time
       $(t_{ij})_{(i,j)\in P}$, and a collection $\mathcal{M}^\star$ of all the
       maximal non-interfering sets on path $P$.
Output: Path delivery time $C$ and corresponding
        collection $\mathcal{I}$ of non-interfering sets.
1  $C \leftarrow 0$;
2  $Q \leftarrow \emptyset$ ;
3  while $Q \neq P$ do
4      $\alpha \leftarrow +\infty$;
5      foreach $I \in \mathcal{M}^\star$ do
6          $I \leftarrow I \setminus Q$;
7          if $I \neq \emptyset \wedge c(I)/|I| < \alpha$ then
8              $\alpha \leftarrow c(I)/|I|$;
9              $Temp \leftarrow I$;
10         end
11     end
12     $C \leftarrow C + c(Temp)$;
13     $\mathcal{I} \leftarrow \mathcal{I} \cup \{Temp\}$;
14     $Q \leftarrow Q \cup Temp$;
15 end
16 return $C$ and $\mathcal{I}$;
```

Theorem 1. *SASR-MIN algorithm achieves an approximation ratio of $H(|P|)$, where $H(|P|)$ is the $|P|$th harmonic number:*

$$H(|P|) = \sum_{k=1}^{|P|} \frac{1}{k} \leq \ln|P| + 1. \tag{9}$$

Proof. Let η_l be the cost-effectiveness of link $l \in I$, where $I \in \mathcal{I}$. Then $\eta_l = c(I)/|I|$. Consequently, the path cost is

$$C = \sum_{i\in P} \eta_l. \tag{10}$$

Next, we number all the links in the order of being covered (for those links covered in the same iteration, number them arbitrarily), and get $\{l_1, l_2, \cdots, l_{|P|}\}$. In the iteration l_k is covered, the picked I must have a cost-effectiveness of at most $OPT/(|P| - k + 1)$, which means that

$$\eta_k \leq \frac{OPT}{|P| - k + 1}. \tag{11}$$

Therefore, the calculated path cost is

$$\begin{aligned} C &= \sum_{i\in P} \eta_l \\ &\leq OPT \sum_{k=1}^{|P|} \frac{1}{k}. \end{aligned} \tag{12}$$

This completes the proof. □

4.1.2 First-Fit Algorithm for Min-Cost Fusion

However, listing all the maximal non-interfering set on path P needs $O(3^{|P|/3})$ time, which is still inefficient when the path P is long. Therefore, we propose a first-fit algorithm, namely SASR-FF, which can achieve good performance in most of the cases.

Algorithm 2: SASR-FF Algorithm

```
   Input: A path $P$, a profile of link cost $(t_{ij})_{(i,j)\in P}$,
          and a link conflict graph $G = \{P, E\}$.
   Output: Path cost $C$ and corresponding collection $\mathcal{I}$
           of non-interfering sets.
1  Sort the links in $P$ by cost in non-increasing order $\mathcal{L}$;
2  $k \leftarrow 0$;
3  $C \leftarrow 0$; $\mathcal{I} \leftarrow \emptyset$;
4  foreach $(i, j) \in \mathcal{L}$ do
5      fused ← FALSE;
6      for $l \leftarrow 1$ to $k$ do
7          reusable ← TRUE;
8          foreach $(i', j') \in \mathcal{I}_l$ do
9              if $[(i, j), (i', j')] \in E$ then
10                 reusable ← FALSE;
11                 break;
12             end
13         end
14         if reusable then
15             $\mathcal{I}_l \leftarrow \mathcal{I}_l \cup \{(i, j)\}$; $c_l \leftarrow \max\{c_l, t_{ij}\}$;
16             fused ← TRUE;
17         end
18     end
19     if not fused then
20         $k \leftarrow k + 1$;
21         $\mathcal{I} \leftarrow \mathcal{I} \cup \{\{(i, j)\}\}$; $c_k \leftarrow t_{ij}$;
22     end
23 end
24 for $l \leftarrow 1$ to $k$ do
25     $C \leftarrow C + c_l$;
26 end
27 return $C$ and $\mathcal{I}$;
```

Algorithm 2 shows the pseudo-code of our SASR-FF algorithm. We first sort the links in P by their costs in non-increasing order, and get an ordered list $\mathcal{L}$ (Line 1). Then, we check each link (i, j) in the ordered list $\mathcal{L}$, and put it into the foremost non-interfering set that does not have any link interfering with it, in the collection $\mathcal{I}$ (Lines 4-18). If no appropriate existing non-interfering set can be found, we create a new set containing the link (i, j) itself, and add it into the collection $\mathcal{I}$ (Lines 19-22). After visiting all the links, we get the collection $\mathcal{I}$ of non-interfering sets, and calculate the path cost C by summing the cost of the non-interfering sets (Lines 24-26). The time complexity of Algorithm 2 is $O(|P|^2)$.

4.2 Cost Maximizing Fusion

In practice, normally there is no MAC-layer synchronization scheme in the wireless networks. Consequently, the wireless links may arbitrarily form non-interfering sets, leading to less cost-efficient end-to-end transmission. Therefore, it also helps to know the worst possible performance (*i.e.*, fused path delivery time) in the process of path selection.

