Multirate Anypath Routing in Wireless Mesh Networks

Rafael Laufer
Computer Science Department
University of California at Los Angeles

Henri Dubois-Ferrière
Riverbed Technology, Inc.
Lausanne, Switzerland

Leonard Kleinrock
Computer Science Department
University of California at Los Angeles

Abstract—**In this paper, we present a new routing paradigm that generalizes opportunistic routing in wireless mesh networks. In multirate anypath routing, each node uses both a set of next hops and a selected transmission rate to reach a destination. Using this rate, a packet is broadcast to the nodes in the set and one of them forwards the packet on to the destination. To date, there is no theory capable of jointly optimizing both the set of next hops and the transmission rate used by each node. We bridge this gap by introducing a polynomial-time algorithm to this problem and provide the proof of its optimality. The proposed algorithm runs in the same running time as regular shortest-path algorithms and is therefore suitable for deployment in link-state routing protocols. We conducted experiments in a 802.11b testbed network, and our results show that multirate anypath routing performs on average 80% and up to 6.4 times better than anypath routing with a fixed rate of 11 Mbps. If the rate is fixed at 1 Mbps instead, performance improves by up to one order of magnitude.**

I. Introduction

The high loss rate and dynamic quality of links make routing in wireless mesh networks extremely challenging [1]. Anypath routing[1] has been recently proposed as a way to circumvent these shortcomings by using multiple next hops for each destination [3]–[6]. Each packet is broadcast to a forwarding set composed of several neighbors, and the packet must be retransmitted only if none of the neighbors in the set receive it. Therefore, while the link to a given neighbor is down or performing poorly, another nearby neighbor may receive the packet and forward it on. This is in contrast to single-path routing where only one neighbor is assigned as the next hop for each destination. In this case, if the link to this neighbor is not performing well, a packet may be lost even though other neighbors may have overheard it.

Existing work on anypath routing has focused on wireless networks that use a single transmission rate. This approach, albeit straightforward, presents two major drawbacks. First, using a single rate over the entire network underutilizes available bandwidth resources. Some links may perform well at a higher rate, while others may only work at a lower rate. Secondly and most importantly, the network may become disconnected at a higher bit rate. We provide experimental measurements from a 802.11b testbed which show that this phenomenon is not uncommon in practice. The key problem is that higher transmission rates have a shorter radio range, which reduces network density and connectivity. As the bit rate increases, links becomes lossier and the network eventually gets disconnected. Therefore, in order to guarantee connectivity, single-rate anypath routing must be limited to low rates.

In *multirate anypath routing*, these problems do not exist; however, we face different challenges. First, we must find not only the forwarding set, but also the transmission rate at each hop that jointly minimizes its cost to a destination. Secondly, loss probabilities usually increase with higher transmission rates, so a higher bit rate does not always improve throughput. Finally, higher rates have a shorter radio range and therefore we have a different connectivity graph for each rate. Lower rates have more neighbors available for inclusion in the forwarding set (i.e., more spatial diversity) and less hops between nodes. Higher rates have less neighbors available for the forwarding set (i.e., less spatial diversity) and longer routes. Finding the optimal operation point in this tradeoff is the focus of this paper.

We thus address the problem of finding both a forwarding set and a transmission rate for every node, such that the overall cost of every node to a particular destination is minimized. We call this the *shortest multirate anypath problem*. To our knowledge, this is still an open problem [3], [4], [7] and we believe our algorithm is the first practical solution for it.

We introduce a polynomial-time algorithm to the shortest multirate anypath problem and present a proof of its optimality. Our solution generalizes Dijkstra's algorithm for the multirate anypath case and is applicable to link-state routing protocols. One would expect that the running time of such an algorithm to be exponential. However, we show that it has the *same* polynomial time as the corresponding shortest-path algorithm, being suitable for implementation at current wireless routers. We also generalize the expected transmission time (ETT) routing metric [8] for multirate anypath routing.

For the performance evaluation, we conducted experiments in an 18-node 802.11b wireless testbed of embedded Linux devices. Our results reveal that the network becomes partially disconnected when we fix the transmission rate of every node at 2, 5.5, and 11 Mpbs. A single-rate routing scheme therefore performs poorly in this case, since 1 Mbps is the only rate at which the network is fully connected. We show that multirate anypath routing improves the end-to-end expected transmission time by 80% on average and by up to 6.4 times compared to single-rate anypath routing at 11 Mbps, while still maintaining full network connectivity. The performance is even higher over the single-rate case at 1 Mbps, with an average gain of a factor of 5.4 and a maximum gain of a factor of 11.3.

The remainder of the paper is organized as follows. Section II reviews the basic theory of anypath routing, our network model and assumptions. In Section III, we introduce multirate anypath routing and the proposed routing metric. Section IV presents the multirate anypath algorithm and proves its optimality. Section V reveals our experimental results, showing the benefits of multirate over single-rate anypath routing. Section VI presents the related work in anypath routing. Finally, conclusions are presented in Section VII.

[1] We use the term anypath rather than opportunistic routing, since opportunistic routing is an overloaded term also used for opportunistic contacts [2].

II. Anypath Routing

In this section we review the anypath routing theory introduced by Zhong *et al.* [5] and Dubois-Ferrière *et al.* [6]. The main contributions of the paper are presented later in Sections III and IV.

A. Overview

In classic wireless network routing, each node forwards a packet to a single next hop. As a result, if the transmission to that next hop fails, the node needs to retransmit the packet even though other neighbors may have overheard it. In contrast, in anypath routing, each node broadcasts a packet to *multiple* next hops simultaneously. Therefore, if the transmission to one neighbor fails, an alternative neighbor who received the packet can forward it on. We define this set of multiple next hops as the *forwarding set* and we usually use J to represent it throughout the paper. A different forwarding set is used to reach each destination, in the same way a distinct next hop is used for each destination in classic routing.

When a packet is broadcast to the forwarding set, more than one node may receive the same packet. To avoid unnecessary duplicate forwarding, only one of these nodes should forward the packet on. For this purpose, each node in the set has a priority in relaying the received packet. A node only forwards a packet if all higher priority nodes in the set failed to do so. Higher priorities are assigned to nodes with shorter distances to the destination. As a result, if the node with the shortest distance in the forwarding set successfully received the packet, it forwards the packet to the destination while others suppress their transmission. Otherwise, the node with the second shortest distance forwards the packet, and so on. A reliable anycast scheme [9] is necessary to enforce this relay priority. We talk more about this in Section II-B. The source keeps rebroadcasting the packet until someone in the forwarding set receives it or a threshold is reached. Once a neighbor in the set receives the packet, this neighbor repeats the same procedure until the packet is delivered to the destination.

