



# High-throughput multicast routing metrics in wireless mesh networks

Sabyasachi Roy, Dimitrios Koutsonikolas, Saumitra Das, Y. Charlie Hu \*

*School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, United States*

Received 24 August 2006; received in revised form 9 July 2007; accepted 22 July 2007

## Abstract

The stationary nature of nodes in a mesh network has shifted the main design goal of routing protocols from maintaining connectivity between source and destination nodes to finding high-throughput paths between them. Numerous link-quality-based routing metrics have been proposed for choosing high-throughput routing paths in recent years. In this paper, we study routing metrics for high-throughput tree or mesh construction in multicast protocols. We show that there is a fundamental difference between unicast and multicast routing in how data packets are transmitted at the link layer, and accordingly how the routing metrics for unicast routing should be adapted for high-throughput multicast routing. We propose a low-overhead adaptive online algorithm to incorporate link-quality metrics to a representative multicast routing protocol. We then study the performance improvement achieved by using different link-quality-based routing metrics via extensive simulation and experiments on a mesh-network testbed, using ODMRP as a representative multicast protocol.

Our extensive simulation studies show that: (1) ODMRP equipped with any of the link-quality-based routing metrics can achieve higher throughput than the original ODMRP. In particular, under a tree topology, on average, ODMRP enhanced with link-quality routing metrics achieve up to 34% higher throughput than the original ODMRP under low multicast sending rate; (2) the improvement reduces to 21% under high multicast sending rate due to higher interference experienced by the data packets from the probe packets; (3) heavily penalizing lossy links is an effective way in the link-quality metric design to avoid low-throughput paths; and (4) the path redundancy from a mesh data dissemination topology in mesh-based multicast protocols provides another degree of robustness to link characteristics and reduces the additional throughput gain achieved by using link-quality-based routing metrics. Finally, our experiments on an eight-node testbed show that on average, ODMRP using SPP and PP achieves 14% and 17% higher throughput over ODMRP, respectively, validating the simulation results.

© 2007 Elsevier B.V. All rights reserved.

**Keywords:** Wireless mesh networks; Multicast; High-throughput routing metrics; Testbed

## 1. Introduction

Recently, wireless mesh networks have attracted much attention [45–49]. These networks typically have low maintenance overhead, high data rates,

\* Corresponding author. Tel.: +1 765 494 9143; fax: +1 765 494 3371.

E-mail addresses: [roy0@purdue.edu](mailto:roy0@purdue.edu) (S. Roy), [dkoutson@purdue.edu](mailto:dkoutson@purdue.edu) (D. Koutsonikolas), [smdas@purdue.edu](mailto:smdas@purdue.edu) (S. Das), [ychu@purdue.edu](mailto:ychu@purdue.edu) (Y.C. Hu).

and are not energy constrained. Such networks, also known as “community wireless networks”, can be used for various applications such as shared broadband access, neighborhood gaming, video surveillance, and media repository.

Unlike traditional mobile ad hoc networks (MANETs), the routers in mesh networks are static, and thus dynamic topology changes are much less of a concern in such networks. As a consequence, the main design goal for routing protocols is shifted from maintaining connectivity between source and destination nodes to finding high-throughput paths between the nodes. Towards this goal, more sophisticated routing metrics than the hop-count metric need to be used to find paths that achieve high throughput, as protocols based on the hop-count metric often choose long links which tend to be lossy and give low throughput.

Unicast is a fundamental routing service in multihop mesh networks. As such, much work has been done recently on finding a high-throughput routing metric for unicast routing protocols in wireless mesh networks. For example, the expected transmission count (ETX) metric was proposed in [8] to take into account the loss rates of the links. Similarly, the round trip time (RTT) and packet pair (PP) metrics were proposed in [1,22], respectively, to take into account the delay characteristics of the links. Finally, the Weighted cumulative expected transmission time (WCETT) metric was proposed in [13] to take into account the link bandwidth and loss rates of links in the presence of multiple non-overlapping channels. In [4,11], two link-quality-based routing metrics were proposed to minimize the expected amount of energy needed for end-to-end data transmission assuming a reliable link layer and an unreliable link layer, respectively. All these metrics have been proposed and evaluated for unicast routing protocols such as DSDV [35], DSR [21], and AODV [36].

Multicast is another fundamental routing service in multihop mesh networks. It provides an efficient means of supporting collaborative applications such as video conferencing, online games, webcast and distance learning, among a group of users [10]. Previously, a large number of multicast protocols have been designed to efficiently maintain a distributed multicast routing structure in dynamically changing topologies in MANETs [38,18,26,43,16,39,19,17,31]. All of these protocols use minimum-hop-count as the routing metric and focus on scenarios with high mobility. Similarly as for unicast protocols, the sta-

tionary nature of nodes in a mesh network also shifts the main design goal of multicast routing protocols from efficiently maintaining a distributed multicast routing structure to finding a high-throughput multicast routing structure such as a tree or a mesh. Despite the numerous recent studies on link-quality-based routing metrics for high-throughput unicast routing in mesh networks, there has been little work on link-quality-based routing metrics for high-throughput multicast routing.

In this paper, we study the design of link-quality-based routing metrics for high-throughput multicast in mesh networks. One approach is to simply adopt the unicast routing metrics such as ETX and WCETT for multicast protocols, i.e., in the tree or mesh construction. However, these metrics may not work well as the way they reflect the link quality depends very much on the link layer data transmission, and multicast routing is inherently different from unicast routing in the way the link layer handles data packets. In particular, while unicast routing protocols use unicast to send data packets at the link layer, most multicast routing protocols use broadcast at the link layer for disseminating data packets. The fundamental difference between link layer broadcast and link layer unicast is that the former has no link layer acknowledgments and consequently no link layer retransmissions. This difference has two immediate implications on multicast routing: (1) the link quality that matters is *unidirectional*, since per-hop data transmission no longer involves two round trips (RTS/CTS/DATA/ACK); and (2) each node has only *one chance* to properly transmit a data packet at the link layer since there are no retransmissions. These implications suggest that link-quality metrics designed for unicast may no longer be appropriate for multicast, i.e., for constructing a high-throughput multicast tree or mesh.

In this paper, we first study how to adapt the routing metrics developed for unicast for use in multicast in mesh networks. We then study the performance of a set of five routing metrics adapted from those for unicast protocols, namely, ETT, ETX, PP, Multicast ETX (METX) and Success Probability Product (SPP), where METX and SPP are adapted from two energy-efficient routing metrics proposed in [4,11]. Our study is performed using ODMRP [26], a state-of-the-art proactive multicast protocol. For each routing metric, we modify ODMRP to construct the routing structure based on that routing metric. We also compare the modi-

fied versions of ODMRP with the original ODMRP. Our studies are conducted via extensive simulations as well as experiments on a mesh network testbed consisting of eight nodes deployed on the second floor of an office building on the Purdue campus.

Our simulation study shows that ODMRP using any of the five metrics, ETT, ETX, METX, PP, and SPP, outperforms the original ODMRP by significant margins. These margins of improvement are similar to those achieved in unicast routing using high-throughput routing metrics [12]. In particular, on average, ODMRP using SPP or PP achieves 34% and 21% higher throughput than the original ODMRP under low and high load, respectively. Our testbed experiments show that on average, ODMRP using SPP and PP achieves 14% and 17% higher throughput over ODMRP, respectively. Finally, our simulation studies show that path redundancy in a mesh topology provides another degree of robustness to link characteristics. In particular, when the number of sources per group is increased from one to two, the increased path redundancy in the original ODMRP causes it to perform reasonably well, and the additional throughput improvement from using link-quality-based routing metrics is reduced by around 15% under both low and high loads.

The main contributions of this paper are as follows:

- We point out that high-throughput routing metrics design for unicast routing cannot be directly applied to multicast routing due to a fundamental difference between unicast and multicast routing in how data packets are transmitted at the link layer.
- We show how to adapt the routing metrics for unicast routing for high-throughput multicast routing.
- We propose a low-overhead online adaptive scheme for incorporating link-quality metrics to a representative multicast protocol.
- We present detailed simulation results of ODMRP based on different routing metrics which demonstrate significant throughput gain from using high-throughput routing metrics.
- We present experimental results on a testbed that validate the simulation results.
- We present tradeoffs between probing overhead and throughput gain in using high-throughput routing metrics.
- We show that path redundancy in a mesh topology provides another degree of robustness to link characteristics which reduces the throughput gain from using high-throughput routing metrics.

To the best of our knowledge, this is the first study on high-throughput routing metrics for multicast in wireless mesh networks.

The rest of the paper is organized as follows. Section 2 describes the routing metrics previously proposed for unicast routing protocols. Section 3 discusses the fundamental difference between multicast and unicast modes of communication in wireless multihop networks and describes how to accordingly modify the existing unicast routing metrics for multicast routing. Section 2.2 gives a brief overview of ODMRP and Section 4 the changes made to ODMRP in order to incorporate the routing metrics. Section 5 presents simulation results and Section 6 presents experimental results on a mesh network testbed. Finally, Section 7 reviews related work and Section 8 concludes the paper.

## 2. Background

In this section, we first give an overview of existing routing metrics for unicast protocols followed by a description of a representative multicast protocol for wireless networks, namely, ODMRP.

### 2.1. Routing metrics for unicast protocols

Recently, several link-quality-based routing metrics have been proposed for unicast routing protocols in wireless networks. They are based on the characteristics of the underlying link such as bandwidth, loss, and delay. In the following, we briefly review these metrics:

- (a) *RTT*. In [1], Adya et al. proposed the per-hop round trip time (RTT) metric which measures the round trip delay seen by unicast probes exchanged between neighboring nodes. To estimate RTT, each node sends a probe packet with a timestamp to each of its neighbors every 500 ms. Each neighbor responds to the probe it receives with an acknowledgment. In this way, the sender can measure the RTT for that probe and estimate an average value based on the current measurement and the previous estimated average. Finally, the

routing algorithm selects the path with the least cumulative RTT of all the links along the path.

- (b) *PP*. The packet pair (PP) metric was first proposed by Keshav in [22] to measure the delay between a pair of back-to-back probes to a neighboring node. The use of two successive probe packets eliminates the effect of queuing delays. It was one of the routing metrics used in a comparison study by Draves et al. [12]. In this implementation, each node periodically unicasts two back-to-back probe packets, one small and one large, to each of its neighbors. Each neighbor computes the delay between the arrival of the two probes and sends this delay back to the sender. An exponentially weighted moving average (EWMA) of the delays is maintained at each node for each of its neighbors. The path with the least sum of delays at all of its constituent links is selected by the routing algorithm.
- (c) *ETX*. The previous two schemes use link delay in their metrics. In [8], De Couto et al. proposed the expected transmission count (ETX) metric which is based on the expected number of transmissions required to send a unicast packet over a link, including retransmissions. To calculate ETX, each node measures the probability that a packet successfully reaches the receiver, denoted as  $d_f$ , and the probability that an ACK is successfully received by the sender, denoted as  $d_r$ . The ETX value of the link is given by

$$ETX = \frac{1}{d_f \times d_r} \quad (1)$$

The routing algorithm then selects the path with the least sum of ETX values of all its constituent links. To measure  $d_f$  and  $d_r$ , each node broadcasts a probe packet every second. Each such probe contains the number of probes the node received from each of its neighbors in the previous 10 s. Since the 802.11 MAC layer protocol does not retransmit broadcast packets, nodes use this information to estimate the forward and reverse delivery probabilities.

