Revisiting Routing in Multi-Hop Wireless Networks: Spatial Reusability-Aware Routing

Tong Meng* Fan Wu* Guihai Chen* Athanasios V. Vasilakos†
*Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiao Tong University, China
†Department of Computer and Telecommunication Engineering, University of Western Macedonia, Greece
Email: mengtong@sjtu.edu.cn; {fwu, gchen}@cs.sjtu.edu.cn; vasilako@ath.forthnet.gr

Abstract—**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 throughput gain is up to 2.9x; for anypath routing, the throughput gain is up to 62.7%.**

I. Introduction

Due to limited capacity of wireless communication media and lossy wireless links [25], 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.*, [3], [11], [17], *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.

In this paper, we investigate two kinds of routing protocols, including single-path routing and anypath routing. 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 [4] and EATX [26]) and link transmission time-based metrics (*e.g.*, ETT [5] and EATT [10]). 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. To the best of our knowledge, most of the existing routing protocols do not take spatial reusability of the wireless communication media into account. We 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 2.9x throughput gain in single-path routing and up to 62.7% gain in anypath routing shown by our evaluation results). 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 spatial reusability-aware single-path routing (SASR) algorithm for path selection.
- 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 algorithm and SAAR algorithm in NS-2. Our 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 2.9x with a median of 40% is achieved; for anypath routing, an improvement more than 10% in general and up to 62.7% is realized.

The rest of the paper is organized as below. In Section II, we briefly review related works. In Section III, we introduce the preliminaries. In Section IV, we present our algorithm for reusability-aware single-path routing. In Section V, we present the algorithm for reusability-aware anypath routing. In Section VI, we show the evaluation results. In Section VII, we conclude the paper.

II. Related Work

A. Routing Metrics

For single-path routing, several link-quality aware metrics [1], [4]–[6] were proposed. ETX [4] assigns the link cost

This work was supported in part by the State Key Development Program for Basic Research of China (973 project 2014CB340303 and 2012CB316201), in part by China NSF grant 61472252, 61422208, 61272443, and 61133006, in part by Shanghai Science and Technology fund 12PJ1404900 and 12ZR1414900, and in part by Program for Changjiang Scholars and Innovative Research Team in University (IRT1158, PCSIRT) China. 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.

Fan Wu is the corresponding author.

with its expected number of transmissions to successfully deliver a packet. Incorporating the multi-rate ability, ETT [5] takes the expected transmission time of a link as its cost. What's more, [22] provided some principles for routing metric design. There're also metrics suitable for anypath routing [10], [26]. In [26], the EATX metric was defined to reflect overall transmissions in any-path forwarding. Laufer *et al.* [10] 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 guidelines in [22], such as consistency, ignored the effect of reusability.

B. Routing Protocols

The earliest single-path routing protocols [7], [14], [15] applied dijkstra algorithm for route selection. When it comes to anypath routing, ExOR [2] appeared as a coordination mechanism between forwarders; MORE [3] broke such coordination where all the forwarders worked according to their workload. On that basis, [10] proposed the shortest anypath first (SAF) algorithm to determine the forwarders' priorities; [16] incorporated rate control and dealt with flow control; CodeOR [11] enabled concurrent transmissions of a window of segments; SOAR [20] considered the problem of path divergence and rate limitation to efficiently support multiple flows; SourceSync [17] utilized sender diversity.Because these 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.

C. Other Related Works

Some existing cross-layer approaches jointly consider routing and link scheduling (*e.g.*, [8], [13], [24]). 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 thus can be implemented in a distributed manner.

Last but not least, there are also works aimed at exploiting spatial reusability. Specifically, authors in [9] and [23] studied the effect of carrier sense range for spatial reuse. However, none of these works deal with the problem of route selection.

III. Technical Preliminaries

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. 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. This is commonly considered in the single path routing as Expected Transmission Count metric (ETX) [4]. 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, 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 (*e.g.*, [3], [10]). 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) [3], [26]. Since 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 [3]. 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.

IV. 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 [7]), 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 III, 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

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

[1]However, our approach can be extended to adapt to multiple transmission rates. We leave it to our future work.

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[2], 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 [18], we compute G with measurement-based techniques [12], [19] within $O(|P|)$ time. Then $\mathcal{I}$ must be a collection of maximal independent sets on the conflict graph.

We present an SASR algorithm to calculate the collection $\mathcal{I}$. It aims to find a collection $\mathcal{I}$ that minimize the path cost, which reflects the best possible performance of the path.

A. *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:

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

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 [21], which is NP-hard.

We propose SASR algorithm to solve the above binary program. Algorithm 1 shows the pseudo-code of the 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, conforming to a first-fit manner, 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 such 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 1 is $O(|P|^2)$.