Since we now use a set of next hops to forward packets, every two nodes are connected through a mesh composed of the union of multiple paths. Figure 1 depicts this scenario where each node uses a set of neighbors to forward packets. The forwarding sets are defined by the multiple bold arrows leaving each node. We define this union of paths between two nodes as an *anypath*. In the figure, the anypath shown in bold is composed by the union of 11 different paths between a source s and a destination d. Depending on the choice of each forwarding set, different paths are included in or excluded from the anypath. At every hop, only a single node of the set forwards the packet on. Consequently, every packet from s traverses only one of the available paths to reach d. We show a path possibly taken by a packet using a dashed line. Succeeding packets, however, may take completely different paths; hence the name anypath. The path taken is determined on-the-fly, depending on which nodes of the forwarding set successfully receive the packet at each hop.

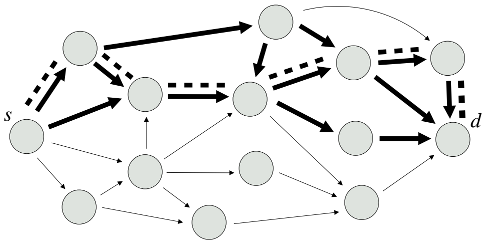


Figure 1. An anypath connecting nodes s and d is shown in bold arrows. The anypath is composed of the union of 11 paths between the two nodes. Every packet sent from s traverses one of these paths to reach d, such as the path shown with a dashed line. Different packets may traverse different paths, depending on who receives the packet at each hop; hence the name anypath.

B. System Model and Assumptions

In order to support the point-to-multipoint links used in anypath routing, we model the wireless mesh network as a hypergraph. A hypergraph $\mathcal{G} = (V, \mathcal{E})$ is composed of a set V of vertices or nodes and a set $\mathcal{E}$ of hyperedges or hyperlinks. A hyperlink is an ordered pair (i, J), where $i \in V$ is a node and J is a nonempty subset of V composed of neighbors of i. For each hyperlink $(i, J) \in \mathcal{E}$, we have a delivery probability p_{iJ} and a distance d_{iJ}. If the set J has a single element j, then we just use j instead of J in our notation. In this case, p_{ij} and d_{ij} denote the link delivery probability and distance, respectively.

The hyperlink delivery probability p_{iJ} is defined as the probability that a packet transmitted from i is successfully received by at least one of the nodes in J. One would expect that the receipt of a packet at each neighbor is correlated due to noise and interference. However, we conducted experiments which suggest that the loss of a packet at different receivers occur independently in practice [10], which is consistent with other studies [11]. With the assumption of independent losses, the probability p_{iJ} is

$$p_{iJ} = 1 - \prod_{j \in J} (1 - p_{ij}) . \tag{1}$$

Previously proposed MAC protocols have been designed to guarantee the relay priority among the nodes in the forwarding set [4], [9]. Such protocols can use different strategies for this purpose, such as time-slotted access, prioritized contention and frame overhearing. Reliable anycast is an active area of research [9] and we assume that such mechanism is in place to make sure that the relaying priority is respected. The details of the MAC, however, are abstracted from the routing layer. Practical routing protocols only incorporate the delivery probabilities into the routing metric in order to abstract from the MAC details [8], [12] and we take the same approach. The only MAC aspect that is important is the effectiveness of the relaying node selection. As long as the relaying node is actually the one with the shortest distance to the destination, there should be no significant impact on the routing performance.

C. Anypath Cost

We are interested in calculating the anypath cost from a node i to a given destination via a forwarding set J. The anypath cost D_i is defined as $D_i = d_{iJ} + D_J$, which is composed of the hyperlink cost d_{iJ} from i to J and the remaining-anypath cost D_J from J to the destination.

The hyperlink cost d_{iJ} depends on the routing metric used. Most previous works on anypath routing have adopted the expected number of anypath transmissions (EATX) as the routing metric [3], [5], [6]. The EATX is a generalization of the unidirectional ETX metric [12], which is defined as $d_{ij} = 1/p_{ij}$. The distance d_{ij} for ETX represents the expected number of transmissions necessary for a packet sent by i to be successfully received by j. For EATX, the distance d_{iJ} is defined as $d_{iJ} = 1/p_{iJ}$, which is the average number of transmissions necessary for at least one node in J to correctly receive the transmitted packet.

The remaining-anypath cost D_J is intuitively defined as a *weighted average* of the distances of the nodes in the forwarding set as

$$D_J = \sum_{j \in J} w_j D_j, \text{ with } \sum_{j \in J} w_j = 1, \tag{2}$$

where the weight w_j in (2) is the probability of node j being the relaying node. For example, let $J = \{1, 2, \ldots, n\}$ with distances $D_1 \leq D_2 \leq \ldots \leq D_n$. We refer to the probability p_{ij} simply by p_j for convenience. Node j will be the relaying node only when it receives the packet and none of the nodes closer to the destination receives it, which happens with probability $p_j(1 - p_{j-1})(1 - p_{j-2}) \ldots (1 - p_1)$. The weight w_j is then defined as

$$w_j = \frac{p_j \prod_{k=1}^{j-1} (1 - p_k)}{1 - \prod_{j \in J} (1 - p_j)}, \tag{3}$$

with the denominator being the normalizing constant.

As an example, consider the network depicted in Figure 2. The distance via J in Figure 2(a) is calculated as

$$\begin{aligned} D_i &= d_{iJ} + D_J \\ &= \frac{1}{1 - (1 - 1/4)(1 - 1/5)} + \frac{(1/4)3 + (3/4)(1/5)3}{1 - (1 - 1/4)(1 - 1/5)} \\ &= 2.5 + 3.0 = 5.5. \end{aligned} \tag{4}$$

One would expect that adding an extra node to the forwarding set is always beneficial because it increases the number of possible paths a packet can take. However, this is not always true, as shown in Figure 2(b). The anypath distance via $J' = J \cup \{j\}$ is $D_i = d_{iJ'} + D_{J'} = 1.8 + 4.6 = 6.4$. On one hand, using J' instead of J reduces the hyperlink cost, that is, $d_{iJ'} \leq d_{iJ}$. On the other hand, the extra node increases the remaining anypath cost, that is, $D_{J'} \geq D_J$. If the increase $D_{J'} - D_J$ is higher than the decrease $d_{iJ} - d_{iJ'}$, adding this extra node is not worthy since the total cost to reach the destination increases. The intuition here is that when node j is the only one in J' that received the packet, it is cheaper to retransmit the packet to one of the two nodes in J and take a shorter path from there than to take the long path via node j.

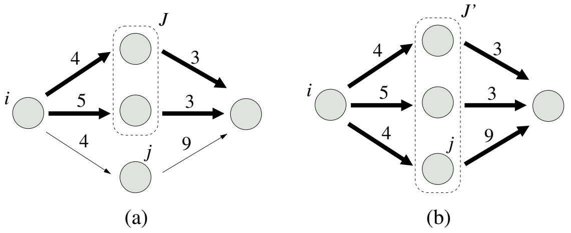


Figure 2. An anypath cost calculation example. The weight of each link is the expected number of transmissions (ETX), which is the inverse of the link delivery probability. The anypath cost in (a) is lower than the cost in (b).