Since finding the cost maximizing collection of non-interfering sets is just the inverse version of the cost minimizing fusion, we can design a similar approximation algorithm as that in previous section, by iteratively picking the least cost-effective maximal non-interfering set (only different from Algorithm 1 in line 4 and 7). Here, we name it SASR-MAX algorithm. Due to limitations of space, we do not present the details of SASR-MAX algorithm.

Considering that cost maximizing fusion does not show superior performance to cost minimizing fusion, we mainly use it as a benchmark or reference in path selection. So in this work, we only consider the pseudo-polynomial time approximation algorithm SASR-MAX, and do not investigate its corresponding fully polynomial greedy algorithm.

5 Spatial Reusability-Aware Anypath Routing (SAAR)

We further consider how to exploit spectrum spatial reusability for anypath routing in this section. In contrast to the single-path routing, which restricts the packets to be forwarded through a predetermined path from the source to the destination, anypath routing enables any intermediate node who overhears the packet to participate in packet forwarding. Therefore, in the case of anypath routing, our objective is to pick a set of participating nodes Q (including the source), and the corresponding profile of "distance"/cost $\vec{C} = (C_i)_{i\in Q}$ and forwarder lists $\vec{F} = (F_i)_{i\in Q}$, to minimize the spatial reusability-aware anypath cost C_{src}. Here, having a smaller C_i means that node i is closer to the destination.

In this section, we first propose the methodology of cost fusion in anypath routing, and then present the design of our spatial reusability-aware anypath routing algorithm (SAAR).

5.1 Anypath Cost Fusion

Suppose that the set of participating nodes Q, the profile of cost $\vec{C}$, and the profile of forwarder lists $\vec{F}$ have been calculated. (We will present the algorithm for calculating Q, $\vec{C}$, and $\vec{F}$ in Section 5.2.)

Given a source/forwarding node $i \in Q$, the probability ω_{ij} that node $j \in F_i$ directly receives a packet from node i, and the packet is not received by any node that is closer to the destination than j in i's forwarding set F_i is

$$\omega_{ij} = \frac{p_{ij} \prod_{k\in F_i \wedge C_k < C_j} (1 - p_{ik})}{1 - \prod_{k\in F_i} (1 - p_{ik})}. \tag{13}$$

Then, we can derive the probability that node $j \in Q$ needs to relay a packet from node i (*s.t.*, $C_i > C_j$), in a recursive way:

$$\Omega(i, j) = \begin{cases} 1, & \text{if } j = i; \\ \sum\limits_{k\in F_i} \omega_{ik} \times \Omega(k, j), & \text{if } j \in Q \wedge C_j < C_i. \end{cases} \tag{14}$$

We note that Ω_{ij} is an integration of delivery probabilities over all the hyperlinks from node i to node j.

After deriving all the nodes' probabilities $(\Omega(Src, i))_{i \in Q}$ as relay from the source, we can calculate the expected time $\bar{t}_{iF_i}$ needed by each node $i \in Q$ to deliver a packet from the source to the destination:

$$\bar{t}_{iF_i} = \Omega(Src, i) \times t_{iF_i}, \tag{15}$$

where t_{iF_i} is defined by formula (4) in Section 3.1.

Given a set of non-interfering hyperlinks I, which can work simultaneously without any interference, we can calculate the fused cost of set I as the largest expected hyperlink delivery time in the set:

$$c(I) = \max\{\bar{t}_{iF_i} | (i, F_i) \in I\}. \tag{16}$$

Consequently, given a collection $\mathcal{I}$ of all the sets of non-interfering hyperlinks, the total cost for delivering a packet from the source to the destination is

$$C_{Src} = \sum_{I \in \mathcal{I}} c(I). \tag{17}$$

We will also present the way to compute $\mathcal{I}$ in Section 5.2.

We note that SASR is actually a special case of SAAR, when each hyperlink only contains a single wireless link.

5.2 Algorithm for Min-Cost Anypath Fusion

Algorithm 3: SAAR Algorithm

```
Input: A network graph $G = \{N, E\}$, a source node Src, a destination node Dst.
Output: A set of participating nodes $Q$, and the
        corresponding profile of cost $C$ and
        forwarder lists $F$.
1  foreach $i \in N$ do
2      $C_i \leftarrow +\infty$; $F_i \leftarrow \emptyset$; $\Omega(i, i) \leftarrow 1$;
3  end
4  $C_{Dst} \leftarrow 0$; $q \leftarrow Dst$; $Q \leftarrow \{Dst\}$; $\mathcal{I} \leftarrow \{\{Dst\}\}$;
5  while $q \neq Src$ do
6      foreach $(i, q) \in E \wedge i \notin Q$ do
7          $F_i \leftarrow F_i \cup \{q\}$; $\Omega_i \leftarrow \Omega$;
8          foreach $j \in Q$ do
9              $\Omega_i(i, j) \leftarrow 0$;
10             foreach $k \in F_i \wedge C_k < C_j$ do
11                 $\Omega_i(i, j) \leftarrow \Omega_i(i, j) + \omega_{ik} \times \Omega_i(k, j)$;
12             end
13         end
14         $(C_i, \mathcal{I}_i) \leftarrow CalculateCost(i, \mathcal{I}, \Omega_i, F)$;
15     end
16     $q \leftarrow \underset{i \in N \setminus Q}{argmin}(C_i)$; $Q \leftarrow Q \cup \{q\}$;
17     $\Omega \leftarrow \Omega_q$; $\mathcal{I} \leftarrow \mathcal{I}_q$;
18 end
19 return $Q$, $C$, and $F$;
```

Algorithm 4: Function *CalculateCost*()

```
Input: A node $i$, a collection of non-interfering sets
       $\mathcal{I}$, a profile of relay probabilities $\Omega$, and a
       profile of forwarding sets $F$.
