High-Throughput Multicast Routing Metrics in Wireless Mesh Networks

Sabyasachi Roy
Dimitrios Koutsonikolas
Saumitra Das
Y. Charlie Hu

TR-ECE-05-17
September 1, 2005

School of Electrical and Computer Engineering
1285 Electrical Engineering Building
Purdue University
West Lafayette, IN 47907-1285

Contents

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 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 study shows that (1) under a tree topology, on average, ODMRP enhanced with link-quality routing metrics can achieve 28% higher throughput than the original ODMRP under low load, i.e., low multicast sending rate; (2) the improvement reduces to 18% under high multicast sending rate due to higher interference experienced by the data packets from the probe packets; and (3) 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.

Keywords: Wireless mesh networks, multicast, high-throughput routing metrics, testbed

1 Introduction

Recently, wireless mesh networks have attracted much attention [41, 42, 43, 44, 45]. These networks typically have low maintenance overhead, high data rates, 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 is 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 [7] 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] and [20], respectively, to take into account the delay characteristics of the links. Finally, the Weighted Cumulative Expected Transmission Time (WCETT) metric was proposed in [12] to take into account the link bandwidth and loss rates of links in the presence of multiple non-overlapping channels. In [3, 10], 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 [31], DSR [19], and AODV [32].

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 [9]. Previously, a large number of multicast protocols have been designed to efficiently maintain a distributed multicast routing structure in dynamically changing topologies in MANETs [34, 16, 22, 39, 14, 35, 17, 15, 27]. 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 stationary 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 how well 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 [3, 10]. Our study is performed using ODMRP [22], a state-of-the-art multicast protocol. For each routing metric, we modify ODMRP to construct the routing structure based on that routing metric. We also compare the modified 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 [11]. In particular, on average, ODMRP using SPP or PP achieves 28% and 18% higher throughput than the original ODMRP under low and high load, respectively. Our testbed experiments show that on average, ODMRP using SPP and PP achieve 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 10% 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 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 4 gives a brief overview of ODMRP and 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 - 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.

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 msec. 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.

PP The Packet Pair (PP) metric was first proposed by Keshav in [20] 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. [11]. 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.

ETX The previous two schemes use link delay in their metrics. In [7], 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

$$ETX = \frac{1}{d_f \times d_r} \tag{1}$$

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 seconds. Since the 802.11 MAC layer protocol does not retransmit broadcast packets, nodes use this information to estimate the forward and reverse delivery probabilities.

WCETT In [12], 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

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

where

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

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

$$ETT = ETX \times \frac{S}{B} \tag{3}$$

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.

Energy Aware Routing Metrics In [3, 10], 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 when 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

$$HHR = \sum_{i=1}^{n} \frac{E_i}{1 - p_{err\,i}} \tag{4}$$

where i denotes the i^{th} link, E_i is the energy required to transmit over that link, and $p_{err\,i}$ is the error rate of that link. Equation (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 [3]

$$EER_{approx} = \frac{\sum_{i=1}^{n} E_i}{\Pi_{i=1}^{n}(1 - p_{err\,i})} \tag{5}$$

with the notations having the same meanings as above. In [10], an optimal metric for EER which improves upon Equation (5) was proposed as

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

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.

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 linker 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, [22, 5, 34, 16, 38]) 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.

The abovementioned difference in unicast and broadcast data transmissions has 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 Figure 1. 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

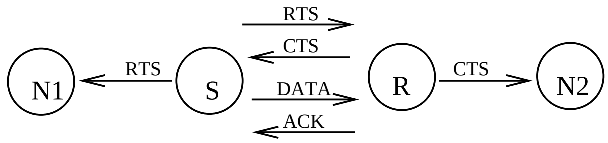


Figure 1: Packet exchanges done at the link layer during unicast data transfer. Successful data transfer involves a successful data transmission and a successful acknowledgment reception. Bidirectional link quality is critical.

Figure 2: 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.

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 Figure 2. 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. Moreover, 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 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 ten seconds.

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 Equation (1). Hence, ETX is now defined as

$$ETX = \frac{1}{d_f} \tag{7}$$

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 five seconds.

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 (Equation (3)), each node periodically sends two back-to-back probe packets every ten seconds, 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 [12], 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.

METX We modify the metric given by Equation (6) to a new metric Multicast ETX (METX). Since mesh networks are not energy constrained, we set $W(u, d)$ in Equation (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

$$METX = \sum_{i=1}^{n} \frac{1}{\Pi_{j=i}^{n}(1 - p_{err\,j})} \tag{8}$$

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

SPP We modified Equation (5) to propose the Success Probability Product (SPP) metric. The value of SPP for a path consisting of n links is given by

$$SPP = \prod_{i=1}^{n} d_{f\,i} \tag{9}$$

where $d_{f\,i}$ is the d_f value for link i ($d_{f\,i} = 1 - p_{err\,i}$). Note that if we set E_i in Equation (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 at 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

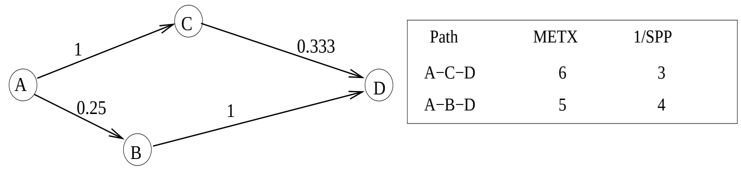


Figure 3: SPP can choose higher-throughput paths as opposed to METX at the expense of greater number of transmissions to reach a destination. The numbers over the links denote the forwarding probability ($d_{f_i} = 1 - p_{err i}$) of each link.