Once the cost of an anypath is defined, it is of interest to find the anypath with the lowest cost to the destination, that is, the shortest anypath. This is called the *shortest-anypath problem* [6]. Interestingly enough, the shortest anypath will always have an equal or lower cost than the shortest single path. This is a direct consequence of the definition of an anypath as a set of paths. Among all possible anypaths between two nodes, we also have the anypath composed only of the path with the shortest ETX. Therefore, if we are to choose the shortest anypath among all these possibilities, we know for sure that its cost can never be higher than the cost of the shortest single path.

III. Multirate Anypath Routing

Previous work on anypath routing focused on a single bit rate [3]–[6]. Such an assumption, however, considerably underutilizes available bandwidth resources. Some hyperlinks may be able to sustain a higher transmission rate, while others may only work at a lower rate. To date, the problem of how to select the transmission rate for anypath routing is still open [7]. We provide a solution to this problem and incorporate the multirate capability inherent in IEEE 802.11 networks into anypath routing. In this case, besides selecting a set of next hops to forward packets, a node must also select one among multiple transmission rates. For each destination, a node then keeps both a forwarding set and a transmission rate used to reach this set. As a result, every two nodes will be connected through a mesh composed of the union of multiple paths, with each node transmitting at a selected rate. Figure 3 depicts the scenario where nodes use a selected bit rate to forward packets to a set of neighbors. We define this union of paths between two nodes, with each node using a potentially different bit rate as a *multirate anypath*. In the figure, assume that a packet is sent from s to d over the multirate anypath. Only one of the available paths is traversed depending on which nodes successfully receive the packet at each hop. We show a path possibly taken by the packet using dashed lines. We use different dash lengths to represent the different transmission rates used by each node. A shorter dash represents a shorter time to send a packet, hence a higher transmission rate. Succeeding packets may take completely different paths with other transmission rates along its way.

In order to support multirate, we must extend the system model in Section II-B. Let R be the set of available bit rates that nodes can use to transmit their packets. For each hyperlink $(i, J) \in \mathcal{E}$, we now have a delivery probability $p_{iJ}^{(r)}$ and a distance $d_{iJ}^{(r)}$ associated with each transmission rate $r \in R$.

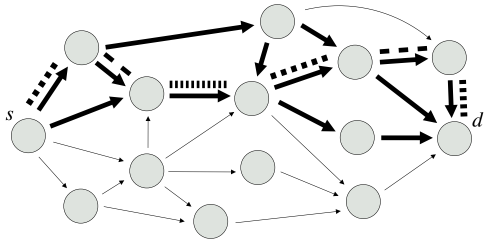


Figure 3. A multirate anypath connecting nodes s and d is shown in bold arrows. Every packet sent from s traverses a path to reach d, such as the path shown with dashed lines. Different dash lengths represent the different bit rates used by each node, with a shorter dash for higher rates.

In real wireless networks, we usually have different delivery probabilities and distances for each transmission rate, which justifies this model extension.

The EATX metric described in Section II-C was originally designed considering that nodes transmit at a single bit rate. To account for multiple bit rates, we introduce the expected anypath transmission time (EATT) metric. For EATT, the hyperlink distance $d_{iJ}^{(r)}$ for each rate $r \in R$ is defined as

$$d_{iJ}^{(r)} = \frac{1}{p_{iJ}^{(r)}} \times \frac{s}{r}, \tag{5}$$

where $p_{iJ}^{(r)}$ is the hyperlink delivery probability defined in (1), s is the maximum packet size, and r is the bit rate. The distance $d_{iJ}^{(r)}$ is basically the time it takes to transmit a packet of size s at a bit rate r over a lossy hyperlink with delivery probability $p_{iJ}^{(r)}$. The EATT metric is a direct generalization of the expected transmission time (ETT) metric [8] commonly used in single-path wireless routing. Note that for each bit rate $r \in R$, we have a different delivery probability $p_{iJ}^{(r)}$, which usually decreases for higher rates. This behavior imposes a tradeoff; a higher bit rate decreases the time of a single packet transmission (i.e., s/r decreases), but it usually increases the number of transmissions required for a packet to be successfully received (i.e., $1/p_{iJ}^{(r)}$ increases).

The remaining-anypath cost $D_J^{(r)}$ now also depends on the transmission rate, since the delivery probabilities change for each rate. Since both the hyperlink distance and the remaining anypath cost depend on the bit rate, node i has a different anypath cost $D_i^{(r)} = d_{iJ}^{(r)} + D_J^{(r)}$ for each forwarding set J and for each transmission rate $r \in R$.

We address the problem of finding both the forwarding set and the transmission rate that minimize the overall cost to reach a particular destination. We call this the *shortest multirate anypath problem*, which generalizes the shortest-anypath problem [6] for the multirate scenario. Interestingly, the shortest multirate anypath will always have equal or lower cost than the shortest single path. Among all possible multirate anypaths between two nodes, we also have the single path with the shortest ETT. As a result, the cost of the shortest multirate anypath can never be higher than the cost of the shortest path. Likewise, due to the same argument, the shortest multirate anypath will also have equal or lower cost than any shortest anypath using a single rate.

IV. Finding the Shortest Multirate Anypath

In this section we introduce the proposed shortest-anypath algorithms. In Section IV-A, we present the Shortest Anypath First (SAF) algorithm used in a single-rate network with the EATX metric. Our SAF algorithm, while derived independently, is similar to the single-rate algorithm proposed by Chachulski [3]. Our main contribution is to provide a proof of its optimality. We also use the single-rate case as the basis for the multirate generalization introduced in Section IV-B. Surprisingly, the Shortest Multirate Anypath First (SMAF) algorithm has the *same* running time as a shortest single-path algorithm for multirate, being suitable for deployment in link-state routing protocols. We only show the proof of optimality of the SMAF algorithm, since by definition this also implies the optimality of the SAF algorithm.