- (d) *WCETT*. In [13], Draves et al. proposed the weighted cumulative expected transmission time (WCETT) metric as an enhancement to ETX by considering both the loss rate and the bandwidth of links. In addition, it considers available channel diversity. WCETT along

a path with  $n$  hops and  $k$  different channels is given by the following formula:

$$WCETT = (1 - \beta) \times \sum_{i=1}^n ETT_i + \beta \times \max_{1 \leq j \leq k} X_j \quad (2)$$

where

$$X_j = \sum_{\text{hop } i \text{ is on channel } j} ETT_i$$

for  $1 \leq j \leq k$  and  $0 \leq \beta \leq 1$ . Finally, ETT is estimated as

$$ETT = ETX \times \frac{S}{B} \quad (3)$$

where  $S$  is the packet size and  $B$  the bandwidth of the link. To calculate ETT for each link, it measures ETX using the technique mentioned above, and the bandwidth using a technique similar to packet pair.

- (e) *Energy aware routing metrics*. In [4,11], the authors propose routing metrics to minimize the total transmission energy. They study the problem of energy-efficient routing under two cases: the case where the link layer ensures reliability through hop-by-hop retransmissions (HHR), and the case where the link layer is unreliable and reliability can be guaranteed only through end-to-end retransmissions (EER). For HHR, the following routing metric is proposed:

$$HHR = \sum_{i=1}^n \frac{E_i}{1 - p_{\text{err}i}} \quad (4)$$

where  $i$  denotes the  $i$ th link,  $E_i$  is the energy required to transmit over that link, and  $p_{\text{err}i}$  is the error rate of that link. Eq. (4) gives the expected amount of energy needed to transmit a packet over a path consisting of  $n$  links. For EER, the following approximate metric was proposed in [4]

$$EER_{\text{approx}} = \frac{\sum_{i=1}^n E_i}{\prod_{i=1}^n (1 - p_{\text{err}i})} \quad (5)$$

with the notations having the same meanings as above. In [11], an optimal metric for EER which improves upon Eq. (5) was proposed as

$$C(s, d) = \frac{1}{1 - p_{\text{err}l}} [C(s, u) + W(u, d)] \quad (6)$$

where  $C(s,d)$  is the expected energy-cost of transmission from a source  $s$  to destination  $d$ ,  $l$  is the link between  $u$  and  $d$  in the path, and  $W(u,d)$  is the transmission energy required between nodes  $u$  and  $d$ . Since the above metrics for EER assume there is no link layer reliability and hence no retransmissions, they can be directly adopted for multicast routing.

## 2.2. Overview of ODMRP

ODMRP employs a route discovery procedure to create a multicast mesh in which a set of nodes are responsible for forwarding data packets to the multicast group member nodes using link-layer broadcast. When a source has data packets to send, it periodically floods the network with a JOIN QUERY. Each intermediate node that receives the JOIN QUERY checks if it is a duplicate based on the sequence number present in the packet header. If not, it stores its upstream node identifier in its ROUTING TABLE and rebroadcasts it. Once the JOIN QUERY reaches a member node of the multicast group, the node creates a JOIN TABLE and broadcasts a JOIN REPLY containing the JOIN TABLE to its neighbors. The JOIN TABLE contains two fields: the source and the last hop from which the node received the JOIN QUERY, as shown in Fig. 1. When a node receives a JOIN REPLY, it checks whether it is the last hop in any of the entries in the JOIN TABLE contained in the JOIN REPLY packets. If so, it sets its FG\_FLAG and makes itself a *forwarding node* for that multicast group. It propagates the JOIN REPLY packet with a new JOIN TABLE containing the source and the last hop from the source to itself. This information about the last hop from the source to itself is stored in its ROUTING TABLE. Once the JOIN REPLY packets from all the members reach the source, the paths have been set up from that source to all the members of the group. This process builds a mesh of nodes called the FORWARDING GROUP. Once

the FORWARDING GROUP is built, the source can start sending data packets to the group members. Any intermediate node that receives a data packet forwards the packet if its FG\_FLAG is set for that multicast group using link-layer broadcast. In the basic ODMRP, the paths are effectively chosen using the minimum hop-count as the path metric. In Section 4, we propose how to incorporate other link-quality metrics in ODMRP. Note that ODMRP is chosen as a representative multicast protocol and our techniques can be employed in any multicast protocol.

## 3. Routing metrics for multicast protocols

In this section, we first discuss the differences in the way the link layer handles data packets in unicast and multicast and the implications on the design of high-throughput link-quality-based routing metrics. We then present how to adapt different link-quality metrics originally designed for unicast routing for use in multicast routing.

### 3.1. Differences between link-layer unicast and multicast

Data packets are handled differently at the link layer in unicast routing and multicast routing, and the difference has direct implications on the design of high-throughput link-quality metrics. Most multicast protocols (for example, [26,6,38,18,42]) use link-layer broadcast for efficiency reasons. In contrast, data packets in unicast are handled using link-layer unicast. The most commonly used link/MAC layer protocol in wireless ad hoc networks is the IEEE 802.11 MAC layer protocol. The 802.11 MAC layer unicast involves an RTS/CTS exchange before sending a data packet. The RTS/CTS exchange avoids the hidden terminal problem by reserving the channel via a virtual carrier sense mechanism. This reduces the probability of collision during data packet transfer. Further, data transmission is acknowledged by the receiver. If an acknowledgment is not received, the MAC layer reattempts the data retransmission for a number of times. In contrast, the 802.11 MAC layer broadcast does not involve any RTS/CTS exchange before sending a broadcast packet. This effectively increases the probability of collisions. Furthermore, it does not involve any link layer acknowledgment or data retransmission. This further reduces the reliability of broadcast transmission.

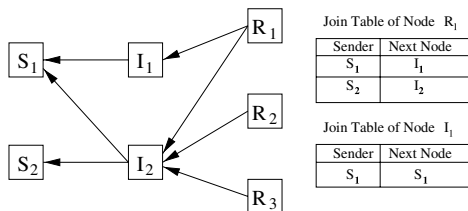


Fig. 1. A simple example describing JOIN TABLE FORWARDING [26]. JOIN TABLES containing the previous hop from a source are sent in JOIN REPLIES.



The abovementioned differences in unicast and broadcast data transmissions have two major implications on the design of link-quality metrics. First, the link quality that matters is *bidirectional* in unicast, but *unidirectional* in multicast. In case of unicast, a successful data transfer consists of a successful transfer of a packet from a sender (S) to a receiver (R) followed by a successful transfer of an acknowledgment from R to S, in addition to an exchange of RTS/CTS between the two nodes, as shown in Fig. 2. Hence, the overall quality of a link depends on the link characteristics in both the forward and reverse direction. For example, if loss characteristics of a link are taken into account, then probabilities of a successful packet transfer in both the forward direction, namely  $d_f$ , and the reverse direction, namely  $d_r$ , should be considered. In case of broadcast, there are no acknowledgments and thus a successful data transfer only depends on the link quality in the forward direction. For example, the loss characteristics of a link are determined only by  $d_f$  from S to R, as shown in Fig. 3. In fact, even if the reverse path is bad, i.e.,  $d_r$  is low, the link S to R can still be a low loss link for broadcast packets as long as  $d_f$  is high. Hence, in case of broadcast, the link quality in the reverse direction should not be considered in the link-quality metric as it may distort the metric value of a link. Second, since in broadcast there are no retransmissions, a data packet has only once chance to properly travel from one node to another. This implies that unlike unicast, for loss-rate-based link-quality metrics such as ETX, simple addition of the metric values of the individual links along a path does not properly

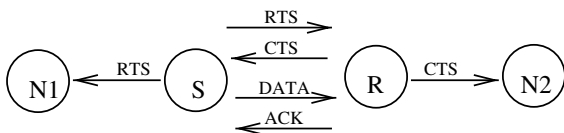


Fig. 2. Packet exchanges done at the link layer during unicast data transfer. Successful data transfer involves a successful data transmission and a successful acknowledgment reception. Bidirectional link quality is critical.

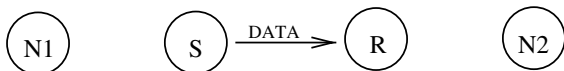


Fig. 3. Packet exchanges done at the link layer during broadcast data transfer. Successful data transfer does not involve any packets other than the data packet. Unidirectional link quality is critical.

reflect the quality of the entire path. Instead, a product of the metric values of the individual links, e.g., the success probabilities ( $d_f$ ) of individual links, better reflects the quality of the path.

### 3.2. Adapting unicast link-quality metrics for multicast

The above differences between unicast and multicast suggest that the link-quality metrics designed for unicast (discussed in Section 2) can not be directly used in multicast protocols. In the following, we describe how to adapt these link-quality metrics for use in multicast protocols.

- PP*. Since in multicast the data packets are broadcast at the link layer, the delay measurement is more accurate if a node broadcasts probe packets instead of unicasting them to each neighbor node. As before, the delay for a link is calculated as an Exponentially Weighted Moving Average. We assign a weight of 90% to the accumulated average and 10% to the current one. Another modification we introduced is that in case either the large or the small packet is lost, a 20% penalty is imposed. As we will see in the performance evaluation, the penalty plays a major role in determining high-throughput paths. In our experiments, the probe packets were sent periodically every 10 s.
- ETX*. In unicast, ETX is calculated using both  $d_f$  and  $d_r$  of a link. In multicast, as there are no link layer acknowledgments, we do not consider  $d_r$  in Eq. (1). Hence, ETX is now defined as

$$ETX = \frac{1}{d_f} \quad (7)$$

To measure ETX, the same probe mechanism as described in Section 2 is used, except the probe packets no longer contain any information, and each node simply counts the number of the probe packets it received in the past 10 probing intervals. The probing interval is set to 5 s.

- ETT*. Since we do not assume multiple channels when comparing different metrics in this paper, we adapt ETT instead of WCETT for use in multicast. The value of the metric for a path is the sum of the ETT values of the individual links. To measure ETT (Eq. (3)),

each node periodically sends two back-to-back probe packets every 10 s, one small and one large, as in Packet Pair. The receiver uses the small packets received in the past 10 probing intervals to calculate ETX. The bandwidth of each link is estimated using packet pair, but unlike in [13], we estimate the available bandwidth instead of the raw bandwidth, by dividing the size of the large packet with the delay between the two probe packets and maintaining an average of the bandwidth metric value over the last 10 probing intervals. Different from the adapted PP metric above, we do not impose a penalty on the delay values in case of probe packet losses, since the loss rate is already taken into account in the ETX component.

- (d) *METX*. We modify the metric given by Eq. (6) to a new metric Multicast ETX (*METX*). Since mesh networks are not energy constrained, we set  $W(u,d)$  in Eq. (6) to 1. Such a transformation gives us the total expected number of transmissions needed by all the nodes from a source to a destination in order to guarantee successful reception of at least one packet at the receiver. *METX* can be expressed as

$$METX = \sum_{i=1}^n \frac{1}{\prod_{j=i}^n (1 - p_{errj})} \quad (8)$$

where  $i$  denotes the  $i$ th link along a path from a source to a destination comprising  $n$  links.