Output: The fused anypath cost $C_i$ from node $i$ to
        the destination node, and an updated
        collection of non-interfering sets $\mathcal{I}$.
fused $\leftarrow$ FALSE; $C_i \leftarrow 0$;
foreach $I \in \mathcal{I}$ do
    reusable $\leftarrow$ TRUE;
    foreach $(j, F_j) \in I$ do
        if $(i, F_i)$ interfere with $(j, F_j)$ then
            reusable $\leftarrow$ FALSE; break;
        end
    end
    if reusable then
        $I \leftarrow I \cup \{(i, F_i)\}$; fused $\leftarrow$ TRUE; break;
    end
end
if not fused then
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{\{(i, F_i)\}\}$;
end
foreach $I \in \mathcal{I}$ do
    $C_i \leftarrow C_i + c(I)$;
end
return $C_i$ and $\mathcal{I}$;
```

In this section, we present the spatial reusability-aware anypath routing algorithm (SAAR). Since finding the minimized end-to-end cost considering the spatial reusability is NP-hard, our algorithm SAAR is designed to calculate a suboptimal route, which can achieve superior performance to existing anypath routing protocols in most of the cases.

Algorithm 3 shows the pseudo-code of our SAAR algorithm. Given a network graph $G = (N, E)$, where N is the set of nodes and E is the set of wireless links, Algorithm 3 calculates a min-cost anypath route from the source to the destination, including a set of participating nodes Q, and the corresponding profile of cost C and forwarder lists F. Generally, we start from the destination node Dst (Line 4), and iteratively find the minimum cost node among the remaining nodes to add into the participating node set. Specifically, in each iteration, for each node i who has a wireless link to the last picked min-cost node q, we update its forwarding set by adding q as a new forwarder (Line 7), and calculate the relaying probability matrix Ω_i if a packet is sent from node i (Lines 8-13). By doing so, the recursive computation of relaying probability in Equation (14) is amortized in different iterations. Then, we compute node i's current cost and the corresponding collection of non-interfering sets $\mathcal{I}_i$ if node i is picked, by calling function *CalculateCost*(), which will be presented shortly (Line 14). At the end of each iteration, the cost of nodes who have direct connection to the already picked nodes in Q are updated. We now pick the minimum cost node q among the remaining nodes and add it into the participating node set Q (Line 16). We also record node q's corresponding relaying probability matrix Ω_i and collection of non-interfering sets $\mathcal{I}_i$ (Line 17). Finally, when the source node is picked and added into the participating node set, the algorithm halts and returns the results.

The pseudo-code of function *CalculateCost*() is shown by Algorithm 4. It takes a collection of non-interfering sets $\mathcal{I}$, a profile of relay probabilities Ω, and a profile of forwarding sets F as inputs, and outputs the fused anypath cost C_i for the given node i and an updated collection of non-interfering sets $\mathcal{I}$ including node i. Specifically, we check the non-interfering set in $\mathcal{I}$ one by one, and add hyperlink (i, F_i) into the first set, in which it does not cause any interference (Lines 2-12). If no such set can be found, we create a new non-interfering set containing the hyperlink (i, F_i) itself (Lines 13-15). Finally, we sum the fused costs of all the sets in $\mathcal{I}$ to get node i's fused anypath cost C_i.

On one hand, the iteration in Algorithm 3 repeats at most $|N|$ times. In each iteration, the calculation of Ω_i takes time $O(|N|^2)$, and function *CalculateCost*() is called at most $|N|$ times. On the other hand, the running time of function *CalculateCost*() is $O(|N|)$. Therefore, the run time of the SAAR algorithm is $O(|N|^3)$.

6 Implementation Issues

In this section, we discuss the important issues on the implementation of the proposed algorithms.

First, similar manner as [22] can be used to create the interference profile at each node. It takes trials only with the same number of nodes in the network, during which a node senses no signal from an interference-free node. What's more, to achieve lower complexity, the imperfect measurement-calibrated propagation model in [33] can also be adopted to generate conservative conflict graph.

Second, the three SASR algorithms can be incorporated with existing single path routing protocols such as DSR in a distributed manner. Generally, in DSR, the source node sends route request for the destination. When the other nodes receive the route request, they will include their own addresses and forward the request. Then, the destination node calculates the path cost of the received route request, and piggybacks the route reply through the reverse path if necessary. To implement the SASR algorithms, each forwarder needs to add the interfering information into the route request. Specifically, on receiving a route request, a node checks all its previous forwarders, and adds one flag bit to the route request for each of them denoting whether the forwarder is interference-free. According to the flag bits in the received request, the destination node can build the non-interfering sets, and run the SASR algorithm to calculate the path cost. So except adding flag bits in route request, the operations of a forwarder in the process of finding the route from the source to destination are the same as in the DSR protocol. The SASR algorithms are executed only at destination nodes.