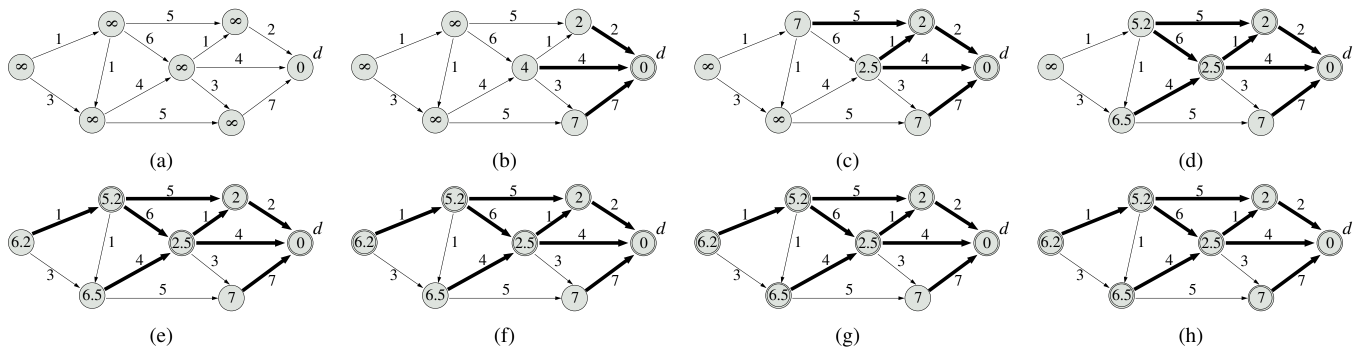
A. The Single-Rate Case

We now present the Shortest Anypath First algorithm used in the simpler single-rate scenario. Given a graph $G = (V, E)$, the algorithm calculates the shortest anypaths from all nodes to a destination d. For every node $i \in V$ we keep an estimate D_i, which is an upper-bound on the distance of the shortest anypath from i to d. In addition, we also keep a forwarding set F_i for every node, which stores the set of nodes used as the next hops to reach d. Finally, we keep two data structures, namely S and Q. The S set stores the set of nodes for which we already have a shortest anypath defined. We store each node $i \in V - S$ for which we still do not have a shortest anypath in a priority queue Q keyed by their D_i values.

```
SHORTEST-ANYPATH-FIRST($G, d$)
for each node $i$ in $V$
    do $D_i \leftarrow \infty$
       $F_i \leftarrow \emptyset$
$D_d \leftarrow 0$
$S \leftarrow \emptyset$
$Q \leftarrow V$
while $Q \neq \emptyset$
    do $j \leftarrow$ EXTRACT-MIN($Q$)
       $S \leftarrow S \cup \{j\}$
       for each incoming edge $(i, j)$ in $E$
           do $J \leftarrow F_i \cup \{j\}$
              if $D_i > D_j$
                  then $D_i \leftarrow d_{iJ} + D_J$
                       $F_i \leftarrow J$
```

As in the shortest-path algorithm, the Shortest Anypath First algorithm is composed of $|V|$ rounds, dictated by the number of elements initially in Q. At each round, the Extract-Min procedure extracts the node with the minimum distance to the destination from Q. Let this node be j. At this point, j is settled and inserted into S, since the shortest anypath from j to the destination is now known. For each incoming edge $(i, j) \in E$, we check if the distance D_i is larger than the distance D_j. If that is the case, then node j is added to the forwarding set F_i and the distance D_i is updated.

Figure 4 shows the execution of Shortest Anypath First algorithm using the EATX metric. We see in Figure 4(a) the graph right after the initialization. Figures 4(b)–4(h) show



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4. Execution of the Shortest Anypath First (SAF) algorithm from every node to d. The weight of each link is the expected number of transmissions (ETX), which is the inverse of the link delivery probability. (a) The situation just after the initialization. (b)–(h) The situation after each successive iteration of the algorithm. Part (h) shows the situation after the last node is settled.

each iteration of the algorithm. At each step, the value inside a node i presents the distance D_i from that node to the destination d and the arrows in boldface present the shortest anypath to d. Nodes with two circles are the settled nodes in S. The graph in Figure 4(h) shows the result of SAF algorithm right after settling the last node.

The running time of the Shortest Anypath First algorithm depends on how Q is implemented. Assuming that we have a Fibonacci heap, the cost of each of the $|V|$ EXTRACT-MIN operations in line 8 takes $O(\log V)$, with a total of $O(V \log V)$ aggregated time. The running time to calculate both d_{iJ} and D_J in line 13 depends on the size of J; however, if we store additional state, it can be reduced to a constant time [10]. The **for** loop of lines 10–13 takes $O(E)$ aggregated time and as a result the total complexity of the algorithm is $O(V \log V + E)$, which is the same complexity of Dijkstra's algorithm.

B. The Multirate Case

We now generalize the SAF algorithm to support multiple transmission rates, introducing the Shortest Multirate Anypath First (SMAF) algorithm. For each node $i \in V$, we now keep a different distance estimate $D_i^{(r)}$ for every rate $r \in R$. The estimate $D_i^{(r)}$ is an upper-bound on the distance of the shortest anypath from i to d using transmission rate r. In addition, we also keep its corresponding forwarding set $F_i^{(r)}$, which stores the set of next hops used for i to reach d using r. We use D_i and F_i without the indicated rates to store the minimum distance estimate among all rates and its corresponding forwarding set, respectively. We also keep a transmission rate T_i for every node, which stores the rate used to reach d.

The key idea of the SMAF algorithm is that each node $i \in V$ has an independent distance estimate $D_i^{(r)}$ for each rate $r \in R$ and we keep the minimum of these estimates as the node distance D_i. At each round of the **while** loop, the node with the minimum distance from Q is settled. Let this node be j. For each incoming edge $(i, j) \in E$, we check for every rate $r \in R$ if the distance $D_i^{(r)}$ is larger than the distance D_j of the node just settled. If that is the case, then node j is added to the forwarding set $F_i^{(r)}$ of that specific rate and distance $D_i^{(r)}$ is updated accordingly. If the new distance $D_i^{(r)}$ is shorter than the node distance D_i, we update the node distance D_i as well as the forwarding set F_i and transmission rate T_i to reflect the new minimum.

```
SHORTEST-MULTIRATE-ANYPATH-FIRST($G, d$)
 1  for each node $i$ in $V$
 2      do $D_i \leftarrow \infty$
 3         $F_i \leftarrow \emptyset$
 4         $T_i \leftarrow$ NIL
 5         for each rate $r$ in $R$
 6             do $D_i^{(r)} \leftarrow \infty$
 7                $F_i^{(r)} \leftarrow \emptyset$
 8  $D_d \leftarrow 0$
 9  $S \leftarrow \emptyset$
10  $Q \leftarrow V$
11  while $Q \neq \emptyset$
12      do $j \leftarrow$ EXTRACT-MIN($Q$)
13         $S \leftarrow S \cup \{j\}$
14         for each incoming edge $(i, j)$ in $E$
15             do for each rate $r$ in $R$
16                    do $J \leftarrow F_i^{(r)} \cup \{j\}$
17                       if $D_i^{(r)} > D_j$
18                          then $D_i^{(r)} \leftarrow d_{iJ}^{(r)} + D_J^{(r)}$
19                               $F_i^{(r)} \leftarrow J$
20                               if $D_i > D_i^{(r)}$
21                                  then $D_i \leftarrow D_i^{(r)}$
22                                       $F_i \leftarrow F_i^{(r)}$
23                                       $T_i \leftarrow r$
```

The running time of the Shortest Multirate Anypath First algorithm also depends on the implementation of Q. The initialization in lines 1–10 takes $O(VR)$ time. Assuming that we have a Fibonacci heap, the EXTRACT-MIN operations in line 12 take a total of $O(V \log V)$ aggregated time. We assume that the distance calculation of $d_{iJ}^{(r)}$ and $D_J^{(r)}$ in line 18 is optimized to take a constant time [10]. As a result, the **for** loop in lines 15–23 takes $O(ER)$ aggregated time. The total running time is therefore $O(V \log V + (E + V)R)$, which is $O(V \log V + ER)$ if all nodes are able to reach the destination. This is the same running time of the shortest single-path algorithm for multiple rates. Compared to the SAF algorithm, the SMAF algorithm allows nodes to take advantage of their multiple transmission rates at the cost of just a small increase in the running time.