- (e) *SPP*. We modified Eq. (5) to propose the success probability product (*SPP*) metric. The value of *SPP* for a path consisting of  $n$  links is given by

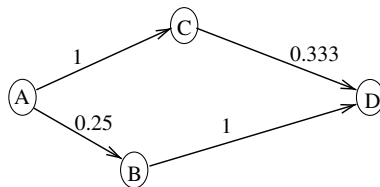
$$SPP = \prod_{i=1}^n d_{fi} \quad (9)$$

where  $d_{fi}$  is the  $d_f$  value for link  $i$  ( $d_{fi} = 1 - p_{err_i}$ ). Note that if we set  $E_i$  in Eq. (5) to 1, the resulting value is  $n$  times the value of

$1/SPP$  along the same path which gives an approximation of the expected total number of transmissions needed by all nodes along a path from the source to the destination. In contrast, *SPP* (actually  $1/SPP$ ) reflects the expected number of transmissions at the source itself. The intuition behind such a modification is as follows. In multicast, data packets are broadcast at the link layer without acknowledgments or retransmissions. Hence, a packet is sent to each intermediate node only once, and is either received properly or lost. As a consequence, the probability for the destination node to receive a packet properly over a path is the product of the probabilities that the packet is properly received at each intermediate node along that path, since the transmissions at these intermediate nodes are independent events. This probability is exactly *SPP*. The routing algorithm selects the path with the maximum *SPP*. Note that unlike all other metrics described in this paper, a high value of *SPP* for a path implies a good (high-throughput) path and a low value implies a bad (low-throughput) path.

Being a product, *SPP* is more effective in avoiding paths containing bad links than *ETX* as a single high-loss link (low  $d_f$ ) decreases the value of the metric for the entire path multiplicatively, as opposed to additively in *ETX*. Fig. 4 gives an example showing why *SPP* is superior to a metric such as the one given by Eq. (5) or *METX* which tries to minimize the total number of transmissions. The network consists of two paths from A to D, namely  $A \rightarrow B \rightarrow D$  and  $A \rightarrow C \rightarrow D$ . Clearly, in this case *METX* and *SPP* choose different paths, namely  $A \rightarrow B \rightarrow D$  and  $A \rightarrow C \rightarrow D$ , respectively. However, it is clear that D has a better chance of getting a packet through  $A \rightarrow C \rightarrow D$  and hence *SPP* chooses the better path.

Note that ideally, *SPP* is likely to be the metric best throughput performance as it maximizes the



Path	METX	1/SPP
A–C–D	6	3
A–B–D	5	4

Fig. 4. *SPP* can choose higher-throughput paths as opposed to *METX* at the expense of greater number of transmissions to reach a destination. The numbers over the links denote the forwarding probability ( $d_{fi} = 1 - p_{err_i}$ ) of each link.

Table 1

Routing metrics for multicast protocols and the link characteristics they measure

Characteristic	Loss	Loss + delay	Loss + B/w
Metric	METX, ETX, SPP	PP	ETT

probability that a packet successfully reaches the intended recipient. However, it is interesting to see how SPP performs under practical circumstances such as varying available channel capacity, fading and collisions. Moreover, SPP does not account for available bandwidth (accounted for by ETT and PP) and total number of packet transmissions in the network (accounted for by ETX and METX). Such factors may further affect the final throughput performance of a routing metric. Thus, it is not obvious that SPP would be an outright winner among all routing metrics. In fact, Section 5 shows that SPP is not the metric with the best throughput under all topology instances (scenarios), though it does achieve the best throughput on average. Furthermore, we also observed that SPP is not as aggressive in penalizing lossy links as PP under some cases we encountered in our testbed experiments in Section 6. Finally, we also study other application layer metrics such as delay. Studying routing metrics other than SPP gives us insights about the tradeoffs involved in their design, for example, SPP causes slightly more end-to-end delay as compared to ETX as it chooses longer paths with short links.

### 3.3. Summary

Table 1 classifies the link-quality metrics into three categories based on the link characteristics that they measure. The first category includes ETX, METX and SPP; these are purely loss-based metrics. The second category includes PP which takes into account both loss and delay. Finally, the third category includes ETT which takes into account both loss and bandwidth. We do not study RTT in this paper because in [12] RTT has been shown to be problematic as the RTT measurements get distorted due to queuing delays.

## 4. Incorporating link quality metrics in a multicast protocol

To evaluate the throughput improvement under the various link-quality metrics for multicast, we choose ODMRP [26] as a representative multicast

protocol for wireless multihop networks. In this section, we describe the distributed implementation of incorporating link-quality metrics in ODMRP.

### 4.1. Incorporating link-quality metrics

To incorporate the new link-quality metrics into ODMRP, we modified ODMRP as follows. Each node maintains a new data structure, called the NEIGHBOR TABLE, that records the costs of the links from its neighbors to itself. The costs are defined according to the link-quality metric being used, and are periodically updated. In the modified ODMRP, each node looks up the NEIGHBOR TABLE for the cost of the link from which it received the JOIN QUERY and using this link cost, it updates the cost in the JOIN QUERY packet before rebroadcasting it. Finally, when the JOIN QUERY reaches a group member, it contains the total cost of the path traveled.

Figs. 5–7 demonstrate the manner in which the link-quality metrics are incorporated in the multicast routing scheme. Initially neighbors probe each other to obtain a value of the link-quality metric to their neighbors in each direction. Nodes store such values in their NEIGHBOR TABLE. For example, Fig. 5 shows the link-quality metric values stored by node B in its NEIGHBOR TABLE (NT-B). It has entries for both its neighbors A and C with link-

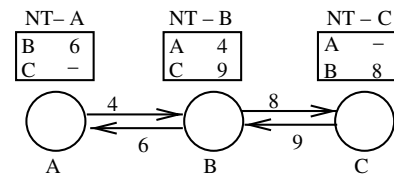


Fig. 5. Maintaining the link-quality metrics in each node's NEIGHBOR TABLE. The solid arrows are labeled with their link-quality metric values.

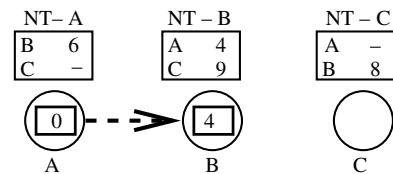


Fig. 6. Propagation of JOIN QUERY denoted by the dashed arrows from source to destination: first hop. An intermediate node updates the value of the path-quality metric in the JOIN QUERY packet based on the link-quality value stored in its NEIGHBOR TABLE.



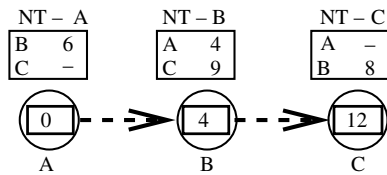


Fig. 7. Propagation of JOIN QUERY from source to destination: second hop. When the JOIN QUERY packet reaches the destination, it has the value of the path-quality of the entire path.

quality metric values of 4 and 9, respectively. Once ODMRP starts running, a JOIN QUERY packet is flooded in the network by the source node A as shown in Fig. 6. Each rectangular box inside a node (denoted by a circle) denotes the JOIN QUERY packet, and the value inside is the path-quality metric value of the partial path traversed. When the JOIN QUERY packet leaves A, its path-quality metric value is 0. When it reaches the intermediary B, B stamps the path-quality metric value (of the partial path traversed till B) using the information from NT-B. B propagates the JOIN QUERY packet further as shown in Fig. 7. Finally when the JOIN QUERY packet reaches the destination node C, C updates the path-quality metric value which becomes the path-quality metric value for the entire path from A to C. C may receive many such JOIN QUERY packets along different paths in the network over a period of time, but it chooses the best path among them (see the following section) and sends back a JOIN REPLY along that path in the reverse direction as before.

#### 4.2. Achieving path diversity

To achieve diversity in the paths received, instead of sending back a JOIN REPLY immediately after getting the first JOIN QUERY, a group member waits for a period of  $\delta$  seconds. During this period, it accumulates several duplicate JOIN QUERY packets and stores the best among them, based on the cost of the path traveled by each JOIN QUERY. After the period of  $\delta$  seconds, the member constructs the JOIN TABLE using the stored JOIN QUERY, i.e., the best among all JOIN QUERY packets received during the  $\delta$  seconds, and broadcasts the JOIN REPLY to its neighbors as before. Note that the  $\delta$ -second period effectively controls the diversity of the paths that a member gets to choose from. The  $\delta$ -second period should be large enough to ensure that the JOIN QUERY packets from most of the diverse paths are received. This imple-

mentation is similar to the version of ODMRP that uses mobility prediction [28].

To achieve more diversity in the paths received by each group member, each intermediate node is allowed to forward duplicate JOIN QUERY packets similarly as in [33]. Such a scheme may lead to a broadcast explosion of the queries. To prevent such a case from happening, we impose two restrictions. First, a duplicate query is forwarded only if the cost of the path it has traveled is less than that of the minimum cost query received so far. Second, each node sets a timer for a period  $\alpha < \delta$  seconds when it receives the first JOIN QUERY with a particular sequence number. The value of  $\alpha$  is kept smaller than  $\delta$  because a member accepts queries only for a period of  $\delta$  seconds and drops all duplicate queries received after the expiration of that period. It is important to choose  $\alpha$  carefully as a very small value will lead to small path diversity, and a too high value may lead to a high query processing overhead. Such a timer  $\alpha$  limits the number of queries transmitted by a node to only as many packets that can be sent during that duration, which is bounded by a constant  $C$ . As a result, the total number of queries transmitted in the network grows as  $C \times N$ , where  $N$  is the number of nodes in the network.

#### 4.3. Adaptive online determination of $\alpha$ and $\delta$

As mentioned above, determining the best value of  $\alpha$  and  $\delta$  is important to the performance of the different routing metrics. The optimal values of these parameters depend on the characteristics of the network in which the different routing metrics operate. Such characteristics may include channel quality, the number of nodes in the network, the number of receivers and senders, the number of multicast groups, and the network diameter. Thus, hard coding the values of these parameters is not appropriate. One of the possible ways to obtain the best value of these parameters is a trial-and-error based approach, periodically conducted by the network administrator. However, such a scheme will not be able to cater to the transient changes in the network conditions. In the following, we propose an adaptive, online scheme for determining the best values of these parameters.

Our adaptive scheme works as follows. Every time the source sends a JOIN QUERY, it also piggy-backs a value for the parameter  $\alpha$  with the same packet. Such a value is unique to every source of every group. On receiving the JOIN QUERY, the

intermediate nodes follow the same procedure as described in Section 4.2. The only difference being the fact that now instead of using a predefined value of  $\alpha$ , they use the value present in the JOIN QUERY packet. The receivers use a  $\delta$  value equal to  $K \cdot \alpha$  where  $K$  is a small constant larger than one. For our simulations, we use  $K = \frac{4}{3}$ .

To search for the best  $\alpha/\delta$  values during initialization and throughout a multicast session, the source continuously collects feedbacks from the group members on the quality of the multicast mesh, as shown in Fig. 8. We denote the periodic interval when the source node  $S$  sends out a new JOIN QUERY as a cycle. Assume at the beginning of cycle  $i + 1$ , the source node sends  $\alpha_{i+1}$  and sets a binary direction vector  $DV_{i+1}$  as *up*.  $DV_{i+1}$  essentially remembers whether the value of  $\alpha$  was increased or decreased for cycle  $i + 1$  and is set as either *up* or *down*. The JOIN QUERY is propagated by intermediate nodes using  $\alpha_{i+1}$  and eventually every receiver  $R$  receives  $\alpha_{i+1}$ , and after waiting for  $\delta_{i+1}$ , it sends back the  $(i + 1)$ th JOIN REPLY. It also calculates the number of packets received during cycle  $i$  (approximated using the number of packets received between receiving the  $i$ th and the  $(i + 1)$ th JOIN QUERY), and piggybacks this information  $FB_i$  to the source via the  $(i + 1)$ th JOIN REPLY packet. The same process repeats at each mesh rebuilding cycle of ODMRP. Note there is a one-cycle delay in the feedback process. The  $(i + 2)$ th JOIN REPLY contains the feedback for the multicast mesh built based on the  $(i + 1)$ th  $\alpha/\delta$  values, i.e., the mesh built based on the  $(i + 1)$ th JOIN REPLY packets from the members. Because of this delay, after receiving the  $(i + 2)$ th JOIN REPLY, the source has to decide on the  $(i + 3)$ th  $\alpha$  value based on the feedbacks from using the  $i$ th and the  $(i + 1)$ th  $\alpha/\delta$  values. For this reason, each  $\alpha$  value (and  $\delta$  value) is used in the JOIN QUERY for two consecutive cycles, i.e.,  $\alpha_{i+2} = \alpha_{i+1}$ ,  $\alpha_i = \alpha_{i-1}$ , and so on.