Characteristic	Loss	Loss + Delay	Loss + Bandwidth
Metric	METX, ETX, SPP	PP	ETT

Table 1: Routing metrics for multicast protocols and the link characteristics they measure.

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. Figure 3 gives an example showing why SPP is superior to a metric such as the one given by Equation (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.

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 [11] RTT has been shown to be problematic as the RTT measurements get distorted due to queuing delays.

4 Methodology

To evaluate the throughput improvement under the various link-quality metrics for multicast, we chose ODMRP [22] as a representative multicast protocol for wireless multihop networks. In this section, we first give a brief overview of ODMRP, and then describe the distributed implementation of the link-quality metrics in ODMRP.

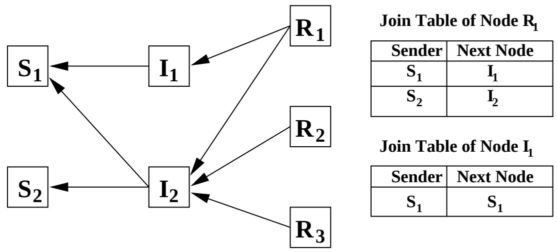


Figure 4: A simple example describing Join Table Forwarding [22]. Join Tables containing the previous hop from a source are sent in Join Replies.

4.1 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 Figure 4. 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.

4.2 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. Instead of sending back a JOIN REPLY immediately after getting the first JOIN QUERY, a group member waits for a period of δ 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 δ seconds expires, the member constructs the JOIN TABLE using

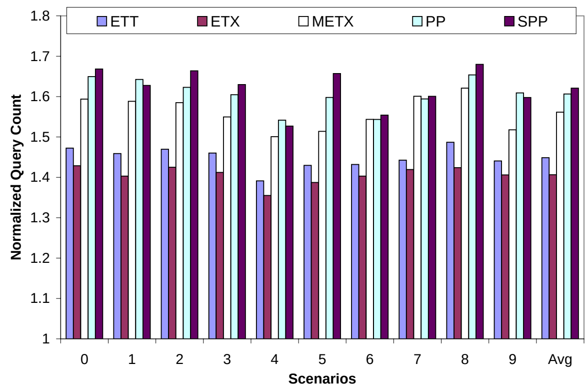


Figure 5: Query overhead comparison of original ODMRP versus ODMRP enhanced with other metrics that allow forwarding of duplicate queries for path diversity. The results show the number of queries for each case normalized with respect to the original ODMRP.

the stored JOIN QUERY, i.e., the best among all JOIN QUERY packets received during the δ period, and broadcasts the JOIN REPLY to its neighbors. Note that the δ period effectively controls the diversity of the paths that a member gets to choose from. This implementation is similar to the version of ODMRP that uses mobility prediction [24].

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 [29]. 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 till then. 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 α is kept smaller than δ because a member accepts queries only for a period of δ seconds and drops all duplicate queries received after the expiration of that period. It is important to choose α carefully as a very small value will lead to minimal path diversity, and a too high value may lead to a high query processing overhead. Such a timer 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. Figure 5 compares the overhead of query messages for the original ODMRP (without duplicate forwarding) and ODMRP using other metrics (with duplicate forwarding) in our simulation studies presented in Section 5 and shows that on average, $C < 1.7$, i.e., in all the cases the increase in the number of query packets transmitted per node is under 70% which implies that the increase in query overhead is not exponential and hence the strategies employed to restrict flooding are effective.

In the rest of the paper, we will 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.

5 Simulation results

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

5.1 Simulation setup

Scenario We used the Glomosim [40] simulator in our simulation study. We simulated a network of 50 static nodes placed randomly in a $1000m \times 1000m$ 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/second in the low-load case and 20 packets/second in a high-load case[1]. The radio propagation range was 250m and the channel capacity was 2 Mbps (the data rate used for broadcast in 802.11 MAC protocol). The simulation duration was 400 seconds. The *TwoRay* propagation model was used. In our simulations we used δ equal to 30 msec and α equal to 20 msec. Such a choice of values for α and δ enables keeping the query overhead from growing out of bounds as shown in Figure 5. In additional simulations, we found using much higher values of α and δ can yield an additional 3-4% throughput improvement. However, the optimal values of α and δ are functions of the network size, and automatically determining such values is part of our future work. 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. decreases) the 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.