[2]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.

Algorithm 1: SASR 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* $\leftarrow$ *FALSE*;
6   **for** $l \leftarrow 1$ *to* k **do**
7   *reusable* $\leftarrow$ *TRUE*;
8   **foreach** $(i',j') \in \mathcal{I}_l$ **do**
9   **if** $[(i,j),(i',j')] \in E$ **then**
10   *reusable* $\leftarrow$ *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* $\leftarrow$ *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 $\mathcal{I}$;

V. Spatial Reusability-Aware Anypath Routing (SAAR)

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.

A. *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 V-B.)

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

$$\omega_{ij} = \frac{p_{ij}\prod_{k\in F_i \wedge C_k < C_j}(1-p_{ik})}{1-\prod_{k\in F_i}(1-p_{ik})}. \tag{9}$$

Then, we 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:

$$\Omega(i,j) = \begin{cases} 1, & \text{if } j = i; \\ \sum_{k \in F_i} \omega_{ik} \times \Omega(k,j), & \text{if } j \in Q \wedge C_j < C_i. \end{cases} \quad (10)$$

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:

$$\bar{t}_{iF_i} = \Omega(Src,i) \times t_{iF_i}, \quad (11)$$

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

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:

$$c(I) = \max\{\bar{t}_{iF_i} | (i, F_i) \in I\}. \quad (12)$$

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

$$C_{Src} = \sum_{I \in \mathcal{I}} c(I). \quad (13)$$

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

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

B. Algorithm for Min-Cost Anypath Fusion

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 2 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 2 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); calculate the relaying probability matrix Ω_i if a packet is sent from node i (Lines 8-13); and then 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,

Algorithm 2: 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 $\quad C_i \leftarrow +\infty$; $F_i \leftarrow \emptyset$; $\Omega(i,i) \leftarrow 1$;
3 **end**
4 $C_{Dst} \leftarrow 0$; $q \leftarrow Dst$; $Q \leftarrow \{Dst\}$; $\mathcal{I} \leftarrow \{\{Dst\}\}$;
5 **while** $q \neq Src$ **do**
6 $\quad$ **foreach** $(i,q) \in E \wedge i \notin Q$ **do**
7 $\quad\quad F_i \leftarrow F_i \cup \{q\}$; $\Omega_i \leftarrow \Omega$;
8 $\quad\quad$ **foreach** $j \in Q$ **do**
9 $\quad\quad\quad \Omega_i(i,j) \leftarrow 0$;
10 $\quad\quad\quad$ **foreach** $k \in F_i \wedge C_k < C_j$ **do**
11 $\quad\quad\quad\quad \Omega_i(i,j) \leftarrow \Omega_i(i,j) + \omega_{ik} \times \Omega_i(k,j)$;
12 $\quad\quad\quad$ **end**
13 $\quad\quad$ **end**
14 $\quad\quad (C_i, \mathcal{I}_i) \leftarrow$ *CalculateCost*$(i, \mathcal{I}, \Omega_i, F)$;
15 $\quad$ **end**
16 $\quad q \leftarrow \underset{i \in N \setminus Q}{argmin}(C_i)$; $Q \leftarrow Q \cup \{q\}$;
17 $\quad \Omega \leftarrow \Omega_q$; $\mathcal{I} \leftarrow \mathcal{I}_q$;
18 **end**
19 **return** Q, C, and F;

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 3. 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 2 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 run time of function *CalculateCost*() is $O(|N|)$. Therefore, the run time of the SAAR algorithm is $O(|N|^3)$.

Algorithm 3: Function *CalculateCost*()

Input: A node i, a collection of non-interfering sets $\mathcal{I}$, a profile of relay probabilities Ω, 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 $\mathcal{I}$;

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

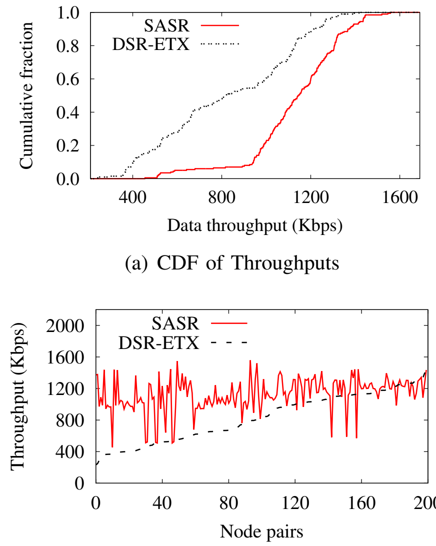
TABLE I
SIMULATION PARAMETERS SETUP

VI. EVALUATION

We evaluated the performance of our SASR and SAAR algorithms in NS-2, and compared them with ETX-based DSR [4] (denoted by DSR-ETX) and the shortest anypath first (SAF) algorithm [10]. Table I lists the parameters in our simulation. To be detailed, we randomly picked 200 source-destination pairs for single-path and anypath routing, respectively, 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 100 runs, each of which lasted 10 minutes.