Based on the number of packets received during the  $i$ th cycle and the  $(i + 1)$ th cycle, the source adapts the value of  $\alpha$  for the  $(i + 3)$ th cycle as follows. If the total number of packets received by the receivers during the  $(i + 1)$ th cycle is less than that during the  $i$ th cycle, i.e.,  $FB_{i+1} < FB_i$ , and  $DV_{i+1}$  was *down*, then the source increases the value of  $\alpha$  by a constant factor  $t$ , i.e.,  $\alpha_{i+3} = t \cdot \alpha_{i+1}$ , else if  $DV_{i+1}$  was *up* then  $\alpha$  is decreased by the factor  $t$ . We use  $t = 2$  in our simulations. Thus depending on the feedback from the receivers and the value of  $DV$ , the value of  $\alpha$  is adjusted accordingly by each source of each group.

During the initialization of a new multicast session, the source node sends a small value of  $\alpha$  (e.g. 40 ms) and sets a binary direction vector  $DV$  as *up*.

Fig. 9 shows the evolution of the  $\delta$  value under one of the scenarios studied in Section 5; the  $\alpha$  value evolves similarly as  $\alpha$  is a constant times  $\delta$ . The results for other scenarios are similar. Fig. 9 shows that under our adaptive scheme the  $\delta$  parameter quickly converges to a range of three values between

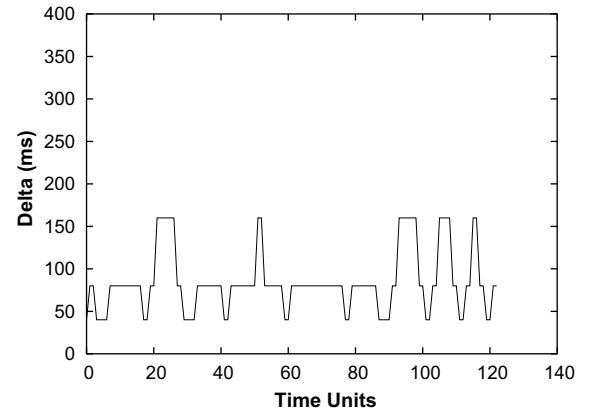


Fig. 9. The dynamics of the  $\delta$  value for one of the sources. The value of delta converges to a small set of values, i.e., between 40 and 160 ms.

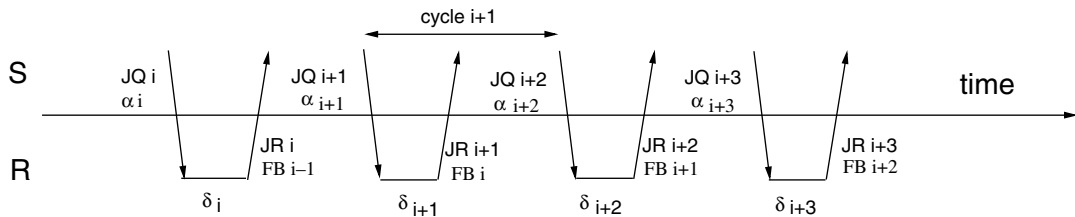


Fig. 8. The multicast source node  $S$  dynamically adjusts the  $\alpha$  value according to the feedbacks from group members  $R$ . There is a one-cycle delay in the feedbacks, as the  $(i + 2)$ th JOIN REPLY contains the feedback for the multicast mesh built based on the  $(i + 1)$ th  $\alpha/\delta$  values, i.e., for the mesh built based on the  $(i + 1)$ th JOIN REPLY packets from the members.

40 and 160 ms. We note that since our adaptive scheme continuously explores better  $\delta$  and  $\alpha$  values, after convergence, the  $\delta$  parameter will almost always oscillate among a range of three values (e.g., 40, 80, 160 ms). Further, there is a tradeoff between the range and the convergence time; by making the  $t$  factor smaller, the range will become smaller, but the time till convergence may be longer.

Finally, we note that our inline scheme can account for all forms of dynamics present in the static mesh network, such as changing link quality and traffic pattern, because it tries to adapt all the time, looking for new  $\alpha$  and  $\delta$  values at every cycle that would further improve the throughput. Effectively at every point of time, the adaptive scheme tries to search for a better value of  $\alpha$  till it converges to a set of optimal values. In the presence of dynamics such as a link-quality change or traffic change, if such change leads to a change in the number of packets received in a particular cycle, the online scheme would again start searching for a new set of optimal values of  $\alpha$  under the current conditions.

## 5. Simulation results

In this section, we present simulation results comparing the performance of ODMRP under different link-quality metrics.

### 5.1. Simulation setup

- (a) *Scenario.* We used the Glomosim [44] simulator in our simulation study. We simulated a network of 50 static nodes placed randomly in a 1000 m  $\times$  1000 m area. We used two multicast groups with nine receivers and one source each. The sources sent CBR traffic, consisting of 512-byte packets sent at a rate of 5 packets/s in the low-load case and 20 packets/s in a high-load case.<sup>1</sup> The radio propagation range was 250 m and the channel capacity was 2 Mbps (the data rate used for broadcast in 802.11 MAC protocol). The simulation duration was 400 s. The *TwoRay* propagation model was used. We initialized  $\alpha$  as 30 ms and  $\delta$  as 40 ms, and we observed that

in all the simulation scenarios, the values of  $\alpha$  and  $\delta$  quickly stabilized to a small set of fixed values. We used the Rayleigh fading model in our simulations, as it is appropriate for environments with many large reflectors, e.g. walls, trees, and buildings, where the sender and the receiver are not in Line-of-Sight of each other. We envision that such environments will be common for mesh networks. We simulated each protocol on 10 different randomly generated topologies and the result for each topology as well as the average over all topologies are presented.

- (b) *Evaluation metrics.* The following metrics are used to evaluate the various multicast protocol versions:

1. *Average throughput.* This metric measures the data packets (in bytes) delivered to a multicast group member per second averaged over all multicast group members.
2. *Probing overhead.* This metric measures the ratio of the total number of bytes of probe packets delivered to the MAC layer (for measuring link-quality metrics) over the total number of data bytes delivered to the MAC layer. For ODMRP, the probing overhead is zero, since ODMRP does not use any probe packets.
3. *Average end-to-end delay.* This metric measures the average delay of a data packet delivery (over all received packets at all receivers) which includes delays caused by queuing at the interfaces, propagation, and transfer times.

### 5.2. Results

In this section, we present the performance results of the various versions of ODMRP. We denote the original version of ODMRP as ODMRP, and the versions that incorporate PP, ETX, METX, ETT, and SPP as ODMRP\_PP, ODMRP\_ETX, ODMRP\_METX, ODMRP\_ETT, and ODMRP\_SPP, respectively. Unless otherwise stated, we show the results of ODMRP using various link-quality metrics normalized with respect to those of the original ODMRP.

#### 5.2.1. Throughput

Fig. 10 shows the relative throughput results for the different ODMRP versions under the low traffic

<sup>1</sup> Although the source sending rate is only 80 kbps, the actual load on the network is much higher as taking the node density into account, the total traffic load within a transmission range is on average 600 kbps.

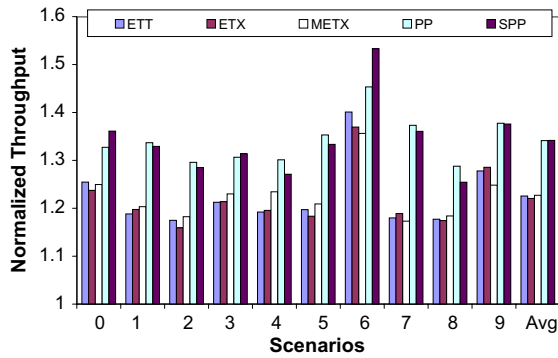


Fig. 10. Throughput comparison of different routing metrics using ODMRP under low load. On average the normalized throughput for all the metrics is higher than the case with high load.

load. It shows that ODMRP has the lowest throughput, ODMRP\_SPP and ODMRP\_PP have the highest throughput, and on average, ODMRP\_SPP, ODMRP\_PP, ODMRP\_METX, ODMRP\_ETX and ODMRP\_ETT achieve about 34%, 34%, 23%, 22%, and 22% higher throughputs than ODMRP, respectively.

Fig. 11 shows that the above relative performance of different versions persists under the high traffic load, in particular, ODMRP has the lowest throughput, ODMRP\_SPP and ODMRP\_PP have the highest throughput, and on average, ODMRP\_SPP, ODMRP\_PP, ODMRP\_METX, ODMRP\_ETX and ODMRP\_ETT achieve about 21%, 20%, 18%, 18%, and 17% higher throughputs than ODMRP, respectively. However, the percentage gains are lower than under the low traffic load

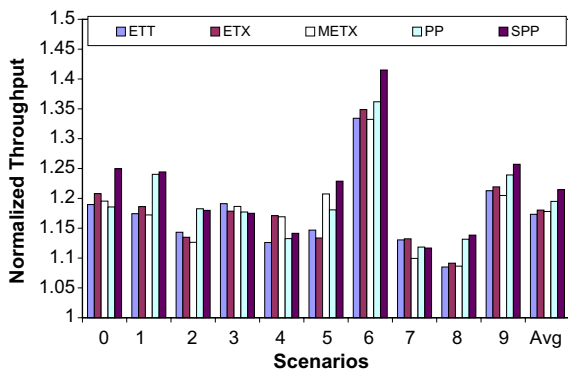


Fig. 11. Throughput comparison of different routing metrics using ODMRP under high load. On average ODMRP\_SPP and ODMRP\_PP achieve about 20% throughput gains over ODMRP.

because under the high traffic load, data traffic is more likely to interfere with probe packets. The effect of probing overhead is discussed later in Section 5.2.2.

ODMRP performs poorly because of fading. Fading is defined as a random change in the attenuation of a communications channel. Fading can directly affect the link quality. Every receiver has a *receive threshold*, which defines the signal strength below which the receiver cannot receive a signal properly. With fading, the signal strength may fluctuate up and down. This can cause a packet that would have been dropped to be received and vice versa. In particular, the quality of long links is adversely affected.

In ODMRP, the path from a source to a receiver is the one taken by the JOIN QUERY that reaches the receiver first, which is, in most cases (except when the JOIN QUERY along the shortest paths is lost), the shortest-hop path from a source to a destination which typically consists of long links. As fading causes long links to be lossy, ODMRP tends to choose low-throughput paths.

In contrast, all other ODMRP versions take into account the link quality in terms of loss rate, delay, or available bandwidth while picking paths, and therefore, they tend to pick paths with shorter links which achieve higher throughput.

Fig. 11 also shows that ODMRP\_ETX performs marginally better than ODMRP\_ETT although both of them take into account the loss characteristics of a link in a similar way (ETT uses ETX to estimate the loss rate). This is due to ODMRP\_ETT's high overhead of probe packets, as we later show in Section 5.2.2. To isolate the effect of probe packet overhead, we performed a simulation in which ODMRP\_ETX was run with ODMRP\_ETT's overhead, i.e., ODMRP\_ETX sends two back-to-back probe packets every 10 s as in ODMRP\_ETT but ignores the larger packet out of the two. We found ODMRP\_ETT performs similarly as ODMRP\_P\_ETX. Moreover, under low load, the impact of overhead is far less, and consequently ODMRP\_P\_ETT performs marginally better than ODMRP\_P\_ETX, as shown in Fig. 10. This confirms that ODMRP\_ETT's high overhead can offset its performance advantage achieved from better mesh construction.