Third, as most existing works on anypath routing (*e.g.*, [4], [8], [14]), SAAR is a central algorithm that calculates the path costs and forms the forwarder lists. It can be utilized to determine the workloads of the forwarders based on MORE [4].

7 Evaluation

We evaluated the performance of our SASR and SAAR algorithms in NS-2. Considering that all the nodes use the same transmission rate, we can compare our algorithms with transmission count-based routing protocols and metrics. To be specific, we compared them with the ETX-based DSR [6] (denoted by DSR-ETX) and the shortest anypath first (SAF) algorithm [13] based on MORE [4][4], respectively. Table 1 lists the parameters in our simulation. Specifically, we used the *setdest* tool in NS-2 to uniformly distribute 80 nodes in a 2000 meter $\times$ 2000 meter area, and considered two data rates of 802.11, including 11 Mbps and 54 Mbps. We used CBR to generate 1500-byte packets at high enough rates. In addition, RTS/CTS is turned off in all the simulations.

Parameter	Value
Number of Nodes	80
Terrain Area	2000 m × 2000 m
RTS/CTS	OFF
Packet Size	1500 Bytes
Traffic Generator	CBR
CBR Rate	5 Mbps / 20 Mbps
802.11 Data Rate	11 Mbps / 54 Mbps

TABLE 1
Simulation Parameters Setup

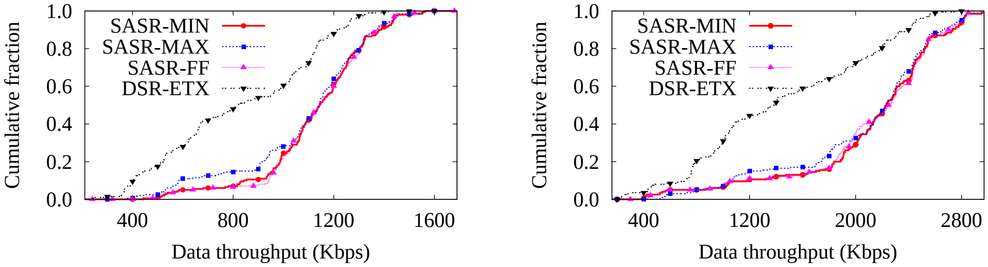
For single-path routing, we first evaluated the single-flow scenario. To be detailed, we randomly picked 200 source-destination pairs, from those that result in different routing decisions from the compared routing algorithms for clarity. The throughput of each source-destination pair was averaged over the UDP transmission that lasted 10 minutes. Then, with the existence of multiple flows, we simulated with 2 and 3 concurrent flows. In each set of simulations, we calculated the average end-to-end throughputs of over 50 groups of source-destination pairs. Furthermore, for anypath routing, we randomly picked 200 source-destination pairs, as well, and focused on the single-flow case.

7.1 Performance of SASR Algorithms

7.1.1 *Single-Flow Scenario*

Fig. 2 shows the cumulative distributions of throughputs achieved by four routing algorithms, including SASR-MIN, SASR-MAX, SASR-FF, and DSR-ETX. We can see that all the three SASR algorithms outperform DSR-ETX.

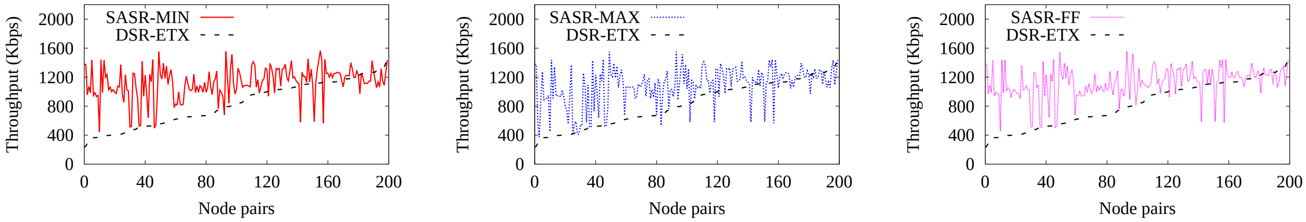
[4] We note that most recent works on anypath routing are based on MORE, while introducing new techniques such as per-packet feedback [8]. As a future work, our SAAR algorithm can be extended to adopt these techniques.