A. Performance of SASR Algorithm

Fig. 1(a) shows the cumulative distribution of throughputs achieved by SASR and DSR-ETX. SASR algorithm achieve a median throughput gain of 40%. We present detailed pairwise comparisons in Fig. 1(b). All the 200 simulated node pairs are sorted by their throughputs under DSR-ETX in a non-decreasing order. We observe that SASR shows clear throughput improvements, especially when DSR-ETX does not perform well (*i.e.*, having a throughput less than 1000Kbps). What's more, except some node pairs suffering from hidden terminals under DSR-ETX, more than 20% node pairs have doubled throughputs, and the throughput gain achieved by SASR can reach 2.9x. Note that the throughput gains tend to be higher for those node pairs which perform bad under DSR-ETX, because these pairs correspond to paths with larger hop-counts, which provide more opportunities for spatial reuse.



(b) Pairwise Comparisons

Fig. 1. Simulation Results of SASR algorithm

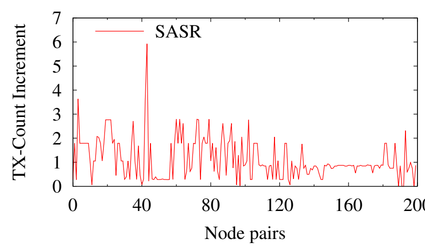
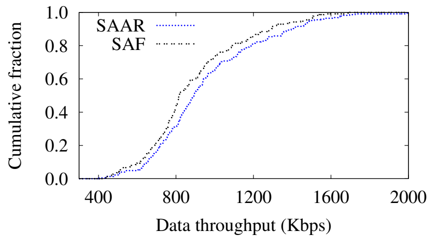


Fig. 2. Overall Transmission Count Increments Induced by SASR Algorithms

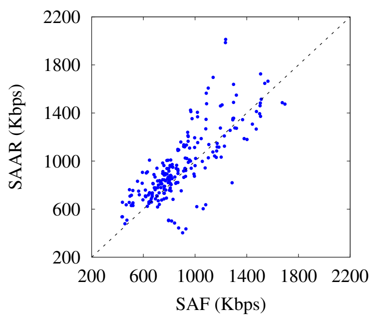
However, owing to cost fusion, SASR algorithm may select longer path than DSR-ETX. Fig. 2 shows that the increment in transmission counts is not much and totally acceptable compared with the throughput gains, *i.e.*, more than 80% of the node pairs only need no more than 2 additional overall transmissions compared with DSR-ETX.

B. Performance of SAAR Algorithm

Fig. 3(a) shows the CDF of throughputs achieved by SAAR and SAF. Although the throughput gains achieved by SAAR are not as great as those of SASR in single-path routing, there is still a 9.3% improvement in the median case. Such results are owing to the more comprehensive interfering situations in anypath routing. Therefore, even considering more nodes participating in packet forwarding in anypath routing and



(a) CDF of Throughputs



(b) Scatters of Pairwise Throughputs

Fig. 3. Simulation Results of SAAR algorithm

consequently leading to more opportunities of concurrent transmissions among hyperlinks, it is non-trivial to achieve as great improvement as in single-path routing. Besides, a scatter plot is used in Fig. 3(b) to directly present the different performance of SAAR and SAF. We are glad to find that most of the simulated node pairs display significant gains in throughput. Some of them can have an improvement up to 62.7%, while the majority realize a gain of more than 10%.

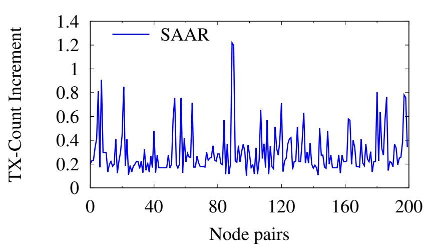


Fig. 4. Increment of Number of Transmissions when Exploiting Reusability.

Fig. 4 shows the amount of additional transmissions required by SAAR, compared with SAF. The figure shows that the increments of transmission count do not exceed one, except for only one source-destination pair.

VII. Conclusion

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. Evaluation results show that SASR achieves a throughput gain of as high as 2.9x, while for SAAR, the maximum gain can reach 62.7%. Meanwhile, they only require acceptable additional transmission overhead.

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

[4] 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.

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

[6] 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.

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

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

[9] 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.

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

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

[12] 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.

[13] 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.

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

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

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

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

[18] 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.

[19] 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.

[20] 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.

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

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

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

[24] 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.

[25] 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.

[26] 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.