Fig. 11 also shows that ODMRP\_PP achieves higher throughput than every other version except ODMRP\_SPP. This result is interesting because intuitively one would expect ODMRP\_PP to per-

form only as well as ODMRP\_ETT because they have the same (large) overhead and both of them take loss as well as delay into account (ETT incorporates delay information via bandwidth).

The reason is the way in which a packet loss is penalized in PP. PP puts a 20% penalty on the EWMA of the delay values. If a link is very lossy, the old EWMA dominates the component from the new measurement, the penalty is effectively incurred repeatedly on the EWMA, and the link cost grows as an exponential function of time. Such an exponential growth due to one bad link can cause the path cost to blow up. This property makes PP penalize bad links heavily and thus more likely to avoid them.

We analyze the dynamics of the link costs in PP. For simplicity, we assume the delay value when a packet is not lost to be a constant  $D$ . After each second, the ODMRP\_PP computes the delay value as an EWMA with a weight of  $W = 0.1$  to the new samples and  $1 - W = 0.9$  to the old samples. In case of a packet loss, a penalty of  $P = 0.2$  is imposed on the delay value. We denote the loss rate over that link as  $L$ , and the expected value of the delay metric after  $n$  seconds as  $d_n$ . At the  $n$ th second, if the probe packet is lost (with probability  $L$ ), we impose a penalty and hence the metric becomes  $(1 + P) * d_n$ . If the probe packet is received (with probability  $1 - L$ ), the new delay value becomes  $W * D + (1 - W)d_n$ . Thus the expected value after  $n + 1$  seconds is given by

$$d_{n+1} = L * (1 + P) * d_n + (1 - L)(W * D + (1 - W)d_n) \quad (10)$$

Fig. 12 plots the above equation with different loss rate values. It shows that for reasonably high loss rates, the value of the PP metric increases exponentially with time. Fig. 13 shows a hypothetical scenario in which the numbers show the loss rates of the links. Fig. 14 shows the values of the link-quality metric for the two paths. Clearly, one bad link causes the metric for the entire path  $A \rightarrow E \rightarrow D$  to grow exponentially. Because of this, ODMRP\_PP can avoid such paths whereas ODMRP\_ETX and ODMRP\_ETT cannot.

Fig. 11 also shows that among all the protocol versions, ODMRP\_SPP along with ODMRP\_PP achieves the highest throughput. On average, ODMRP\_SPP outperforms ODMRP\_METX, ODMRP\_ETX and ODMRP\_ETT by 3.7%, 3.5% and 4.5%, respectively. Being a product of probabilities, ODMRP\_SPP is more effective in avoiding

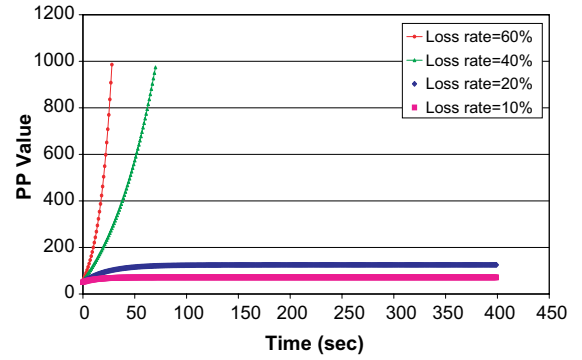


Fig. 12. A plot of Eq. (10) with different loss rates. PP penalty causes the link-quality metric value of a lossy link to grow exponentially with time.

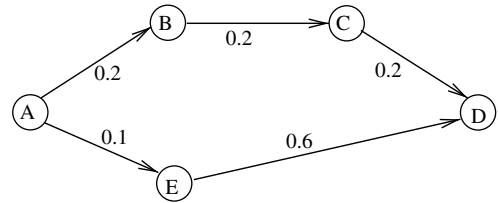


Fig. 13. An example scenario in which PP chooses longer and high-throughput paths. A high-loss rate of link  $E \rightarrow D$  causes the metric for the path  $A \rightarrow E \rightarrow D$  to grow exponentially.

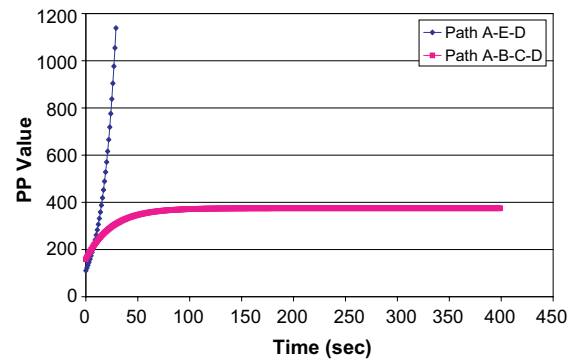


Fig. 14. The aggressive penalizing by PP causes the metric value of the entire path to blow up even if it contains one lossy link.

paths containing high-loss links than other protocols as one such link decreases the metric value of the entire path multiplicatively. It is for this reason that ODMRP\_SPP outperforms ODMRP\_ETX and ODMRP\_ETT, both of which take the sum of the link-quality metrics of the individual links constituting a path. Fig. 15 illustrates how ODMRP\_SPP is capable of choosing better throughput paths than ODMRP\_ETX using an



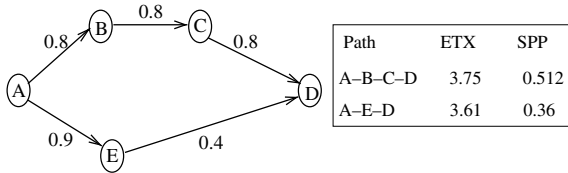


Fig. 15. SPP can choose longer and higher-throughput paths as opposed to ETX. A product of link-quality metrics makes the metric value of the entire path degrade even if one link is lossy. In ETX a low-loss link can compensate for a high-loss link.

example network. The network consists of two paths from A to D, namely  $A \rightarrow B \rightarrow C \rightarrow D$  and  $A \rightarrow E \rightarrow D$ . The first path is longer (3 hops) but it consists of low-loss links with each link having a probability of 0.8 of successfully transmitting a packet, while the second path is shorter (two hops) but consists of one very low- and one very high-loss link with success probabilities of 0.9 and 0.4, respectively. Clearly, the bad link  $E \rightarrow D$  makes the SPP metric for the path  $A \rightarrow E \rightarrow D$  very small. The value of SPP for the first path is 0.512 and for the second path is 0.36, and SPP selects the longer (high-throughput) path. The ETX values for the two paths are 3.75 and 3.61, respectively, and ETX selects the shorter (low throughput) path. Hence, ODMRP\_SPP can avoid selecting the path with even one high-loss link while ODMRP\_ETX and ODMRP\_ETT cannot.

Finally, ODMRP\_METX outperforms ODMRP\_PP and ODMRP\_ETX because it is more aggressive in avoiding lossy links and unlike ETX and ETT, METX takes into account the unreliability of the link layer while calculating the expected number of transmissions. But, it is less aggressive than ODMRP\_PP and ODMRP\_SPP and hence the difference in performance. In summary, ODMRP\_SPP and ODMRP\_PP achieve higher throughputs by heavily penalizing lossy links and thereby avoiding them. ODMRP\_ETX and ODMRP\_ETT also penalize lossy links but they are less aggressive in doing so and therefore not as effective. ODMRP\_METX is a hybrid of ETX and SPP and hence its performance lies between those two. ODMRP does not consider any link characteristics and tends to choose short paths consisting of long links that are lossy, and hence it performs poorly in terms of throughput.

### 5.2.2. Probing overhead

In this section, we compare the probing overhead of various protocol versions that use link-quality

metrics. Since under all metrics, probe packets are sent periodically, we can estimate the total number of bytes of the probe packets sent during the 400-s duration of the simulation. ODMRP\_ETX and ODMRP\_SPP send one 160-byte probe packet every 5 s, for a total of 80 packets per node. ODMRP\_PP and ODMRP\_ETT send one small 160-byte probe packet and one large 1160-byte packet every 10 s, for a total of 40 small and 40 large packets per node. Since we have 50 nodes in the network, the total number of bytes in the network from probe packets for ODMRP\_ETX and ODMRP\_SPP is  $50 \times 80 \times 160 = 640,000$  bytes and for ODMRP\_PP and ODMRP\_ETT is  $50 \times 40 \times (160 + 1160) = 2,640,000$  bytes. Fig. 16 shows the percentage of bytes from probe packets out of the total number of data bytes. We observe that ODMRP\_PP and ODMRP\_ETT have about 4.5% higher probing overhead than ODMRP\_ETX, ODMRP\_METX and ODMRP\_SPP. This has two implications. First, although ODMRP\_ETX and ODMRP\_ETT have similar ways of estimating the link loss rates, the latter will have lower throughput values. Second, the overhead affects the relative end-to-end delay which will be discussed in Section 5.2.3.

Note that although the number of bytes from probe packets in ODMRP\_ETX and ODMRP\_METX is the same as in ODMRP\_SPP, the percentage overhead for ODMRP\_SPP is slightly lower than for ODMRP\_ETX and ODMRP\_METX. The same observation can be made for ODMRP\_PP and ODMRP\_ETT. This is because ODMRP\_SPP and ODMRP\_PP achieve higher

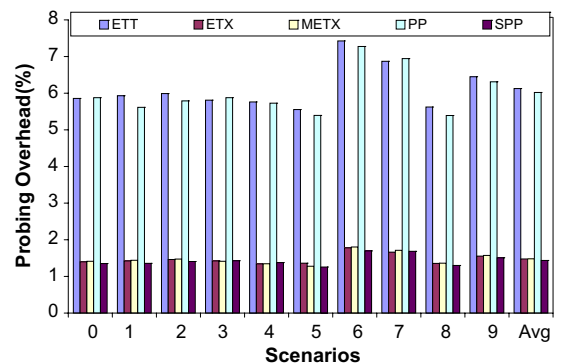


Fig. 16. Comparison of the probing overhead of different routing metrics. ODMRP\_PP, and ODMRP\_ETT have much higher probing overhead than ODMRP\_ETX, ODMRP\_ETX, and ODMRP\_SPP.

throughput than ODMRP\_ETX, ODMRP\_METX, and ODMRP\_ETT. Higher throughput implies more bytes of data packets are transmitted throughout the network, and hence the lower percentage of bytes from probing packets.

There is a tradeoff between the probing overhead and the throughput achieved. Higher probing rate implies more recent information about the network condition and hence more informed decision making. However, probing itself can be a source of interference to the data traffic and cause loss in throughput. Thus choosing the correct probing rate is crucial. To underline the importance of choosing the probing rate carefully, in Fig. 17 we show the throughput gains for all versions using link-quality metrics when the probing rate is increased by five times. Compared to Fig. 11, we see that the throughputs of all the metrics drop by about 2–3%. We also conducted simulations with a probing rate 10 times lower (the results are not shown due to page limitation), and found the throughput gains are improved by around 3%. These results suggest that the probing rate indeed affects the throughput gains achieved. Finding an optimal probing rate is part of our future work. These results also indicate that high overhead metrics such as PP and ETT are more sensitive to the probing rate than ETX, METX, or SPP as these metrics incur much higher probing overhead than the others.