(a) Data Rate: 11 Mbps

(b) Data Rate: 54 Mbps

Fig. 2. CDF of Throughputs for Single-Path Routing

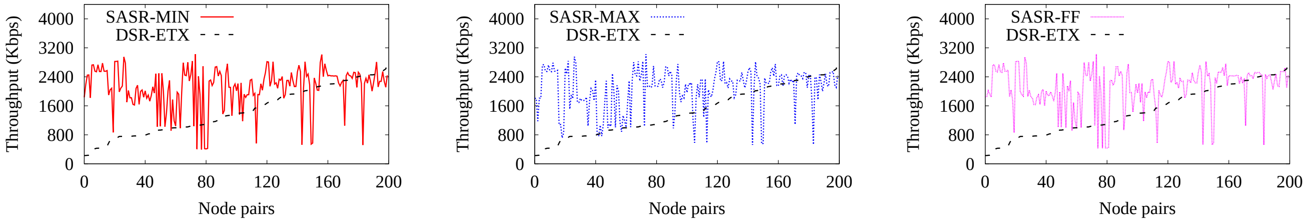


(a) SASR-MIN vs. DSR-ETX

(b) SASR-MAX vs. DSR-ETX

(c) SASR-FF vs. DSR-ETX

Fig. 3. Pairwise Results of Single-Path Routing Algorithms at 11 Mbps Data Rate



(a) SASR-MIN vs. DSR-ETX

(b) SASR-MAX vs. DSR-ETX

(c) SASR-FF vs. DSR-ETX

Fig. 4. Pairwise Results of Single-Path Routing Algorithms at 54 Mbps Data Rate

They all achieve a median throughput gain of around 40% under 11 Mbps, and more than 60% under 54 Mbps. What's more, under both data rates, the three SASR algorithms realize a throughput gain of 10% in the worst case. Therefore, the performance of SASR algorithms is better under higher data rate, because a higher data rate needs a shorter transmission time, which results in more opportunities of spatial reuse between links. In addition, SASR-MIN/FF get better results than SASR-MAX. This implies that cost minimizing fusion tends to give more reasonable path cost than maximizing fusion. Fig. 2 also shows that SASR-FF achieves similar performance as SASR-MIN. Considering its low computation complexity, the SASR-FF algorithm is likely to achieve satisfactory performance in practice.

We present detailed pairwise comparisons in Fig. 3 and Fig. 4. The 200 simulated node pairs are sorted by their throughputs under DSR-ETX in a non-decreasing order. The SASR algorithms can realize up to 3.9× and 6.3× throughputs compared with DSR-ETX under 11 Mbps and 54 Mbps, respectively, let alone those node pairs that suffer from hidden terminals under DSR-ETX. We note that the throughput gains tend to be higher for those node pairs which perform bad under DSDR-ETX, because these pairs correspond to paths with larger hop-counts, which contain more interference-free links. Moreover, as shown by Fig. 3(a), 22% node pairs have doubled throughputs when SASR-MIN is used. While in Fig. 4(a), there are 35% of such node pairs. Consequently, similar to the cumulative distributions in Fig. 2, SASR algorithms can achieve larger throughput gains under 54 Mbps. We can also observe that cost maximizing fusion has inferior performance to minimizing fusion, especially for those pairs with small throughputs under DSR-ETX. Besides, the performance fluctuations of SASR algorithms in Fig. 3 and Fig. 4 are due to different numbers of non-interfering links in different areas of the topology, *i.e.*, the SASR algorithms do not have much potential to improve the throughputs in those areas with limited number of non-interfering links.

However, owing to cost fusion, SASR algorithms inevitably need longer routing requests, and select paths

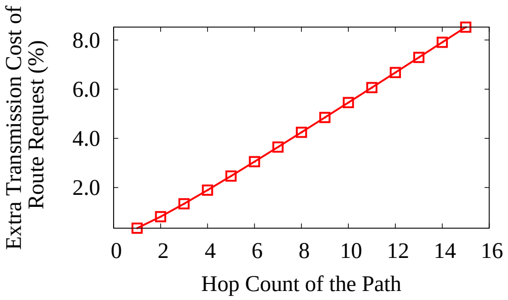


Fig. 5. Extra Transmission Cost of Route Request Induced by SASR Algorithms

that require more transmissions, and thus consume more energy. As mentioned above, a node adds one additional bit for each of the previous forwarders in the route request. Hence, on one hand, for a path P, there will be $|P|(|P|+1)/2$ such bits in a corresponding route request. On the other hand, in original DSR-ETX, besides the packet header (calculated as 20 bytes), a route request contains the MAC address of each forwarder (48 bits), as well as the cost of each link (calculated as 4 bytes). Fig. 5 illustrates the extra transmission cost of route request. Clearly, the extra cost is below 10% even for a path of as long as 15 hops. If taking the size of a whole batch into account, the extra cost of route request is even lower.