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.

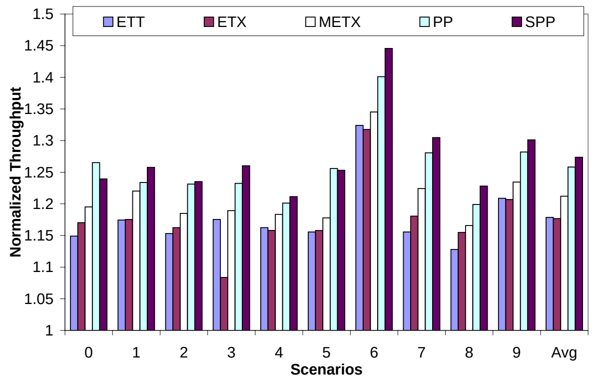
2. *Probing overhead:* This metric measures the ratio of the total number of bytes of probe packets delivered to the MAC layer over the total number of data bytes delivered to the MAC layer. For ODMRP, the probing overhead is zero, since ODMRP does not use any probe packets.

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. Unless otherwise stated, we show the results of ODMRP using various link-quality metrics normalized with respect to those of the original ODMRP.

[1] 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 traffi c load within a transmission range is on average 600 Kbps.



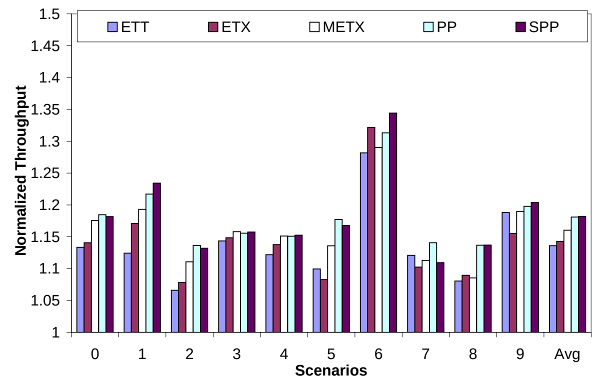


Figure 6: Throughput comparison of different routing metrics using ODMRP under low load. On average the normalized throughput for all the metrics is higher than the case with high load.

Figure 7: Throughput comparison of different routing metrics using ODMRP under high load. On average ODMRP_SPP and ODMRP_PP achieve 18% throughput gains over ODMRP.

5.2.1 Throughput

Figure 6 shows the relative throughput results for the different ODMRP versions under the low traffic 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_ETX, ODMRP_METX and ODMRP_ETT achieve about 28%, 26%, 21%, 18%, and 18% higher throughputs than ODMRP, respectively.

Figure 7 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 18%, 18%, 16%, 14.5%, and 13.5% higher throughputs than ODMRP, respectively. However, the percentage gains are lower than under the low traffic load 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.

The path from a source to a receiver, chosen by ODMRP, depends on the path 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.

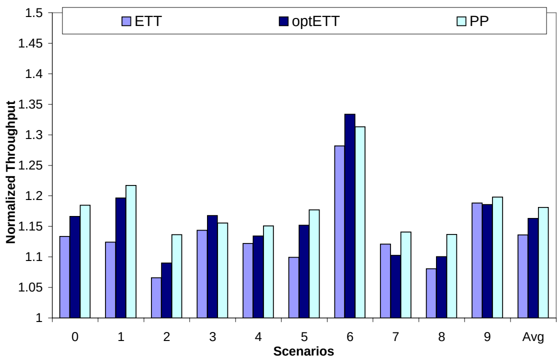


Figure 8: Throughput comparison of ODMRP_ETT, optimized ODMRP_ETT, and ODMRP_PP. Using a moving weighted average and a larger probe packet size for loss rate estimation improves the performance of ODMRP_ETT, but it still remains lower than the performance of ODMRP_PP.

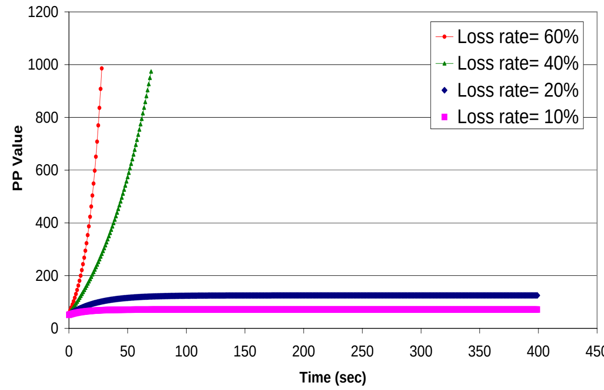


Figure 9: A plot of Equation (10) with different loss rates. PP penalty causes the link-quality metric value of a lossy link to grow exponentially with time.

Figure 7 also shows that ODMRP_ETX performs 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 ten seconds as in ODMRP_ETT but ignores the larger packet out of the two. We found ODMRP_ETT performs similarly as ODMRP_ETX. This confirms that ODMRP_ETT's high overhead can offset its performance advantage achieved from better mesh construction.