### 5.2.3. Delay

Besides throughput and probing overhead, end-to-end delay is another aspect that we considered in our simulation results. We measured the normal-

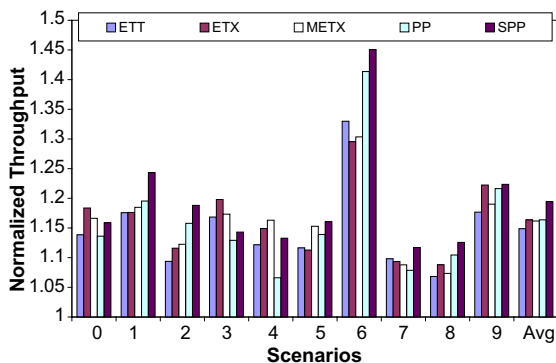


Fig. 17. Throughput comparison of different routing metrics using ODMRP under high load with high probing rate. The high overhead metrics, namely ETT and PP, suffer due to increase in probing rate. The rest of the low probing overhead metrics perform similarly.

ized average end-to-end delay for ODMRP under each of the metrics with respect to ODMRP. The results, shown in Fig. 18, show that in most cases, ODMRP\_SPP and ODMRP\_ETX achieve lower end-to-end delays than the rest ODMRP versions. This is because ODMRP\_SPP and ODMRP\_ETX have very low probing overhead which reduces the delay at each hop, because each node faces less contention for the channel. This is also the reason why ODMRP\_ETX and ODMRP\_SPP achieve lower delay than ODMRP\_ETT despite that ETT takes into account delay (the available bandwidth incorporates delay information). Due to similar reasons, ODMRP\_PP is also outperformed by ODMRP\_ETX and ODMRP\_SPP, in terms of delay. Besides lower probing overhead, smaller end-to-end delay is another advantage that ODMRP\_SPP has over ODMRP\_PP.

### 5.3. Impact of multiple sources

In the previous section, each group has one source. Owing to the way the ODMRP builds the routing structure, one source per group gives a tree structure. When there are multiples sources per group, a mesh structure is formed, as ODMRP creates forwarding group members per group and not per source. As a result, a node that is made a forwarder as consequence of a JOIN QUERY sent by one source may also act as a forwarder for the packets from some other source. This creates path redundancy in data delivery in a mesh structure. In this section, we investigate the interactions between the path redundancy in a mesh structure and the various routing metrics. The scenarios used are the same as in Section 5.1, except one difference: instead of

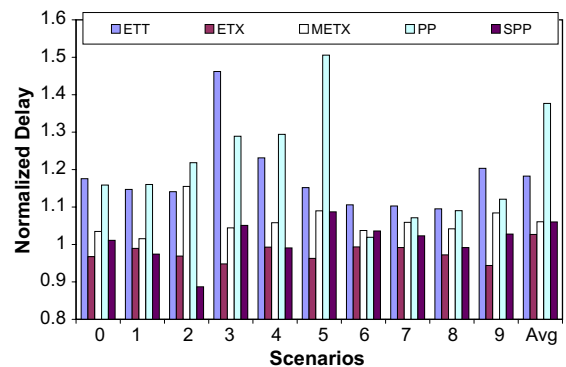


Fig. 18. Delay comparison of different routing metrics. ODMRP\_PP and ODMRP\_ETT have the highest delay.

having two multicast groups with one source each, a single multicast group with two sources is simulated.

Fig. 19 shows the relative throughput performance of each routing metric under the low traffic load. ODMRP\_SPP, ODMRP\_PP, ODMRP\_METX, ODMRP\_ETX, ODMRP\_ETT outperform ODMRP by 12%, 11.5%, 8%, 8% and 7%, respectively. However, compared to the single source per group case depicted in Fig. 10, the relative throughput gain is reduced by around 15% for the different metrics.

To investigate the reasons for such a reduction in throughput gain under a mesh structure, we measured the number of forwarding nodes (FGs) in the network with different numbers of sources per group. Fig. 20 shows that the number of FGs increases monotonically with the number of sources per group, using the original ODMRP as the routing protocol. Fig. 21 shows that a higher redundancy in data delivery obtained from the increased number of FGs leads to an improved PDR. In this case, the original ODMRP's inability to choose high-throughput paths is compensated by the increased redundancy of data delivery paths. The improved PDR from the increased path redundancy reduces the window of opportunity for throughput improvement from using high-throughput routing metrics. This explains why the percentage gain of high-throughput metrics diminishes as the number of sources increases from one (Fig. 10) to two (Fig. 19).

We note that previously [29,10,23], mesh-based and tree-based protocols have been compared in mobile ad hoc networks and it was shown that under mobility the former outperforms the latter as the availability of alternative paths provides

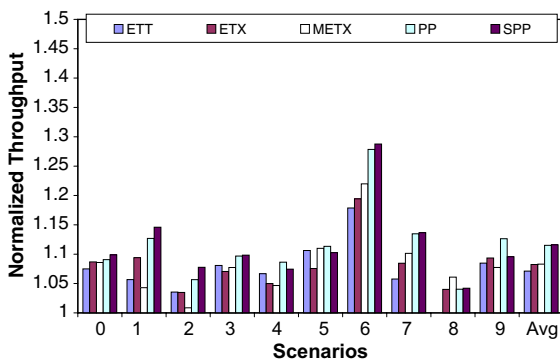


Fig. 19. Normalized throughput for different high-throughput metrics under different scenarios with two sources per group under low load.

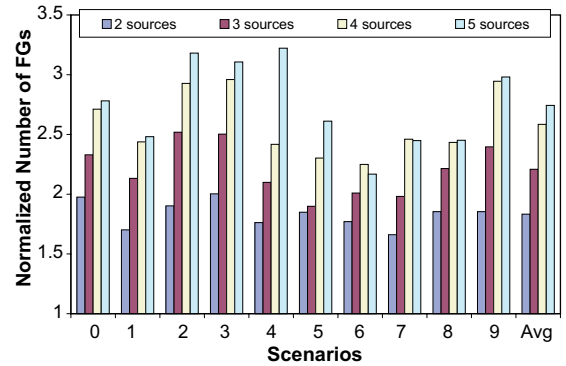


Fig. 20. The average number of forwarding nodes in the network normalized under ten different scenarios with different number of sources when original ODMRP is used. All the values are normalized with respect to the case with only one source per group.

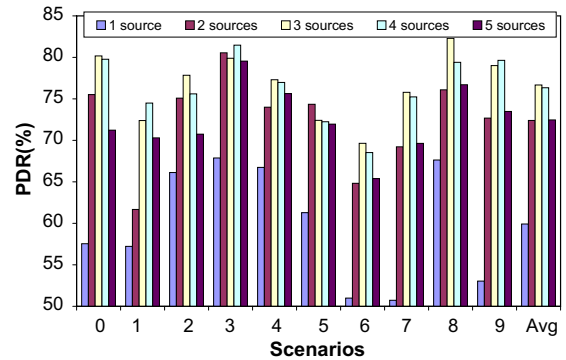


Fig. 21. The packet delivery ratio (PDR) of ODMRP with different number of sources per group. As the number of sources increases, the PDR increases due to increased path redundancy. At a very large number of nodes, PDR drops because of a too high load that causes packet drops due to collision and congestion.

robustness to mobility. In this paper we show the throughput advantage of mesh-based protocols exists even without mobility as the path redundancy also provides robustness to loss characteristics of wireless links.

Finally, compared to Figs. 11 and 22 shows that a reduction of throughput improvement happens also under the high load when the number of sources is increased from one to two per group for the same reasons discussed above.

To summarize, high-throughput link-quality metrics can not give much gain in case there is a lot of path redundancy in the network because such a network already achieves a high PDR and hence the scope for improvement is small. However, this

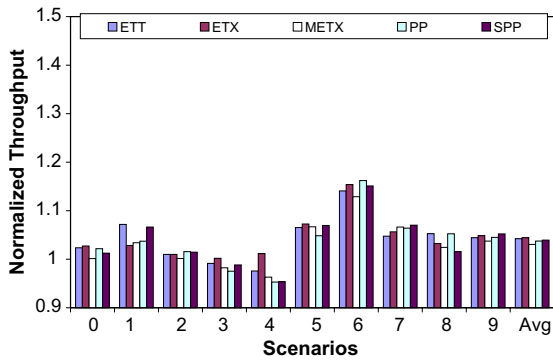


Fig. 22. Normalized throughput for different high-throughput metrics under different scenarios with two sources per group under high load. ODMRP\_SPP, ODMRP\_PP, ODMRP\_METX, ODMRP\_ETX, ODMRP\_ETT outperforms ODMRP only by a small margin of around 5%.

does not undermine the importance of high-throughput metrics for several reasons. First, such metrics continue to be effective in multicast protocols that are tree-based such as MAODV [38]. Second, when the network is large and the number of sources is comparatively small, i.e., the number of sources per group may not be high enough to create enough path redundancy, high-throughput metrics can still significantly improve the throughput. Third, higher path redundancy may lead to more and sometimes unnecessary data traffic in the network.

## 6. Testbed experiments

To verify the effectiveness of the high-throughput link-quality metrics for multicast observed in our simulation study, we performed experiments on an 8-node wireless mesh network testbed. Specifically,

we evaluated the performance of ODMRP using all the different routing metrics in comparison to the original ODMRP using a real implementation on this testbed.

### 6.1. Setup

Our testbed consists of eight wireless mesh routers (small form factor desktops) with Intel Pentium 4 processors spread out over a typical academic building floor of length 240 feet and width 86 feet, approximately. Each mesh router is equipped with a single Atheros 5212 802.11b wireless card. Each radio is attached to a 2 dBi rubber duck omnidirectional antenna with a low loss pigtail to provide flexibility in antenna placement. Each mesh router runs Linux kernel 2.4.20-8 and the open-source *hostap* drivers are used to enable the wireless cards. The IP addresses are statically assigned. The wireless cards we use can support a wide range of power settings (0–18 dbm). We used them in their default operational mode.

The nodes are statically placed in the offices on the second floor of an office building on the Purdue campus, as shown in Fig. 23. The testbed deployment environment is not wireless friendly, having floor-to-ceiling office walls instead of cubicles, as well as some laboratories with structures that limit the propagation of wireless signals. Apart from structural impediments, interference exists in our deployment from other 802.11b networks.

### 6.2. Protocol implementation

We implemented our own version of the original ODMRP and enhanced it with the different

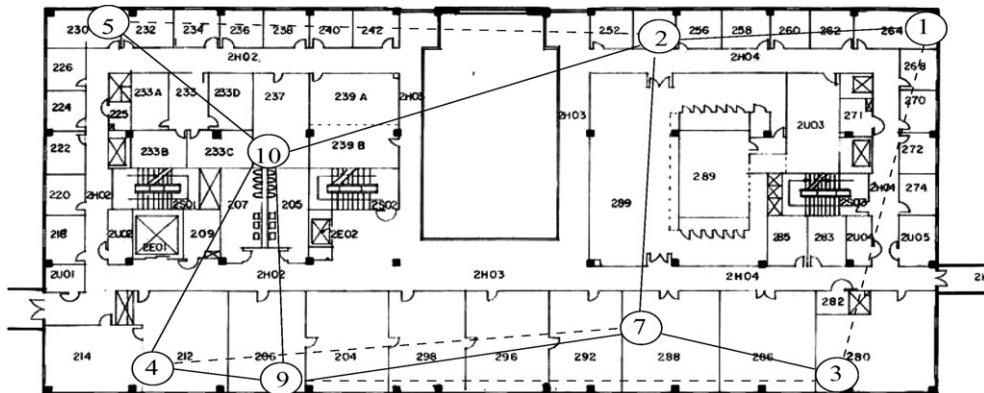


Fig. 23. The floor map of our eight-node mesh network testbed deployed in an office building. The lines show the links with connectivities in the network. Solid lines denote low-loss links and dashed ones denote lossy links.

link-quality metrics. We were unable to obtain the only known implementation of ODMRP [3]. In addition, the previous implementation has been developed for a much older Linux kernel (v2.0) and would have incurred portability issues in our testbed. Different from the implementation in [3], we chose to implement ODMRP as an application layer daemon *odmrpd* for ease of debugging, deployment and use. Similar to our approach, many unicast protocols are currently being developed or have been developed [30,7,34] as user-level daemons with loadable kernel modules for packet capturing and routing. Our implementation is based on the ODMRP specification in [27].