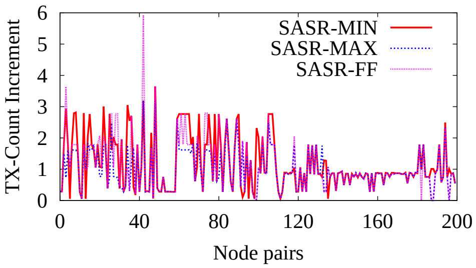
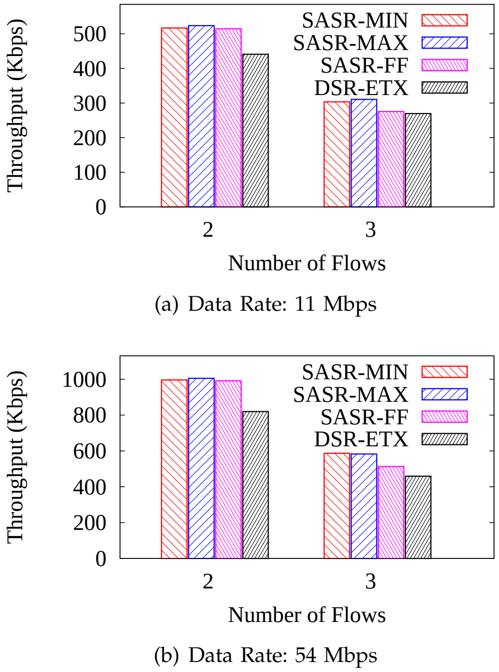


Fig. 6. Overall Transmission Count Increments Induced by SASR Algorithms

What's more, Fig. 6 shows that the increment in transmission counts is not much and acceptable considering the throughput gains. For all the three algorithms, more than 80% of the node pairs only need no more than 2 additional overall transmissions compared with DSR-ETX. Interestingly, the SASR algorithms show greater transmission counts increments for longer paths, from which they achieve higher throughput gains, as well.

7.1.2 *Multi-Flow Scenario*

Fig. 7 presents the average per-flow throughputs in the multi-flow case. With 2 and 3 concurrent flows, the three SASR algorithms can still improve the throughputs compared with DSR-ETX. Specifically, under the data rate of 11 Mbps, the throughput gains of SASR-MIN over DSR-ETX are 17.2% with 2 flows and 12.6% with 3 flows, respectively. When it comes to 54 Mbps, the corresponding throughput gains increase to 21.5% and 27.9%. Considering that with the existence of multiple flows, the SASR algorithms will induce a larger extra transmission cost than in the single-flow case, load balancing throughout the network can be applied to improve the performance of SASR with multiple flows. However, that is beyond the scope of this work.



(a) Data Rate: 11 Mbps

(b) Data Rate: 54 Mbps

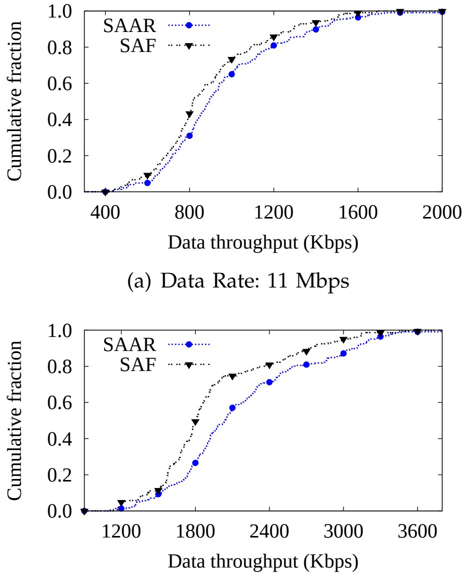
Fig. 7. Average Throughputs with Multiple Concurrent Flows

7.2 Performance of SAAR Algorithm

We present the cumulative distributions of end-to-end throughputs achieved by SAAR and SAF in Fig. 8. For anypath routing, the throughput gains are also more significantly under the data rate of 54 Mbps. The median gains over SAF are 9.3% and 13.2% under 11 Mbps and 54 Mbps, respectively. Considering the more comprehensive interfering situations in anypath routing, although there are more nodes participating in packet forwarding, and more opportunities of concurrent transmissions among hyperlinks, it is non-trivial to achieve as great improvements as in single-path routing. However, the gains in Fig. 8 are quite obvious, as well.

What's more, we provide the pairwise end-to-end throughputs with the scatter plots in Fig. 9. We are glad to find that most of the simulated node pairs display significant gains in throughputs. For the data rate of 11 Mbps, the throughput gain is up to 62.7%. For 54 Mbps, it is up to 71.6%. In addition, there are 26.5% and 33.2% of the node pairs realizing a throughput gain of at least 20% under the two data rates, respectively.

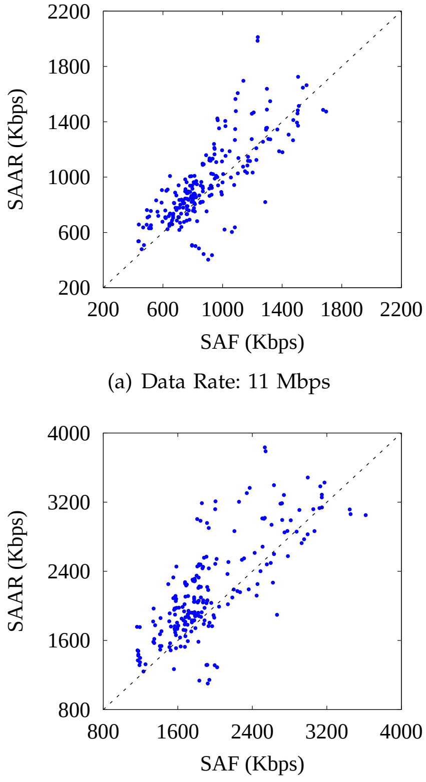
Then, Fig. 10 shows the amount of additional transmissions required by SAAR, compared with SAF. Except



(a) Data Rate: 11 Mbps

(b) Data Rate: 54 Mbps

Fig. 8. CDF of Throughputs for Anypath Routing



(a) Data Rate: 11 Mbps

(b) Data Rate: 54 Mbps

Fig. 9. Pairwise Comparisons between SAAR and SAF

for only one source-destination pair, the increments of transmission counts do not exceed one.