Figure 7 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 perform 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). However, a few subtle differences between the two in the way they take loss into account explain their performance difference. First, ETT uses the small packets to calculate loss in the same way as ETX does, while Packet Pair puts a penalty when either the small packet or the large packet is lost. Since the data packet size is closer to the large probe packet size, the information of large packet loss gives a better estimate of the link quality when transmitting data packets. Second, ETT keeps a history of packet losses in the past 10 probing intervals (100 seconds), while PP keeps an EWMA of the delay values. Since EWMA reflects the history of a link better, PP gets a better estimate of the link quality. To confirm this, we ran an optimized version of ODMRP_ETT which incorporates two changes to ODMRP_ETT: first, the loss rate is calculated based on the losses of the small as well as the large probe packets, and second, the EWMA of the loss values is used. However, Figure 8 shows that these optimizations only marginally improved the performance of ODMRP_ETT.

Figure 8 shows that ODMRP_PP still outperforms the optimized ODMRP_ETT. This suggests the only remaining difference between PP and ETT – the way in which a packet loss is penalized – must have contributed to their perfor-

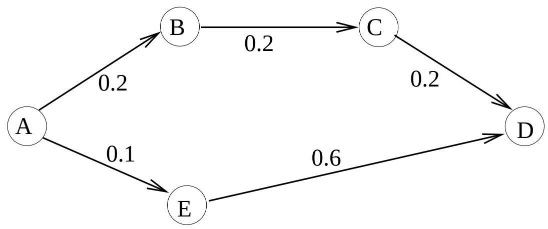


Figure 10: An example scenario in which PP chooses longer and high-throughput paths. A high-loss rate of link $E \rightarrow D$ causes the metric for the path $A \rightarrow E \rightarrow D$ to grow exponentially.

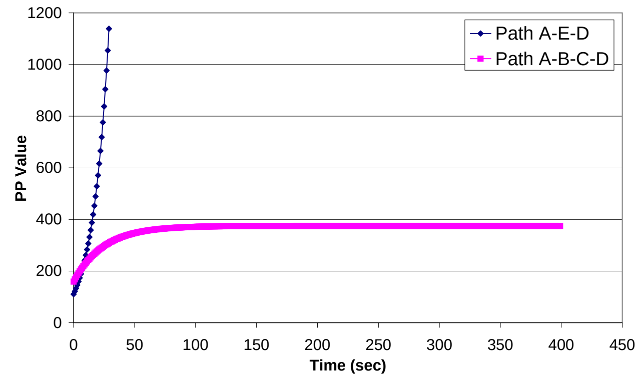


Figure 11: The aggressive penalizing by PP causes the metric value of the entire path to blow up even if it contains one lossy link.

mance difference. 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 at high loss rates, 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 nth 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

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

Figure 9 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. Figure 10 shows a hypothetical scenario in which the numbers show the loss rates of the links. Figure 11 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.

Figure 7 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 2%, 3.5% and 4.5%, respectively. With SPP being a product of probabilities, ODMRP_SPP is more effective in avoiding 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. Figure 12 illustrates how ODMRP_SPP is capable of choosing better throughput paths than ODMRP_ETX using an example network. The

Path	ETX	SPP
A−B−C−D	3.75	0.512
A−E−D	3.61	0.36

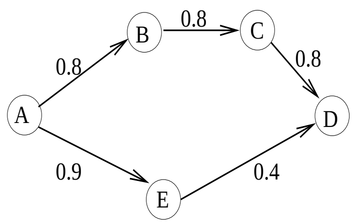


Figure 12: SPP can choose longer and higher-throughput paths as opposed to ETX. A product of link-quality metrics makes the metric value of the entire path degrade even if one link is lossy. In ETX a low-loss link can compensate for a high-loss link.

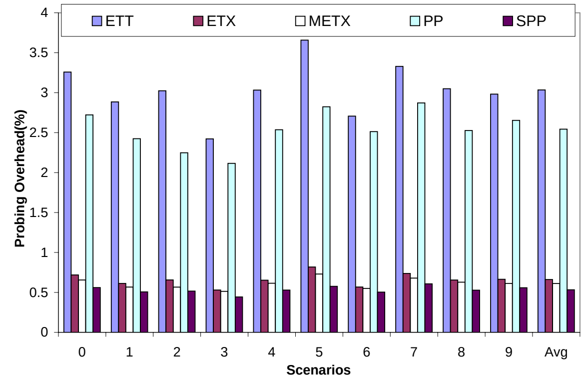


Figure 13: Comparison of the probing overhead of different routing metrics. ODMRP_PP, and ODMRP_ETT have much higher probing overhead than ODMRP_ETX, ODMRP_ETX, and ODMRP_SPP.

network consists of two paths from A to B, 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 (2 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_ETT 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-second duration of the simulation. ODMRP_ETX and ODMRP_SPP send one 160-byte probe packet every five seconds, 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 ten seconds, 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 $50 \times 80 \times 160 = 640000$ bytes and for ODMRP_PP and ODMRP_ETT is $50 \times 40 \times (160 + 1160) = 2640000$ bytes. Figure 13 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 3% 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 former will have higher 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 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 Figure 14 we show the throughput gains for all versions using link-quality metrics when the probing rate is increased by 5 times. Compared to Figure 7, we see that the throughputs of all the metrics drop by about 2%. 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 normalized average end-to-end delay for ODMRP under each of the metrics with respect to ODMRP. The results, shown in Figure 15, 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.