Fig. 24 shows the functioning of the *odmrpd*. *odmrpd* captures IP packets with multicast addresses using the Linux NetFilter mechanism and uses these addresses as group IDs. It then uses UDP broadcast to propagate each JOIN QUERY packet through out the network. JOIN REPLY packets are similarly propagated using UDP broadcast. Once the forwarding group for a multicast group is formed, each data packet for that multicast group is propagated via the corresponding forwarding group by the *odmrpd* at each hop. Each node that wishes to receive packets for a multicast group opens a socket to receive data on the multicast address for that multicast group. The *odmrpd* at each node can deliver data packets for all multicast addresses to the applications running on the node.

### 6.3. Results

Fig. 23 shows the links with connectivities in our testbed. Note that in this case, the link quality and the link distances do not directly correspond. The link quality mainly depends on the obstacles present, such as walls and metallic objects. In order to

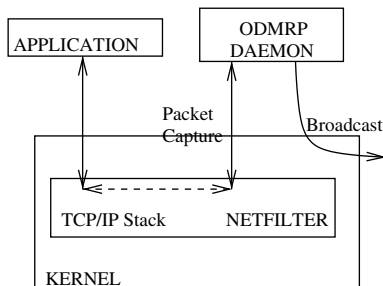


Fig. 24. The architecture of our *odmrpd* implementation. The multicast packets from and to the application are intercepted by *odmrpd* using the NetFilter mechanism.

get an estimate of the link quality, we transferred a series of ping messages between each pair of nodes. The number of packets lost during the ping exchange gave us an idea of the quality of the link. Based on the results obtained using ping messages, we qualitatively classify each of the links as low-loss or lossy. The dashed lines show the links which are lossy and the solid lines show links that have low or almost no loss. The pair of nodes between which there are no lines are not able to communicate with each other. We do not show any numerical values of loss rates of the links because these values change fairly quickly.

We performed our multicast experiments with two multicast groups, each having one source and two receivers. The first multicast group had node 2 as the source and nodes 3 and 5 as the receivers, and the second group had node 4 as the source and nodes 1 and 7 as the receivers. The rest of the nodes acted only as forwarding nodes. Each source sent CBR traffic at a rate of 20 packets/s, each of size 512 bytes. The experiments were run for 400 s. The same experiment was run five times to make the results resilient to random changes in the environment.

Fig. 25 shows the throughput obtained by all the metrics normalized with respect to the throughput obtained by the original ODMRP, averaged over all receivers. ODMRP\_SPP, ODMRP\_METX, ODMRP\_ETX, and ODMRP\_ETT achieve gains of around 14%, 7.5%, 8% and 7%, respectively. Somewhat surprisingly, ODMRP\_PP achieves on average a 17.5% gain, 3.5% higher than that of ODMRP\_SPP. Such a gain is not seen in simulations because of the following reason. Figs. 12 and

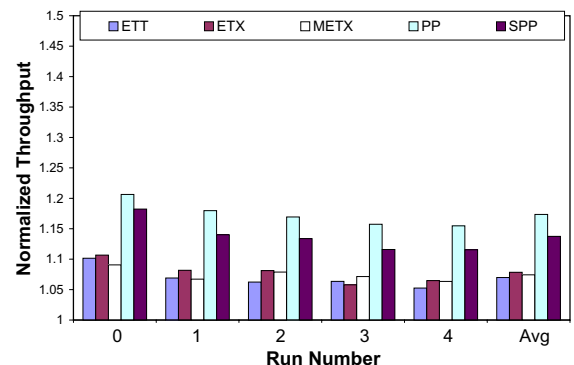


Fig. 25. Throughput performance of different routing metrics on a real-mesh testbed. ODMRP using any link-quality metric outperforms ODMRP by significant margins.



14 show that if the loss rate of any one of the links constituting a path is high, PP can cause the cost of a path to blow up exponentially, but for moderately low loss rates, the cost stabilizes to a constant value. In the testbed scenario, all the dashed links have loss rates in the range of 40–60%, which are higher than seen in the simulations. Consequently, PP causes the cost of paths using such links to go up very fast and once the cost explodes, any path containing such links is never chosen in the future. On the other hand, SPP, ETX, ETT and METX penalize such links during some of the route request phases. However, when such links become relatively less lossy due to random temporal variations, those metrics are misled into choosing them and consequently suffer throughput losses.

Independent of the above observations, the reason that ODMRP\_PP, ODMRP\_SPP, ODMRP\_METX, ODMRP\_ETX and ODMRP\_ETT achieve throughput gains over ODMRP (though with varying amounts) can be explained by the difference between the multicast trees constructed by ODMRP and ODMRP using the various routing metrics. We use ODMRP\_PP as an example for further illustration. Fig. 26 shows the paths taken by ODMRP versus those by ODMRP\_PP. The solid and dashed arrows denote the heavily used links for ODMRP\_PP and ODMRP, respectively. For the sake of clarity, we removed the floor map from the background and kept only the node positions in the figure. First we discuss about the paths to receivers 5 and 7. ODMRP chooses the one-hop path from node 2 to 5 which is lossy (see Fig. 23). Similarly, node 4 chooses a one-hop path to 7 which is lossy. In contrast, ODMRP\_PP is able to avoid the lossy links, and chooses relatively longer but higher-throughput paths. For example, node 2

reaches 5 along a two-hop path, via 10. Similarly, node 4 reaches 7 along a two-hop path, via 9. For receivers 1 and 3, sources 2 and 4 have more than one paths. Node 2 can reach 3 via 7 or 1; similarly, node 4 can reach 1 via 10 and 2, or 7 and 2, or 7 and 3, or 9 and 3. But ODMRP can not distinguish between the various alternative paths and often chooses the lossy path containing the link from 1 to 3, or 4 to 7, or 9 to 3. ODMRP\_PP is again able to figure out the lossy links and avoid them.

## 7. Related work

There is a large body of work comparing the performance of various ad hoc routing protocols (for example, [5,9,20]). Most of these protocols (for example, [35,36,21]) assume minimum-hop-count as the routing metric and focus on scenarios that involve significant node mobility.

The numerous link-quality metrics that have been proposed for unicast routing in stationary mesh networks have already been discussed in Section 2. Several works have compared the relative performance gain in using these metrics. In [8], the authors propose the expected transmission count metric (ETX) and illustrate the reasons of poor performance of the minimum-hop-count metric. They also modify DSR and DSDV to incorporate ETX. Measurements on a static wireless testbed using the modified versions of DSR and DSDV show that ETX achieves throughput improvement by a factor of two or more over the traditional minimum-hop-count metric. In [12], the authors conduct an evaluation of the performance of three link-quality metrics – ETX, per-hop RTT, and per-hop packet pair – and compare them against the minimum-hop-count metric. They implemented a modified version of DSR as a loadable Windows driver to conduct experiments on an ad hoc wireless testbed. The measurements indicate that ETX metric has the best throughput performance when the nodes are static. RTT and Packet Pair perform poorly because they suffer from self-interference. Interestingly however, the hop-count metric outperforms all of the link-quality metrics in a mobile scenario as it can react quickly to fast topology changes. In [13], the authors propose Weighted Cumulative Expected Transmission Time metric (WCETT), and based on experiments on a static wireless testbed show that WCETT significantly outperforms ETX and shortest-hop metric by efficiently using multiple radios tuned to non-overlapping channels. Finally,

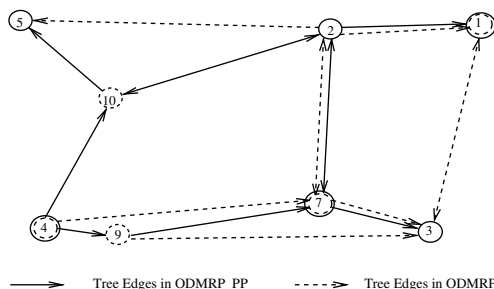


Fig. 26. The trees constructed by ODMRP and ODMRP\_SPP. The dashed circles denote nodes that do not belong to any group. The solid and concentric circles denote nodes of two different multicast groups.

in [4,11], the authors propose algorithms and routing metrics to find minimum-energy paths. They consider the case when there is no reliability at the link layer as well as the case when there is partial reliability at the link layer. The former case is related to this paper as most of the multicast protocols use unreliable broadcast at the link layer. They show via simulations that considering the lossy behavior of wireless links and the retransmissions done at the link layer gives 30–70% energy gains over existing energy-efficient routing schemes. In [24], an energy-efficient routing scheme is proposed that takes into account the interference created by existing flows in the network. The authors prove that under certain circumstances their algorithm is asymptotically optimal in time in terms of minimizing the average energy consumption. In [2], a high-throughput metric median time metric (MTM) for route selection under multiple data rates was proposed. Simulation studies show that MTM can achieve a throughput gain of 20–60% depending on the network density. Finally, in [15], the authors advocate the use of an energy-efficient routing metric that is based on the geometric mean, as opposed to the arithmetic mean, of the residual energies of the individual nodes constituting a path. They also propose a method similar to ours to incorporate this metric in a routing protocol, with the difference being that intermediate nodes forward a query only after some query aggregation interval, and they derive the optimal query aggregation interval for each intermediate node for a simple linear network.

In addition to ODMRP, a large number of multicast protocols have been designed to efficiently maintain a distributed multicast routing structure in dynamically changing topologies in MANETs. These include traditional tree- or mesh-based protocols such as MAODV [38], ADMR [18], ODMRP [26], CAMP [32], overlay-based protocols such as AMRoute [43], PAST-DM [16], and back-bone-based and hybrid protocols such as MCEDAR [39] and PUMA [40], and more recent stateless protocols such as DDM [19], HDDM [17], and RDG [31].

There is a significant amount of work comparing the performance of various multicast protocols for ad hoc networks (for example, [29,10,23,41]). As is the case for unicast, most of these protocols also follow minimum-hop-count as the routing metric and focus on scenarios with high mobility. Studies show that in such scenarios the mesh-based protocols outperform tree-based protocols, specifically under

high mobility. This is because in mesh-based protocols the availability of alternative paths provides robustness to mobility. However, mesh-based protocols suffer from high overhead that causes degradation in performance under high load. Moreover, such protocols do not scale as the group size increases. Among all these protocols, ODMRP has been shown to be effective in most cases, although its overhead rapidly increases as the number of senders increases. Stateless protocols alleviate the problem of maintaining the distributed delivery tree or mesh but they assume that the underlying routing layer will take care of forwarding the packets to the correct destinations. Moreover, they are efficient only for small group multicast. Despite the abundance of work on designing multicast routing protocols, there has been no study of high-throughput link-quality-based routing metrics for the tree or mesh construction in such protocols.

To the best of our knowledge, our work is the first to study the impact of different link-quality routing metrics on the performance of multicast routing protocols. In this paper, we have chosen ODMRP as a representative multicast protocol in our study of link-quality metrics. We expect most of the findings to be applicable to any other broadcast-based multicast protocol.