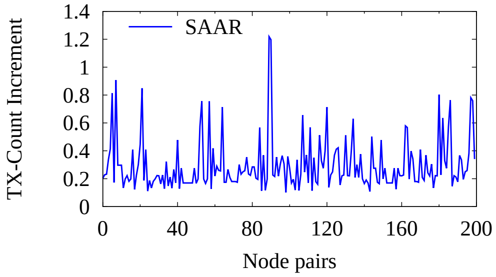


Fig. 10. Overall Transmission Count Increments Induced by SAAR Algorithm

8 Conclusion and Future Work

In this paper, we have demonstrated that we can significantly improve the end-to-end throughput in multi-hop wireless networks, by carefully considering spatial reusability of the wireless communication media. We have presented two protocols, SASR and SAAR, for spatial reusability-aware single-path routing and anypath routing, respectively. We have also implemented our protocols, and compared them with existing routing protocols with the data rates of 11 Mbps and 54 Mbps. Evaluation results show that SASR and SAAR algorithms can achieve more significant end-to-end throughput gains under higher data rates. For the case of single-flow, SASR achieves a throughput gain of as high as $5.3\times$ under 54 Mbps, while for SAAR, the maximum gain can reach 71.6%. Furthermore, in multi-flow case, SASR can also improve the per-flow average throughputs by more than 20%. Meanwhile, the tremendous throughput gains only require acceptable additional transmission overheads. The extra transmission overheads of route request are less than 10% in our evaluation. In 80% cases, the overall transmission counts are increased by no more than 2 with SASR, while for SAAR, most of the increments are below 1.

As for the future work, one direction is to further explore opportunities to improve the performance of our routing algorithms by analyzing special underperforming cases identified in the evaluation. Another direction is to investigate inter-flow spatial reusability, and to optimize system-wide performance.

References

[1] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multi-radio unification protocol for ieee 802.11 wireless networks," in *BROADNETS*, 2004.

[2] S. Biswas and R. Morris, "Exor: opportunistic multi-hop routing for wireless networks," in *SIGCOMM*, 2005.

[3] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. G. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *MOBICOM*, 1998.

[4] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *SIGCOMM*, 2007.

[5] R. Cohen and S. Havlin, "Scale-free networks are ultrasmall," *Phys. Rev. Lett.*, vol. 90, p. 058701, Feb 2003. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.90.058701

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

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

[8] W. Hu, J. Xie, and Z. Zhang, "Practical opportunistic routing in high-speed multi-rate wireless mesh networks," in *MOBIHOC*, 2013.

[9] G. Jakllari, S. Eidenbenz, N. W. Hengartner, S. V. Krishnamurthy, and M. Faloutsos, "Link positions matter: A noncommutative routing metric for wireless mesh network," in *INFOCOM*, 2008.

[10] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, vol. 353, pp. 153–181, 1996.

[11] N. M. Jones, B. Shrader, and E. Modiano, "Optimal routing and scheduling for a simple network coding scheme," in *INFOCOM*, 2012.

[12] T.-S. Kim, J. C. Hou, and H. Lim, "Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks," in *MOBICOM*, 2006.

[13] R. P. Laufer, H. Dubois-Ferrière, and L. Kleinrock, "Multirate anypath routing in wireless mesh networks," in *INFOCOM*, 2009.

[14] Y. Lin, B. Li, and B. Liang, "Codeor: Opportunistic routing in wireless mesh networks with segmented network coding," in *ICNP*, 2008.

[15] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link interference in static multi-hop wireless networks," in *Internet Measurment Conference*, 2005.

[16] M. Pan, C. Zhang, P. Li, and Y. Fang, "Joint routing and link scheduling for cognitive radio networks under uncertain spectrum supply," in *INFOCOM*, 2011.

[17] C. E. Perkins and E. M. Belding-Royer, "Ad-hoc on-demand distance vector routing," in *WMCSA*, 1999.

[18] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," in *SIGCOMM*, 1994.

[19] B. Radunovic, C. Gkantsidis, P. B. Key, and P. Rodriguez, "An optimization framework for opportunistic multipath routing in wireless mesh networks," in *INFOCOM*, 2008.

[20] H. Rahul, H. Hassanieh, and D. Katabi, "Sourcesync: a distributed wireless architecture for exploiting sender diversity," in *SIGCOMM*, 2010.

[21] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *INFOCOM*, 2006.

[22] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Measurement-based models of delivery and interference in static wireless networks," in *SIGCOMM*, 2006.

[23] E. Rozner, M. K. Han, L. Qiu, and Y. Zhang, "Model-driven optimization of opportunistic routing," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 594–609, 2013.

[24] E. Rozner, J. Seshadri, Y. A. Mehta, and L. Qiu, "Soar: Simple opportunistic adaptive routing protocol for wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 12, pp. 1622–1635, 2009.