In order to prove the optimality of the algorithm, we first introduce five lemmas that show a few properties of multirate anypath routing. We use $\delta_i^{(r)}$ as the distance of the shortest multirate anypath from a node i to the destination d, when i transmits at a fixed rate $r \in R$. Likewise, $\phi_i^{(r)}$ represents the corresponding forwarding set used in this multirate anypath. We use δ_i without the indicated rate to represent the distance of the shortest multirate anypath from i to d via the optimal forwarding set ϕ_i and optimal transmission rate $\rho \in R$. That is, $\delta_i = \min_{r \in R} \delta_i^{(r)}$, $\rho = \arg\min_{r \in R} \delta_i^{(r)}$, and $\phi_i = \phi_i^{(\rho)}$. We use D_i as the distance of a particular multirate anypath from i to d, but not necessarily the shortest one. The proof for each of these lemmas is available in [10].

Lemma 1: For a fixed transmission rate, let D_i be the distance of a node i via forwarding set J and let D'_i be the distance via forwarding set $J' = J \cup \{n\}$, where $D_n \geq D_j$ for every node $j \in J$. We have $D'_i \leq D_i$ if and only if $D_i \geq D_n$.

We use Lemma 1 for the comparisons in line 12 of the SAF algorithm and in line 17 of the SMAF algorithm. By this lemma, if the distance D_i via J is larger than the distance D_n of a neighbor node n, with $D_n \geq D_j$ for all $j \in J$, then the distance D'_i via $J' = J \cup \{n\}$ is always smaller than D_i. That is, it is always beneficial to include node n in the forwarding set in order to obtain a shorter distance to the destination.

Lemma 2: The shortest distance δ_i of a node i is always larger than or equal to the shortest distance δ_j of any node j in the optimal forwarding set ϕ_i. That is, we have $\delta_i \geq \delta_j$ for all $j \in \phi_i$.

Lemma 2 guarantees that if a node i uses another node j in its optimal forwarding set ϕ_i, then distance δ_i can never be smaller than δ_j. This is equivalent to the restriction that all weights in the graph must be nonnegative in Dijkstra's algorithm.

Lemma 3: For any transmission rate, if a node i uses a node n in its optimal forwarding set ϕ_i and $\delta_i = \delta_n$, we can safely remove n from ϕ_i without changing δ_i. The link (i, n) is said to be "redundant."

By Lemma 3, if the distances $\delta_i = \delta_n$ of two nodes i and n are the same, then the distance δ_i via forwarding set ϕ_i is the same as the distance via forwarding set $\phi_i - \{n\}$. That is, the distance of node i does not change if it uses n in its forwarding set or not.

Lemma 4: If the shortest distances from the neighbors of a node i to a given destination are $\delta_1 \leq \delta_2 \leq \ldots \leq \delta_n$, then $\phi_i^{(r)}$ is always of the form $\phi_i^{(r)} = \{1, 2, \ldots, k\}$, for some $k \in \{1, 2, \ldots, n\}$.

According to Lemma 4, the best forwarding set $\phi_i^{(r)}$ for transmission rate $r \in R$ is a subset of neighbors with the shortest distances to the destination. That is, given a set of neighbors with distances $\delta_1 \leq \delta_2 \leq \ldots \leq \delta_n$, the best forwarding set $\phi_i^{(r)}$ when using rate $r \in R$ is always one of $\{1\}$, $\{1, 2\}$, $\{1, 2, 3\}, \ldots, \{1, 2, \ldots, n\}$. As a result, forwarding sets with gaps between the neighbors, such as $\{2, 3\}$ or $\{1, 4\}$, can never yield the shortest distance to the destination. This property is the key factor that allows us to reduce the complexity of the proposed algorithms from exponential to polynomial time. For n neighbors, we do not have to test every one of the $2^n - 1$ possible forwarding sets. Instead, we only need to check at most n forwarding sets.

Lemma 5: For a given transmission rate $r \in R$, assume that $\phi_i^{(r)} = \{1, 2, \ldots, n\}$ with distances $\delta_1 \leq \delta_2 \leq \ldots \leq \delta_n$. If D_i^j is the distance from node i using transmission rate r via forwarding set $\{1, 2, \ldots, j\}$, for $1 \leq j \leq n$, then we always have $D_i^1 \geq D_i^2 \geq \ldots \geq D_i^n = \delta_i^{(r)}$.

Lemma 5 explains another important property necessary for the SMAF algorithm to converge. Assuming now that the best forwarding set $\phi_i^{(r)}$ for transmission rate $r \in R$ is defined as $\phi_i^{(r)} = \{1, 2, \ldots, n\}$ with distances $\delta_1 \leq \delta_2 \leq \ldots \leq \delta_n$, the distance D_i monotonically decreases as we use each of the forwarding sets $\{1\}, \{1, 2\}, \{1, 2, 3\}, \ldots, \{1, 2, \ldots, j\}$.

We now present the proof of optimality of the algorithm.

Theorem 1: Optimality of the algorithm.

Let $G = (V, E)$ be a weighted, directed, graph and let d be the destination. After running the Shortest Multirate Anypath First algorithm on G, we have $D_i = \delta_i$ for all nodes $i \in V$.

Proof: This proof is similar to the proof of Dijkstra's algorithm [13]. We show that for each node $s \in V$, we have $D_s = \delta_s$ at the time s is added to S.

For the purpose of contradiction, let s be the first node added to S for which $D_s \neq \delta_s$. We must have $s \neq d$ because d is the first node added to S and $D_d = \delta_d = 0$ at that time. Just before adding s to S, we also have that S is not empty, since $s \neq d$ and S must contain at least d. We assume that there must be a multirate anypath from s to d, otherwise $D_s = \delta_s = \infty$, which contradicts our initial assumption that $D_s \neq \delta_s$. If there is at least one multirate anypath, there is a shortest multirate anypath α from s to d. Let us consider a cut $(V - S, S)$ of α, such that we have $s \in V - S$ and $d \in S$, as shown in Figure 5. Let the set J be composed of nodes in $V - S$ that have an outgoing link to a node in S. Likewise, let the set K be composed of nodes in S that have an incoming link from a node in $V - S$.