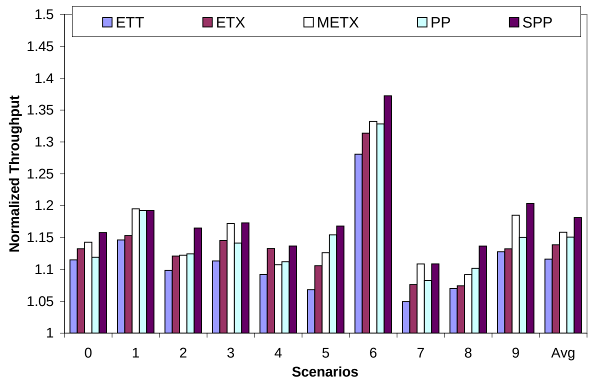


Figure 14: Throughput comparison of different routing metrics using ODMRP under high load with high probing rate. The high overhead metrics, namely ETT and PP, suffer due to increase in probing rate. The rest of the low probing overhead metrics perform similarly.

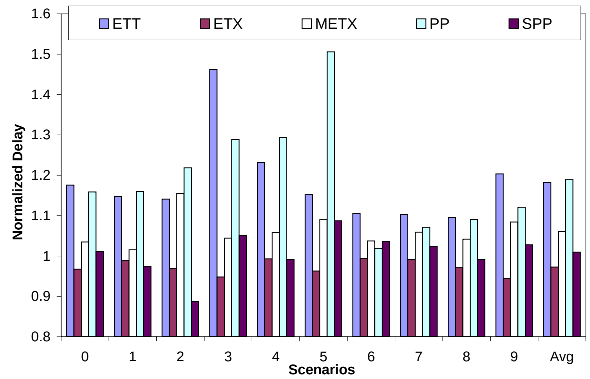


Figure 15: Delay comparison of different routing metrics. ODMRP_PP and ODMRP_ETT have the highest delay.

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

Figure 16 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%, 10%, 7% and 7%, respectively. However, compared to the single source per group case depicted in Figure 6, the relative throughput gain is reduced by around 10-15% 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. Figure 17 shows that the number of FGs increases monotonically with the number of sources per group, using the original ODMRP as the routing protocol. Figure 18 shows that a higher redundancy in data delivery obtained from the increased number of FGs leads to the 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 (Figure 6) to two (Figure 16).

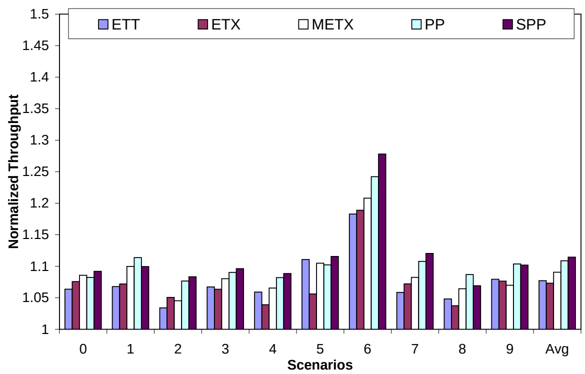


Figure 16: Normalized throughput for different high-throughput metrics under different scenarios with two sources per group under low load.

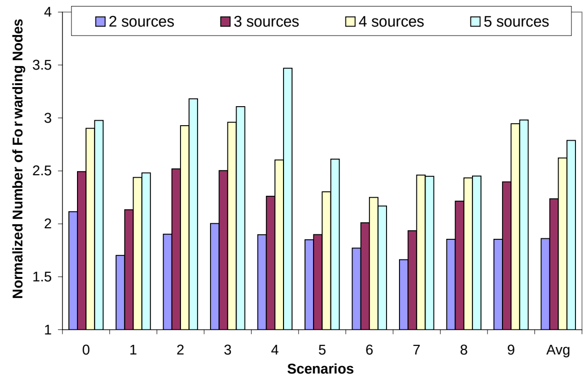


Figure 17: The average number of forwarding nodes in the network normalized under ten different scenarios with different number of sources when original ODMRP is used. All the values are normalized with respect to the case with only one source per group.

We note that previously [25, 9, 21], 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 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 Figure 7, Figure 19 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 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 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 [34]. Second, when the network is large and the number of sources is comparatively small, i.e, the number of sources per group is not 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.