[25] E. Tomita, A. Tanaka, and H. Takahashi, "The worst-case time complexity for generating all maximal cliques and computational experiments," *Theoretical Computer Science*, vol. 363, no. 1, pp. 28–42, 2006.

[26] V. V. Vazirani, *Approximation Algorithms*. Springer, 2001.

[27] Y. Yang and J. Wang, "Design guidelines for routing metrics in multihop wireless networks," in *INFOCOM*, 2008.

[28] H. Zhai and Y. Fang, "Physical carrier sensing and spatial reuse in multirate and multihop wireless ad hoc networks," in *INFOCOM*, 2006.

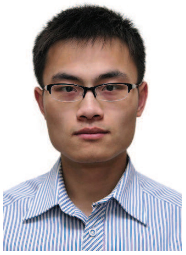
[29] J. Zhang, H. Wu, Q. Zhang, and B. Li, "Joint routing and scheduling in multi-radio multi-channel multi-hop wireless networks," in *BROADNETS*, 2005, pp. 678–687.

[30] S. Zhao, L. Fu, X. Wang, and Q. Zhang, "Fundamental relationship between nodedensity and delay in wireless ad hoc networks with unreliable links," in *MOBICOM*, 2011.

[31] X. Zhao, J. Guo, C. T. Chou, A. Misra, and S. Jha, "A high-throughput routing metric for reliable multicast in multi-rate wireless mesh networks," in *INFOCOM*, 2011.

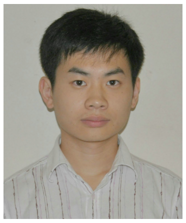
[32] Z. Zhong, J. Wang, S. Nelakuditi, and G.-H. Lu, "On selection of candidates for opportunistic anypath forwarding," *Mobile Computing and Communications Review*, vol. 10, no. 4, pp. 1–2, 2006.

[33] X. Zhou, Z. Zhang, G. Wang, X. Yu, B. Y. Zhao, and H. Zheng, "Practical conflict graphs for dynamic spectrum distribution," in *SIGMETRICS*, 2013.



Tong Meng is a graduate student from the Department of Computer Science and Engineering at Shanghai Jiao Tong University, P. R. China. He received his B.S. in Computer Science from Shanghai Jiao Tong University in 2013. His research interests encompass neighbor discovery, routing in wireless networks and mobile social networks. He is a student member of ACM, CCF, and IEEE.

Fan Wu is an associate professor in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. He received his B.S. in Computer Science from Nanjing University in 2004, and Ph.D. in Computer Science and Engineering from the State University of New York at Buffalo in 2009. He has visited the University of Illinois at Urbana-Champaign (UIUC) as a Post Doc Research Associate. His research interests include wireless networking and mobile computing, algorithmic game theory and its applications, and privacy preservation. He has published more than 80 peer-reviewed papers in leading technical journals and conference proceedings. He is a receipt of China National Natural Science Fund for Outstanding Young Scientists, CCF-Intel Young Faculty Researcher Program Award, CCF-Tencent Rhinoceros bird Open Fund, and Pujiang Scholar. He has served as the chair of CCF YOCSEF Shanghai, on the editorial board of Elsevier Computer Communications, and as the member of technical program committees of more than 40 academic conferences. For more information, please visit http://www.cs.sjtu.edu.cn/~fwu/.



Zheng Yang received the BE degree in the Department of Computer Science from Tsinghua University, Beijing, China, and the PhD degree in the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology. He is currently with the School of Software of Tsinghua University. His research interests include wireless networking, mobile computing, and pervasive computing. He is a member of the IEEE and ACM.

Guihai Chen earned his B.S. degree from Nanjing University in 1984, M.E. degree from Southeast University in 1987, and Ph.D. degree from the University of Hong Kong in 1997. He is a distinguished professor of Shanghai Jiaotong University, China. He had been invited as a visiting professor by many universities including Kyushu Institute of Technology, Japan in 1998, University of Queensland, Australia in 2000, and Wayne State University, USA during September 2001 to August 2003. He has a wide range of research interests with focus on sensor networks, peer-to-peer computing, high-performance computer architecture and combinatorics. He has published more than 200 peer-reviewed papers, and more than 120 of them are in well-archived international journals such as IEEE Transactions on Parallel and Distributed Systems, Journal of Parallel and Distributed Computing, Wireless Networks, The Computer Journal, International Journal of Foundations of Computer Science, and Performance Evaluation, and also in well-known conference proceedings such as HPCA, MOBIHOC, INFOCOM, ICNP, ICPP, IPDPS and ICDCS.

Athanosios V. Vasilakos is currently a Professor with Lulea University of Technology, Sweden. He served or is serving as an Editor or/and Guest Editor for many technical journals, such as the IEEE Transactions on Network and Service Management, IEEE Transactions on cloud computing, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Cybernetics, IEEE Transactions on Information Technology in Biomedicine, ACM Transactions on Autonomous and Adaptive Systems, and IEEE Journal of Selected Areas in Communications. He is also General Chair of the European Alliances for Innovation (www.eai.eu).