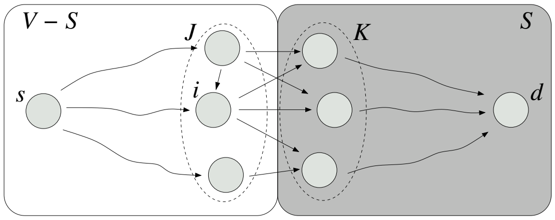


Figure 5. The shortest multirate anypath α from s to d. Set S must be nonempty before node s is inserted into it, since it must contain at least d. We consider a cut $(V - S, S)$ of α, such that we have $s \in V - S$ and $d \in S$. Nodes s and d are distinct but we may have no hyperlinks between s and J, such that $J = \{s\}$, and also between K and d, such that $K = \{d\}$.

Without loss of generality, assume that node $i \in J$ has the shortest distance to d among all nodes in $V - S$. That is, $\delta_i \leq \delta_j$ for all $j \in V - S$. We claim that every edge leaving node i must necessarily cross the cut $(V - S, S)$. Thus, for every edge (i, j) leaving node i, we must have $j \in S$. To prove this claim, let us assume that node i has an edge (i, j) to another node $j \in V - S$. By Lemma 2, we know that in this case we must have $\delta_i \geq \delta_j$. However, since we assumed that node i has the shortest distance in $V - S$, then $\delta_i \leq \delta_j$

and such an edge could only exist if $\delta_i = \delta_j$. By Lemma 3, we know that if $\delta_i = \delta_j$ then the link (i, j) is redundant and we can safely remove it from the multirate anypath without changing its distance. As a result, for every edge (i, j) we must have $j \in S$. Figure 5 shows this situation where node i only has links to nodes in S.

Additionally, we claim that the nodes in S were settled in ascending order of distance. That is, if $\delta_j < \delta_k$ then node j was settled before node k. Since node i has the shortest distance to d among all nodes in $V - S$, settling s before i implies that s is settled "out of order." For the purpose of contradiction, let s be the first node settled out of order. This is an assumption which is independent from the initial assumption that $D_s \neq \delta_s$.

We now claim that $D_i = \delta_i$ at the time s is inserted into S. To prove this claim, notice that $K \subseteq S$. Since s is the first node for which $D_s \neq \delta_s$ when it is added to S, then we must have $D_k = \delta_k$, for every $k \in K$. Let $\phi_i \subseteq K$ be the forwarding set used in the shortest multirate anypath from i to d using the optimal transmission rate $\rho \in R$. By Lemma 4, ϕ_i is composed of the neighbors of i with the shortest distances to d. Assume that $\phi_i = \{1, 2, \ldots, j\}$ with $\delta_1 \leq \delta_2 \leq \ldots \leq \delta_j$. Since s is the first out-of-order node, we know that the nodes in S were settled in order. Therefore, node 1 was settled before node 2, which was settled before node 3, and so on. At the time node 1 is settled, the forwarding set $F_i^{(\rho)}$ is initialized to $F_i^{(\rho)} = \{1\}$. When node 2 is settled, there is no need to check the forwarding set $\{2\}$. By Lemma 4, this forwarding set is never optimal so we just check the set $\{1, 2\}$. By Lemma 5, using $\{1, 2\}$ always provides a shorter distance than using just $\{1\}$. The forwarding set is then updated to $F_i^{(\rho)} = \{1, 2\}$. The same procedure is repeated for each settled node, until we finally have $F_i^{(\rho)} = \phi_i = \{1, 2, \ldots, j\}$. At this time, we also have $D_i^{(\rho)} = \delta_i$, which triggers the update $D_i = D_i^{(\rho)} = \delta_i$, $F_i = F_i^{(\rho)} = \phi_i$, and $T_i = \rho$. Once D_i is equal to the shortest distance δ_i, it does not change anymore and we have $D_i = \delta_i$ at the time s is inserted into S.

We can now prove the theorem with two contradictions. Since node i occurs after node s in the shortest multirate anypath to d, by Lemma 2 we have $\delta_i \leq \delta_s$. In addition, we must also have $\delta_s \leq D_s$ because D_s is never smaller than δ_s. Since both i and s are in $V - S$ and node s was chosen as the one with the minimum distance from Q, then we must have $D_s \leq D_i$ and $\delta_i \leq \delta_s \leq D_s \leq D_i$. From our previous claim, we know that $D_i = \delta_i$ and therefore $D_i = \delta_i \leq \delta_s \leq D_s \leq D_i$. As a result, s is not settled out of order since i has the shortest distance in $V - S$ and $\delta_s = \delta_i$. From this we conclude that the nodes in S are settled in ascending order of distance. Additionally, we also have $D_s = \delta_s$ at the time s is added to S, which contradicts our initial choice of s. We conclude therefore that for each node $s \in V$ we have $D_s = \delta_s$ at the time s is added to S. ■

V. Experimental Results

We evaluated the proposed multirate algorithm using an 18-node 802.11b indoor testbed. Each node is a Stargate microserver equipped with an Intel 400-MHz Xscale PXA255 processor, 64 MB of SDRAM, 32 MB of Flash, and an SMC EliteConnect SMC2532W-B PCMCIA 802.11b wireless network card using the Prism2 chipset. This card has a maximum transmission power of 200 mW and it defaults to a proprietary power control algorithm. The nodes are located in an 2x9 grid and roughly ten meters apart from each other. Each node is equipped with a 3-dB omni-directional rubber duck antenna for the wireless communication. In order to emulate a wireless mesh network with multiple hops, we use a 30-dB SA3-XX attenuator between the wireless interface and its antenna. For 11 Mbps, we have paths of up to 8 hops between each pair of nodes, with 3.1 hops on average. For 1 Mbps, we have a longer transmission range, which reduces the maximum path length to 3 hops, with an average of 1.5 hops between each pair of nodes.

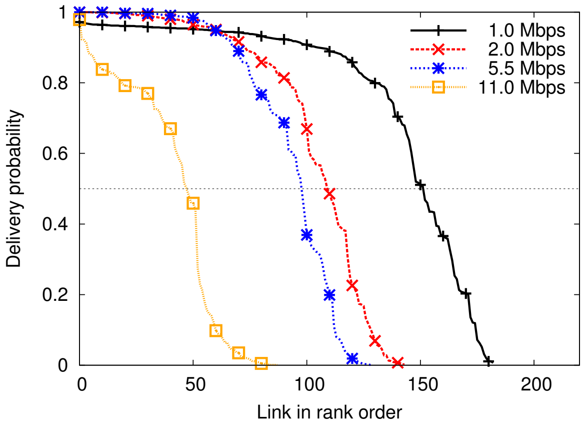
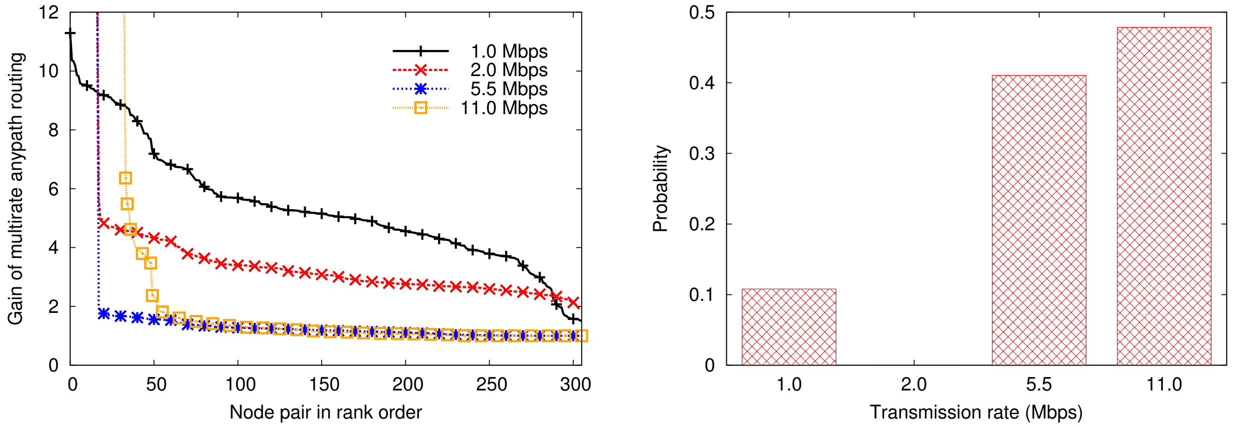