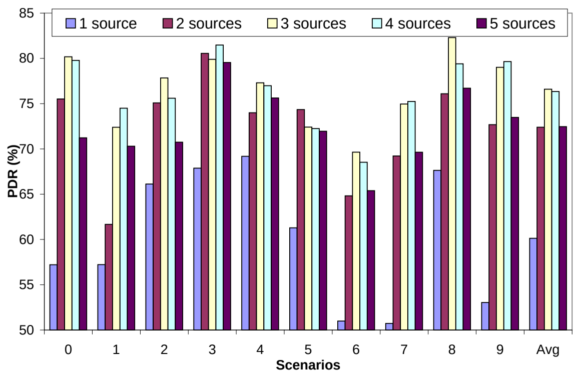


Figure 18: The Packet Delivery Ratio (PDR) of ODMRP with different number of sources per group. As the number of sources increases, the PDR increases due to increased path redundancy. At a very large number of nodes, PDR drops because of a too high load that causes packet drops due to collision and congestion.

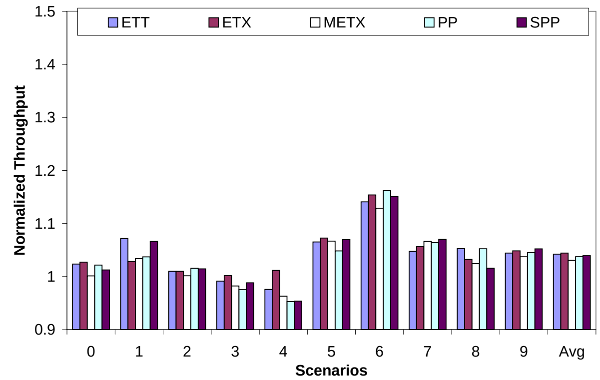


Figure 19: Normalized throughput for different high-throughput metrics under different scenarios with two sources per group under high load. ODMRP_SPP, ODMRP_PP, ODMRP_METX, ODMRP_ETX, ODMRP_ETT outperforms ODMRP only by a small margin of around 5%.

6.1 Setup

Our testbed consists of 8 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 2dBi 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 - 18dbm). 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 Figure 20. 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 the different link-quality metrics. We were unable to obtain the only known implementation of ODMRP [2]. 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 [2], 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 [26, 6, 30] as user-level daemons with loadable kernel modules for packet capturing and routing. Our implementation is based on the ODMRP specification in [23].

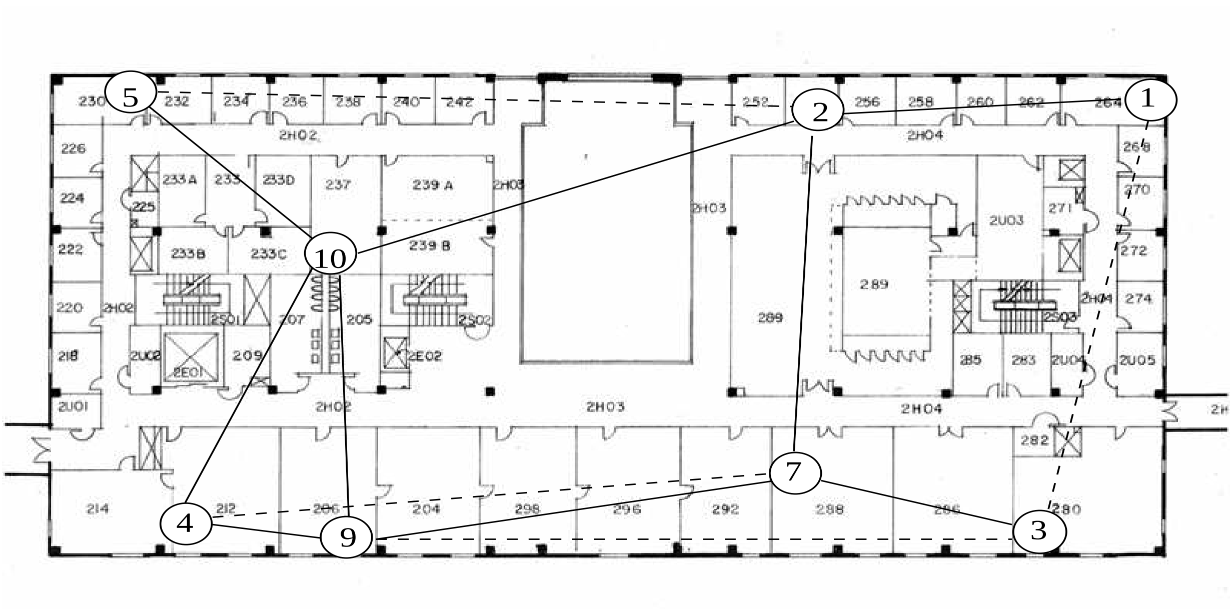


Figure 20: The floor map of our eight-node mesh network testbed deployed in an office building. The lines show the links with connectivities in the network. Solid lines denote low-loss links and dashed ones denote lossy links.