## 8. Conclusions and future work

In this paper, we have studied the link-quality routing metrics for high-throughput multicast in mesh networks. We first discussed the fundamental difference between unicast and multicast routing in how data packets are transmitted at the link layer, and then showed accordingly how to adapt routing metrics for unicast routing to be used in multicast routing. We studied the performance of different metrics via extensive simulation and experiments on a mesh network testbed, using ODMRP as a representative multicast protocol.

Our studies have shown that ODMRP equipped with any of the link-quality-based routing metrics can achieve higher throughput than the original ODMRP. We also found that heavily penalizing lossy links is an effective way to avoid low-throughput paths and SPP and PP achieve the highest throughput performance because of their aggressive manner of penalizing lossy links. Moreover, SPP has much less overhead than PP, which reduces the end-to-end delay. We have also observed a tradeoff between throughput gains achieved and

the probing overhead incurred, i.e., higher probing rate gives more recent information about the network but also causes interference for data packets. As our future work, we plan to investigate the optimal probing rate required. We also showed how path redundancy due to mesh creation provides another degree of resilience to lossy links and reduces the throughput improvement from using high-throughput link-quality metrics. Finally, our experimental results on an eight-node mesh network testbed validate the results obtained in the simulation study.

In this paper, we have considered a mesh network equipped with single-radio nodes transmitting on a single-channel at a fixed rate. There have been several efforts on improving the throughput of mesh networks [37,14,25] using multiple radios that utilize multiple channels (available in the IEEE 802.11a/b/g standards) to separate the contending transmissions in the frequency domain. In our future work, we plan to extend the high-throughput link-quality metrics studied in this paper for multicast routing in multiple-radio/multiple-channel mesh networks. We also plan to significantly expand our testbed which will give more diversity in the network topologies.

## Acknowledgement

This work was supported in part by NSF Grant ANI-0338856 and CNS-0626703.

## References

- [1] A. Adya, P. Bahl, J. Padhye, A. Wolman, L. Zhou, A multi-radio unification protocol for IEEE 802.11 wireless networks, in: *BroadNets*, July 2004.
- [2] B. Awerbuch, D. Holmer, H. Rubens, The medium time metric: high throughput route selection in multi-rate ad hoc wireless networks, *J. Mob. Netw. Appl.* 11 (2) (2006).
- [3] S.H. Bae, S.-J. Lee, W. Su, M. Gerla, The design, implementation, and performance evaluation of on-demand multicast routing protocol in multihop wireless networks, *IEEE Netw.* 14 (2000) 70–77.
- [4] S. Banerjee, A. Misra, Minimum energy paths for reliable communication in multi-hop wireless networks, in: *Proceedings of ACM MobiHoc*, 2002.
- [5] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: *Proceedings of ACM MobiCom*, October 1998.
- [6] C.-C. Chiang, M. Gerla, L. Zhang, Forwarding group multicast protocol (FGMP) for multihop mobile wireless networks, *Cluster Comput.* 1 (2) (1998) 187–196.
- [7] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot, Optimized link state routing protocol (OLSR), RFC 3626, October 2003.
- [8] D.S.J.D. Couto, D. Aguayo, J.C. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, in: *Proceedings of ACM MobiCom*, 2003.
- [9] S.R. Das, C.E. Perkins, E.M. Royer, Performance comparison of two on-demand routing protocols for ad hoc networks, in: *Proceedings of IEEE INFOCOM*, March 2000.
- [10] C. de Morais Cordeiro, H. Gossain, D.P. Agrawal, Multicast over wireless mobile ad hoc networks: present and future directions, *IEEE Netw.* 17 (1) (2003).
- [11] Q. Dong, S. Banerjee, M. Adler, A. Misra, Minimum energy reliable paths using unreliable wireless links, in: *Proceedings of ACM MobiHoc*, 2005.
- [12] R. Draves, J. Padhye, B. Zill, Comparison of routing metrics for static multi-hop wireless networks, in: *Proceedings of ACM SIGCOMM*, 2004.
- [13] R. Draves, J. Padhye, B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, in: *Proceedings of ACM MobiCom*, 2004.
- [14] R. Draves, J. Padhye, B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, in: *Proceedings of ACM MobiCom*, September 2004.
- [15] S. Gabriel, D. Mosse, R. Melhem, Mitigating the flooding waves problem in energy efficient routing for MANETs, in: *Proceedings of ICDCS*, 2006.
- [16] C. Gui, P. Mohapatra, Efficient overlay multicast for mobile ad hoc networks, in: *Proceedings of IEEE WCNC*, March 2003.
- [17] C. Gui, P. Mohapatra, Scalable multicasting for mobile ad hoc networks, in: *Proceedings of IEEE INFOCOM*, March 2004.
- [18] J.G. Jetcheva, D.B. Johnson, Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks, in: *Proceedings of ACM MobiHoc*, October 2001.
- [19] L. Ji, S. Corson, Differential destination multicast – a MANET multicast routing protocol for small groups, in: *Proc. of IEEE INFOCOM*, April 2001.
- [20] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, M. Degermark, Scenario-based performance analysis of routing protocols for mobile ad hoc networks, in: *Proceedings of ACM MobiCom*, August 1999.
- [21] D.B. Johnson, D.A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*, Kluwer Academic, 1996.
- [22] S. Keshav, A control-theoretic approach to flow control, in: *Proceedings of the Conference on Communications Architecture and Protocols*, 1993, pp. 3–15.
- [23] T. Kunz, E. Cheng, On-demand multicasting in ad hoc networks: comparing AODV and ODMRP, in: *Proceedings of IEEE ICDCS*, July 2002.
- [24] S. Kwon, N. Shroff, Energy-efficient interference-based routing for multi-hop wireless networks, in: *Proceedings of IEEE INFOCOM*, 2006.
- [25] P. Kyasanur, N.H. Vaidya, Capacity of multi-channel wireless networks: impact of number of channels and interfaces, in: *MobiCom'05: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, 2005, pp. 43–57.
- [26] S.-J. Lee, M. Gerla, C.-C. Chiang, On-demand multicast routing protocol, in: *Proceedings of IEEE WCNC*, September 1999.

- [27] S.-J. Lee, W. Su, M. Gerla, On-demand multicast routing protocol (ODMRP) for ad hoc networks, January 2000. <http://www.ietf.org/internet-drafts/draft-ietf-manet-odmrp-02.txt>, IETF Internet Draft.
- [28] S.J. Lee, W. Su, M. Gerla, On-demand multicast routing protocol in multihop wireless mobile networks, *Mob. Netw. Appl.* 7 (6) (2002).
- [29] S.-J. Lee, W. Su, J. Hsu, M. Gerla, R. Bagrodia, A performance comparison study of ad hoc wireless multicast protocols, in: *Proceedings of IEEE INFOCOM*, March 2000.
- [30] H. Lundgren, E. Nordstrom, AODV implementation code (Uppsala University), 2004. <http://core.it.uu.se/AdHoc/AodvUUIImpl>.
- [31] J. Luo, P.T. Eugster, J.-P. Hubaux, Route driven gossip: probabilistic reliable multicast in ad hoc networks, in: *Proceedings of IEEE INFOCOM*, March 2003.
- [32] E.L. Madruga, J.J. Garcia-Luna-Aceves, Scalable multicasting: the core-assisted mesh protocol, *ACM/Baltzer Mob. Netw. Appl.* 6 (2) (2001) 51–165, Special Issue on Management of Mobility.
- [33] T. Nadeem, A. Agrawala, S. Banerjee, Energy-efficient reliable paths for on-demand routing protocols, in: *Proceedings of IEEE MWCN*, 2004.
- [34] E. Nordstrom, DSR implementation code (Uppsala University), 2005. <http://core.it.uu.se/AdHoc/DsrUUIImpl>.
- [35] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in: *Proceedings of ACM SIGCOMM*, August 1994.
- [36] C.E. Perkins, E.M. Royer, Ad hoc on-demand distance vector routing, in: *Proceedings of IEEE WMCSA*, February 1999.
- [37] A. Raniwala, T. Chiueh, Architectures and algorithms for an IEEE 802.11-based multi-channel wireless mesh network, in: *Proceedings of IEEE INFOCOM*, March 2005.
- [38] E.M. Royer, C.E. Perkins, Multicast operation of the ad hoc on-demand distance vector routing protocol, in: *Proceedings of MobiCom*, August 1999.
- [39] P. Sinha, R. Sivakumar, V. Bharghavan, MCEDAR: Multicast core-extraction distributed ad hoc routing, in: *Proceedings of IEEE WCNC*, September 1999.
- [40] R. Vaishampayan, J.J. Garcia-Luna-Aceves, Efficient and robust multicast routing in mobile ad hoc networks, in: *Proceedings of MASS 2004*, September 2004.
- [41] U. Varshney, Multicast over wireless networks, *Commun. ACM* 45 (12) (2002) 31–37.
- [42] C.W. Wu, Y.C. Tay, AMRIS: A multicast protocol for ad hoc wireless networks, in: *IEEE MILCOM*, November 1999.
- [43] J. Xie, R.R. Talpade, A. Mcauley, M. Liu, AMRoute: ad hoc multicast routing protocol, *Mob. Netw. Appl.* 7 (6) (2002) 429–439.
- [44] X. Zeng, R. Bagrodia, M. Gerla, Glomosim: A library for parallel simulation of large-scale wireless networks, in: *Proceedings of PADS Workshop*, May 1998.
- [45] MIT Roofnet, <http://www.pdos.lcs.mit.edu/roofnet>.
- [46] Seattle wireless, <http://www.seattlewireless.net>.
- [47] Bay area wireless users group, <http://www.bawug.org>.
- [48] Mesh Networks Inc., mesh networks technology overview, <http://www.meshnetworks.com>.
- [49] Radiant networks, <http://www.radiantnetworks.com>.



Purdue University in 2005.

**Sabyasachi Roy** is currently a Ph.D. candidate in the School of Electrical and Computer Engineering at Purdue University, USA. Previously, he received a B.Tech. degree from IIT Kanpur, India. His research interests include wireless networks, peer-to-peer networks and distributed systems. He is a recipient of the Tellabs Fellowship by Center for Wireless Systems and Applications (CWSA), Department of ECE,



for Wireless Systems and Applications (CWSA), Purdue University, 2006.

**Dimitrios Koutsonikolas** is currently a PhD student in the School of Electrical and Computer Engineering at Purdue University, USA. Previously, he received a B.Engg. degree from the National Technical University of Athens (NTUA), Greece. His research interests include wireless ad hoc, mesh and sensor networks. He is a recipient of the Ross Fellowship, Purdue University, 2004, and a Tellabs Fellowship, Center



and mobile robotics.

**Saumitra M. Das** is currently a Ph.D. candidate in the School of Electrical and Computer Engineering at Purdue University, USA. Previously, he received a MS degree from Carnegie Mellon University, USA and a B.Engg. degree from the University of Bombay, India. His research interests include cross-layer system design for multi-hop wireless networks, scalable routing strategies in wireless ad hoc networks,



and parallel computing. He has published over 100 papers in these areas. He received the NSF CAREER Award in 2003. He served as a TPC Vice Chair for ICDCS 2007 and the 2004 International Conference on Parallel Processing and a co-founder and TPC co-chair for the International Workshop on Mobile Peer-to-Peer Computing. He is a member of USENIX and a senior member of ACM and IEEE.

**Y. Charlie Hu** is an Associate Professor of Electrical and Computer Engineering and Computer Science at Purdue University. He received his M.S. and M.Phil. degrees from Yale University in 1992 and his Ph.D. degree in Computer Science from Harvard University in 1997. From 1997 to 2001, he was a research scientist at Rice University. His research interests include operating systems, distributed systems, wireless networking,