Figure 6. The delivery probabilities of the testbed links for each transmission rate. The data points for each curve are placed in order from largest to smallest (i.e., in rank order). As the rate increases, less links are available and thus path diversity decreases.

We use the testbed to measure the delivery probability of each link at different transmission rates. For that purpose, each node broadcasts one thousand 1500-byte packets and later on we collect the number of received packets at neighbor nodes. We repeat this process for 1, 2, 5.5, and 11 Mbps to have a link estimate for each transmission rate. We use the Click toolkit [14] and a modified version of the MORE software package [3] for the data collection. Our implementation is capable of sending and receiving raw 802.11 frames by using the wireless network interface in monitor mode. We modified the HostAP Prism driver for Linux in order to allow not only 802.11 frame overhearing but also frame injection while in monitor mode. In addition, we extended the HostAP driver to enable it to control the transmission rate of each 802.11 frame sent. The Click toolkit tags each frame with a selected transmission rate and this information is then passed along to the driver. For each frame, our modification reads the information tagged by Click and notifies the wireless interface firmware about the specified transmission rate.

Figure 6 shows the distribution of the delivery probability of each link in the testbed at different 802.11b transmission rates. Every node pair contributes with two links in the graph, one for each direction. Links of each rate are placed in order from largest to smallest (i.e., in rank order). The points of each curve are sorted separately and, therefore, the delivery probabilities



(a)

(b)

Figure 7. Results of the SMAF algorithm for the wireless testbed. (a) Gain of multirate over single-rate anypath routing. For each node pair, we indicate in the y-axis how many times multirate anypath routing is better than single-rate anypath. (b) Histogram of the transmission rate chosen by each node. Optimal transmission rates are not concentrated at any particular rate, indicating that a single-rate algorithm can not perform as well as a multirate algorithm.

of a given x-value are not necessarily from the same link. In wireless mesh networks, higher transmission rates usually have shorter radio ranges and therefore a lower network density. We can see this behavior in Figure 6. As the transmission rate increases, we can see that we have less links available and therefore less path diversity between nodes. For instance, as shown by the dashed horizontal line, the number of links with a delivery probability higher than 50% is 151 at 1 Mbps, 109 at 2 Mbps, 95 at 5.5 Mbps, and only 47 at 11 Mbps. With less paths available at higher rates, we have an interesting tradeoff for multirate anypath routing. With a lower transmission rate, we have more path diversity and a shorter number of hops to traverse, but also a lower throughput. On the other hand, a higher rate results in a higher throughput, but also in less path diversity and a larger number of hops. Our algorithm explores this tradeoff and selects the optimal transmission rate and forwarding set for every node.

The shortest multirate anypath *always* has an equal or lower cost than the shortest single-rate anypath. Otherwise, we would have a contradiction since we can find another multirate anypath (i.e., the single-rate anypath) with a shorter distance to the destination. It is important, however, to quantify how much better multirate anypath routing is over single-rate anypath. For this purpose, we calculate the gain of multirate over single-rate anypath. We define the gain of a given pair of nodes as the ratio between the single-rate anypath distance and the multirate anypath distance between these two nodes. This metric reflects how many times the end-to-end expected transmission time is longer when using single-rate anypath routing as opposed to multirate anypath routing.

Figure 7(a) shows the distribution of this gain for every pair of nodes in the network. Each curve represents the gain over single-rate anypath routing at a fixed rate. We see that the end-to-end transmission time with multirate anypath routing is at least 50% and up to 11.3 times shorter than with single-rate anypath routing at 1 Mbps, with an average gain of 5.4. For higher rates, we also see an interesting behavior depicted by the vertical lines. These lines indicate that several node pairs have an *infinite gain*. The infinite gain occurs because these nodes can not talk to each other at that particular rate due to the poor link quality; the network therefore becomes disconnected. We have 17 (5.6%) node pairs that can not reach each other at both 2 and 5.5 Mbps and 33 (10.8%) node pairs out of reach at 11 Mbps. For the network to be connected, we must then either use a lower rate (e.g., 1 Mbps) for the whole network at the cost of a lower throughput or use multirate anypath routing. For 2 Mbps, if we remove the node pairs with infinite gains, we have a gain of at least 91% and up to 5.6, with an average of 3.2. For 5.5 Mbps, we have a gain up to 2.0, with an average of 22%. Finally, for 11 Mbps, we have a gain up to 6.4, with an average of 80%.

Figure 7(b) shows the reason why multirate always performs better than single-rate anypath routing. In this graph, we show the distribution of the optimal transmission rates selected by each node to reach every other node. We can see that the optimal transmission rates are not concentrated at a single rate, but rather distributed among over several possibilities. We have 10.8% of node pairs using 1 Mbps, 41% using 5.5 Mbps, and 47% using 11 Mbps as the optimal rate. Interestingly enough, no node pair selected 2 Mbps as the optimal rate since it was more beneficial to use another rate instead. If these rates were concentrated at a particular rate, then multirate and single-rate anypath routing would have the same cost. This assumption, however, does not hold in practice and therefore multirate anypath routing always has a higher performance, sometimes manyfold higher as shown in Figure 7(a), than single-rate anypath routing.

VI. Related Work

Most of the work in anypath routing focuses on using a single transmission rate. Due to space constraints, other related works are discussed in [10].

Biswas and Morris [4] designed and implemented ExOR, an opportunistic routing protocol for wireless mesh networks. ExOR follows the same guidelines of single-rate anypath routing explained in Section II. They show that opportunistic routing increases throughput by a factor of two to four compared to single-path routing. Chachulski *et al.* [3] introduce MORE, a routing protocol which uses both opportunistic routing and network coding to increase the network end-to-end throughput. Results show that MORE allows an even higher throughput than ExOR. Our results go beyond and show that an even better performance can be achieved with multirate anypath routing. The authors, however, focus on a single-rate scenario and do not address the multiple bit rates in 802.11.