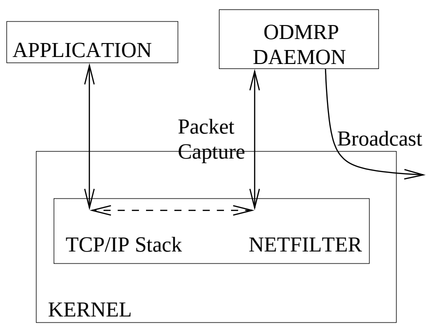


Figure 21: The architecture of our *odmrpd* implementation. The multicast packets from and to the application are intercepted by *odmrpd* using the NetFilter mechanism.

Figure 21 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

Figure 20 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

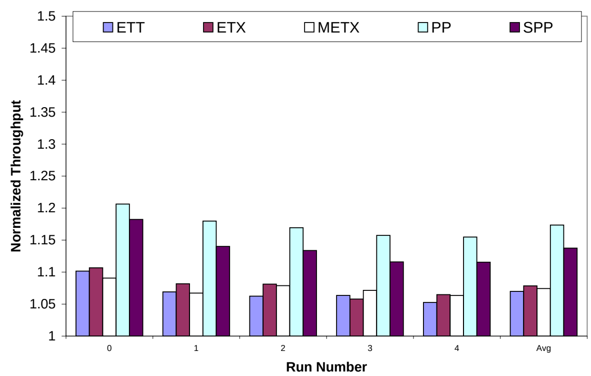


Figure 22: Throughput performance of different routing metrics on a real-mesh testbed. ODMRP using any link-quality metric outperforms ODMRP by significant margins.

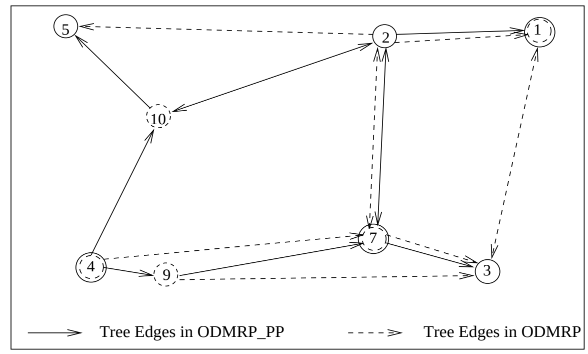


Figure 23: The trees constructed by ODMRP and ODMRP_SPP. The dashed circles denote nodes that do not belong to any group. The solid and concentric circles denote nodes of two different multicast groups.

metallic objects. In order to get an estimate of the link quality, we transfered 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 2 multicast groups, each having 1 source and 2 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/second, each of size 512 bytes. The experiments were run for 400 seconds. The same experiment was run five times to make the results resilient to random changes in the environment.

Figure 22 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. Figures 9 and 11 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% to 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. Figure 23 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 Figure 20). 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, [4, 8, 18]). Most of these protocols (for example, [31, 32, 19]) 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 [7], 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 [11], 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 [12], 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, in [3, 10], 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 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 [34], ADMR [16], ODMRP [22], CAMP [28], overlay-based protocols such as AMRoute [39], PAST-DM [14], and back-bone-based and hybrid protocols such as MCEDAR [35] and PUMA [36], and more recent stateless protocols such as DDM [17], HDDM [15], and RDG [27].

There is a significant amount of work comparing the performance of various multicast protocols for ad hoc networks (for example, [25, 9, 21, 37]). 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 more about 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 [33, 13] 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.

References

[1] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. A multi-radio unification protocol for IEEE 802.11 wireless networks. In *BroadNets*, July 2004.

[2] S. H. Bae, S.-J. Lee, W. Su, and M. Gerla. The design, implementation, and performance evaluation of on-demand multicast routing protocol in multihop wireless networks. *IEEE Network*, 14:70–77, 2000.

[3] S. Banerjee and A. Misra. Minimum energy paths for reliable communication in multi-hop wireless networks. In *Proc. of ACM MobiHoc*, 2002.

[4] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proc. of ACM MobiCom*, October 1998.

[5] C.-C. Chiang, M. Gerla, and L. Zhang. Forwarding group multicast protocol (FGMP) for multihop, mobile wireless networks. *Cluster Computing*, 1(2):187–196, 1998.

[6] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L.Viennot. Optimized link state routing protocol (OLSR). RFC 3626, October 2003.

[7] D. S. J. D. Couto, D. Aguayo, J. C. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Proc. of ACM MobiCom*, 2003.

[8] S. R. Das, C. E. Perkins, and E. M. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proc. of IEEE INFOCOM*, March 2000.

[9] C. de Morais Cordeiro, H. Gossain, and D. P. Agrawal. Multicast over wireless mobile ad hoc networks: Present and future directions. *IEEE Network*, 17(1), January/February 2003.

[10] Q. Dong, S. Banerjee, M. Adler, and A. Misra. Minimum energy reliable paths using unreliable wireless links. In *Proc. of ACM MobiHoc*, 2005.

[11] R. Draves, J. Padhye, and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. In *Proc. of ACM SIGCOMM*, 2004.

[12] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proc. of ACM MobiCom*, 2004.

[13] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proc. of ACM MobiCom*, September 2004.

[14] C. Gui and P. Mohapatra. Efficient overlay multicast for mobile ad hoc networks. In *Proc. of IEEE WCNC*, March 2003.

[15] C. Gui and P. Mohapatra. Scalable multicasting for mobile ad hoc networks. In *Proc. of IEEE INFOCOM*, March 2004.

[16] J. G. Jetcheva and D. B. Johnson. Adaptive Demand-Driven Multicast Routing in Multi-Hop Wireless Ad Hoc Networks. In *Proc. of ACM MobiHoc*, October 2001.

[17] L. Ji and S. Corson. Differential Destination Multicast–A MANET Multicast Routing Protocol for Small Groups . In *Proc. of IEEE INFOCOM*, April 2001.

[18] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *Proc. of ACM MobiCom*, August 1999.

[19] D. B. Johnson and D. A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*. Kluwer Academic, 1996.

[20] S. Keshav. A control-theoretic approach to flow control. *Proceedings of the conference on Communications architecture and protocols*, pages 3–15, 1993.

[21] T. Kunz and E. Cheng. On-demand multicasting in ad-hoc networks:comparing AODV and ODMRP. In *Proc. of IEEE ICDCS*, July 2002.

[22] S.-J. Lee, M. Gerla, and C.-C. Chiang. On-Demand Multicast Routing Protocol. In *Proc. of IEEE WCNC*, September 1999.

[23] S.-J. Lee, W. Su, and 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.

[24] S. J. Lee, W. Su, and M. Gerla. On-demand multicast routing protocol in multihop wireless mobile networks. *Mob. Netw. Appl.*, 7(6), 2002.

[25] S.-J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia. A performance comparison study of ad hoc wireless multicast protocols. In *Proc. of IEEE INFOCOM*, March 2000.

[26] H. Lundgren and E. Nordstrm. Aodv implementation code (Uppsala University), 2004. `http://core.it.uu.se/AdHoc/AodvUUImpl`.

[27] J. Luo, P. T. Eugster, and J.-P. Hubaux. Route Driven Gossip: Probabilistic Reliable Multicast in Ad Hoc Networks. In *Proc. of IEEE INFOCOM*, March 2003.

[28] E. L. Madruga and J. J. Garcia-Luna-Aceves. Scalable multicasting: The core-assisted mesh protocol. *ACM/Baltzer Mobile Networks and Applications, Special Issue on Management of Mobility*, 6(2):151–165, Apr. 2001.

[29] T. Nadeem, A. Agrawala, and S. Banerjee. Energy-efficient reliable paths for on-demand routing protocols. In *Proc. of IEEE MWCN*, 2004.

[30] E. Nordstrm. Dsr implementation code (Uppsala University), 2005. `http://core.it.uu.se/AdHoc/DsrUUImpl`.

[31] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proc. of ACM SIGCOMM*, August 1994.

[32] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. In *Proc. of IEEE WMCSA*, February 1999.

[33] A. Raniwala and T. Chiueh. Architectures and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *Proc. of IEEE INFOCOM*, March 2005.

[34] E. M. Royer and C. E. Perkins. Multicast operation of the ad-hoc on-demand distance vector routing protocol. In *Proc. of MobiCom*, August 1999.

[35] P. Sinha, R. Sivakumar, and V. Bharghavan. MCEDAR: Multicast Core-Extraction Distributed Ad hoc Routing. In *Proc. of IEEE WCNC*, September 1999.

[36] R. Vaishampayan and J. J. Garcia-Luna-Aceves. Efficient and robust multicast routing in mobile ad hoc networks. In *Proc. of MASS 2004*, September 2004.

[37] U. Varshney. Multicast over wireless networks. *Commun. ACM*, 45(12):31–37, 2002.

[38] C. W. Wu and Y. C. Tay. Ad hoc multicast routing protocol utilizing increasing id-numbers (amris) networks. In *IEEE MILCOM*, Nov. 1999.

[39] J. Xie, R. R. Talpade, A. Mcauley, and M. Liu. AMRoute: ad hoc multicast routing protocol. *Mob. Netw. Appl.*, 7(6):429–439, 2002.

[40] X. Zeng, R. Bagrodia, and M. Gerla. Glomosim: A library for parallel simulation of large-scale wireless networks. In *Proc. of PADS Workshop*, May 1998.

[41] MIT Roofnet. `http://www.pdos.lcs.mit.edu/roofnet`.

[42] Seattle wireless. `http://www.seattlewireless.net`.

[43] Bay area wireless users group. `http://www.bawug.org`.

[44] Mesh Networks Inc. mesh networks technology overview. `http://www.meshnetworks.com`.

[45] Radiant networks. `http://www.radiantnetworks.com`.