Besides using a single bit rate, the above-mentioned systems also do not have a systematic approach for selecting the forwarding set. The selection is commonly based on the heuristic that if a neighbor has a smaller ETX distance to the destination, then it should be in the forwarding set. However, the ETX is a single-path metric and do not represent correctly the node's true distance when using anypath routing. To our knowledge, Zhong *et al.* [5] was the first to propose the expected anypath number of transmissions (EATX) metric described in Section II.

Dubois-Ferrière *et al.* [6] introduced a shortest anypath algorithm capable of finding optimal forwarding sets. The authors generalize the well-known Bellman-Ford algorithm for anypath routing and prove its optimality. Chachulski [3] presents a generalization of Dijkstra's algorithm for anypath routing that is similar to our algorithm in Section IV-A, but no proof of optimality was provided. Both algorithms, however, are designed for networks using a single transmission rate. Our work complements their work by generalizing anypath routing for multiple rates. We also provide the proof of optimality of our algorithm, which proofs the single-rate algorithm in [3].

Multiple transmission rates have been recently addressed in opportunistic routing. Zeng *et al.* [15] and Gray *et al.* [16] proposed multirate opportunistic schemes, but their routing algorithms are not provably optimal and they use a different routing metric. Radunovic *et al.* [17] presents an optimization framework to derive routing, scheduling, and rate adaptation schemes for wireless mesh networks. Zeng *et al.* [7] presents a linear-programming formulation to optimize the end-to-end throughput of opportunistic routing, considering multiple rates and transmission conflict graphs. However, in both cases the authors try to optimize several components simultaneously and therefore the posed problem becomes NP-hard. Heuristics are then applied to find a solution, which is not necessarily optimal and may not be easily implemented in practice. Instead, we focus on the shortest multirate anypath problem and provide an optimal solution for it in polynomial time.

VII. Conclusions

In this paper we introduced multirate anypath routing, a new routing paradigm for wireless mesh networks. We provided a solution to integrate opportunistic routing and multiple transmission rates. The available rate diversity imposes several new challenges to routing, since radio range and delivery probabilities change with the transmission rate. Given a network topology and a destination, we want to find both a forwarding set and a transmission rate for every node, such that their distance to the destination is minimized. We pose this as the *shortest multirate anypath problem*. Finding the rate and forwarding set that jointly optimize the distance from a node to a given destination is considered an open problem. To solve it, we introduced the EATT routing metric as well as the Shortest Multirate Anypath First (SMAF) algorithm and presented a proof of its optimality. Our algorithm has the same complexity as Dijkstra's algorithm for multirate single-path routing, being easy to implement in link-state routing protocols.

We conducted experiments in a 18-node 802.11b testbed to evaluate the performance of multirate over single-rate anypath routing. Our main findings are: (1) when the network uses a single bit rate, it may become disconnected since some links may not work at the selected rate; (2) multirate outperforms 11-Mbps anypath routing by 80% on average and up to a factor of 6.4 while still maintaining full connectivity; (3) multirate also outperforms 1-Mbps anypath routing by a factor of 5.4 on average and up to a factor of 11.3; (4) the distribution of the optimal transmission rates are not concentrated at any particular rate, corroborating the assumption that nodes in single-rate anypath routing usually do not transmit at their optimal rates.

Acknowledgment

This work was done in part while the first author was visiting the Ecole Polytechnique Fédérale de Lausanne (EPFL). We thank Martin Vetterli for hosting the first author at EPFL and introducing anypath routing to him. We also thank Deborah Estrin for her help and discussions. This work was supported by the U.S. National Science Foundation under Grants NBD-0721963 and CCF-0120778.

References

[1] M. Campista, P. Esposito, I. Moraes, L. H. Costa, O. C. Duarte, D. Passos, C. V. de Albuquerque, D. C. Saade, and M. Rubinstein, "Routing Metrics and Protocols for Wireless Mesh Networks," *IEEE Network*, vol. 22, no. 1, pp. 6–12, Jan.-Feb. 2008.

[2] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav, "Low-cost Communication for Rural Internet Kiosks using Mechanical Backhaul," in *ACM Mobicom'06*, Sep. 2006, pp. 334–345.

[3] S. Chachulski, "Trading Structure for Randomness in Wireless Opportunistic Routing," Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, May 2007.

[4] S. Biswas and R. Morris, "ExOR: Opportunistic Multi-Hop Routing for Wireless Networks," in *ACM SIGCOMM'05*, Aug. 2005, pp. 133–143.

[5] Z. Zhong, J. Wang, S. Nelakuditi, and G.-H. Lu, "On Selection of Candidates for Opportunistic AnyPath Forwarding," *ACM SIGMOBILE Mobile Comp. and Comm. Review*, vol. 10, no. 4, pp. 1–2, Oct. 2006.

[6] H. Dubois-Ferriere, "Anypath Routing," Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, Nov. 2006.

[7] K. Zeng, W. Lou, and H. Zhai, "On End-to-End Throughput of Opportunistic Routing in Multirate and Multihop Wireless Networks," in *IEEE Infocom'08*, Apr. 2008, pp. 816–824.

[8] R. Draves, J. Padhye, and B. Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks," in *ACM MobiCom'04*, Sep. 2004, pp. 114–128.

[9] S. Jain and S. R. Das, "Exploiting Path Diversity in the Link Layer in Wireless Ad Hoc Networks," *Ad Hoc Networks*, vol. 6, no. 5, pp. 805–825, Jul. 2008.

[10] R. Laufer and L. Kleinrock, "Multirate Anypath Routing in Wireless Mesh Networks," UCLA Computer Science Department, Tech. Rep. UCLA-CSD-TR080025, Aug. 2008.

[11] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Measurement-Based Models of Delivery and Interference in Static Wireless Networks," in *ACM SIGCOMM'06*, Sep. 2006, pp. 51–62.

[12] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing," in *ACM MobiCom'03*, Sep. 2003, pp. 134–146.

[13] T. T. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to algorithms*, 2nd ed. Cambridge, MA, USA: MIT Press, 2001.

[14] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click Modular Router," *ACM Transactions on Computer Systems*, vol. 18, no. 3, pp. 263–297, 2000.

[15] K. Zeng, W. Lou, and Y. Zhang, "Multi-Rate Geographic Opportunistic Routing in Wireless Ad Hoc Networks," in *IEEE MILCOM'07*, Oct. 2007, pp. 1–7.

[16] C. Gray, N. Santhapuri, and S. Nelakuditi, "On Bit-Rate Selection for Opportunistic Routing," in *WiMesh'08*, Jun. 2008, pp. 1–6.

[17] B. Radunovic, C. Gkantsidis, P. Key, and P. Rodriguez, "An Optimization Framework for Opportunistic Multipath Routing in Wireless Mesh Networks," in *IEEE Infocom'08*, Apr. 2008, pp. 2252–2260.