High-Throughput Multicast Routing Metrics in Wireless Mesh Networks

Sabyasachi Roy, Dimitrios Koutsonikolas, Saumitra Das, and Y. Charlie Hu
School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN 47907
{roy0, dkoutson, smdas, ychu}@purdue.edu

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. In recent years, numerous link-quality-based routing metrics have been proposed for choosing high-throughput paths for unicast protocols. 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 there is a difference in how the routing metrics for each of these primitives are designed. We adapt certain routing metrics for unicast for high-throughput multicast routing and propose news ones not previously used for high-throughput. 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 testbed experiment results show that ODMRP enhanced with link-quality routing metrics can achieve up to 17.5% throughput improvement as compared to the original ODMRP.

1. Introduction

Recently, wireless mesh networks have attracted much attention. 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 have been proposed in the past [7, 1, 16, 10, 3, 8]. All these metrics have been proposed and evaluated for unicast routing protocols such as DSDV [24], DSR [15], and AODV [25].

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. Unlike unicast, all the routing algorithms proposed for multicast [27, 13, 17, 30, 11, 28, 14, 12, 20] use minimum-hop-count as the routing metric and focus on scenarios with high mobility.

In this paper, we study the design of link-quality-based routing metrics for high-throughput multicast in mesh networks. Our approach is based on the observation that there is a fundamental difference in the way the MAC layer handles multicast packets as opposed to unicast packets. Typically multicast packets are broadcast at the MAC layer as opposed to unicast in order to leverage the wireless multicast advantage (WMA) [4]. Thus directly using the link-quality-based metrics proposed for unicast is not appropriate.

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 comparative 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, 8]. Our study is performed using ODMRP [17], a state-of-the-art multicast protocol.

Our simulation study with a 50-node mesh network shows that ODMRP using any of the five metrics, ETT, ETX, METX, PP, and SPP, outperforms the original ODMRP by significant margins of improvement similar to those achieved in unicast routing using high-throughput routing metrics [9]. In particular, on average, ODMRP using SPP or PP achieves 18% higher throughput than the original ODMRP. Our experiments on an eight-node testbed show that on average, ODMRP using SPP and PP achieve 14% and 17% higher throughput over ODMRP, respectively. 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 discusses the fundamental difference between multicast and unicast modes of communication in multihop wireless networks and describes how to accordingly modify the existing unicast routing metrics for multicast routing. Section 3 describes the changes made to ODMRP in order to incorporate the routing metrics. Section 4 presents simulation results and Section 5 presents experimental results on a mesh network testbed. Finally, Section 6 concludes the paper.

2. 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.

2.1. 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, [17, 5, 27, 13, 29]) use link-layer broadcast to leverage WMA. WMA improves the reliability of data transfer and hence increases efficiency. 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 data. 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 transfer. Further, data transmission is acknowledged by the receiver. If an acknowledgment is not received, the MAC layer reattempts the data transmission for a number of times. In contrast, the 802.11 MAC layer broadcast does not involve any RTS/CTS exchange. 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 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 to a receiver followed by a successful transfer of an acknowledgment back, in addition to an exchange of RTS/CTS between the two nodes. Hence, the overall quality of a link depends on the link characteristics in both the forward and reverse direction. In case of broadcast, there are no acknowledgments and thus a successful data transfer only depends on the link quality in the forward direction. 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 one chance to properly travel from one node to another. This implies that unlike unicast, for loss-rate-based link-quality metrics such as ETX, simply adding 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 better reflects the quality of the path.

2.2. Adapting unicast link-quality metrics for multicast

The above differences between unicast and multicast suggest that the link-quality metrics designed for unicast 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. All metrics involve sending periodic probes from a node to each of its neighbors. The metric of each link is calculated by the receiver which adds it to the path metric as query packets flow through, i.e., during multicast tree construction.

PP We adapt unicast PP [16, 9] for multicast by broadcasting probe packets instead of unicasting them to each neighbor node. A pair of probe packets are sent every 10 seconds. The delay for a link is calculated as an Exponentially Weighted Moving Average (EWMA). 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. The value of the metric for a path is the sum of the PP values of the individual links.

ETX We adapt unicast ETX [7] for multicast by not considering reverse path link quality. A probe packet is sent every five seconds, and ETX is now defined as $ETX = \frac{1}{d_f}$, where d_f is the loss rate of the link in the forward direction. The value of the metric for a path is the sum of the ETX values of the individual links.

ETT Since we do not assume multiple channels when comparing different metrics in this paper, we adapt ETT [10] 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. Probing is performed in the same way as in PP. ETT of a link is composed of information about the loss rate and the bandwidth of a link. In the adapted ETT, the receiver uses the small packets received to calculate ETX. The bandwidth of each link is estimated by dividing the size of the big packet by the inter-arrival time between the small and the large packets.

METX In [8], the authors propose routing metrics to minimize the total transmission energy. Under the assumption of an unreliable link layer, they propose an energy-efficient routing metric for a path

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

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, $p_{err\,l}$ is the error rate of that link, and $W(u,d)$ is the transmission energy required between nodes u and d. We modify the metric given by Equation (1) to a new metric, Multicast ETX (METX). Since mesh networks are not energy constrained, we set $W(u,d)$ in Equation (1) to 1. Such a transformation gives us the total expected number of transmissions needed by all the nodes along a path 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

$$METX = \sum_{i=1}^{n} \frac{1}{\Pi_{j=i}^{n}(1-p_{err\,j})} \qquad (2)$$

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

SPP In [3], the authors propose an energy-efficient routing metric for a path

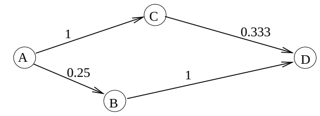
$$EER_{approx} = \frac{\sum_{i=1}^{n} E_i}{\Pi_{i=1}^{n}(1-p_{err\,i})} \qquad (3)$$

where i denotes the i^{th} link, E_i is the energy required to transmit over that link. We modified Equation (3) to propose the Success Probability Product (SPP) metric. The value of SPP for a path consisting of n links is given by $SPP = \prod_{i=1}^{n} d_{f\,i}$, where $d_{f\,i} = 1 - p_{err\,i}$. Note that if we set E_i in Equation (3) to 1, the resulting value is n times the value of 1/SPP. Note that SPP gives the the probability for the destination node to receive a packet properly over a path with link-layer broadcast, and hence 1/SPP reflects the expected number of transmissions at the source itself. The routing algorithm selects the path with the maximum SPP (minimum 1/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.

Figure 1 gives an example showing why SPP is superior to a metric such as METX that tries to minimize the total number of transmissions.

3. Methodology

To evaluate the throughput improvement under the various link-quality metrics for multicast, we chose ODMRP [17], a state-of-the-art protocol, as a representative multicast protocol for wireless multihop networks. Note that the various link-quality metrics can easily be incorporated into any other routing protocol and we believe that the findings would not change drastically if the underlying multicast protocol is changed. In this section we describe the distributed implementation of the link-quality metrics over ODMRP.



Path	METX	1/SPP
A–C–D	6	3
A–B–D	5	4

Figure 1. SPP can choose higher-throughput paths than METX by minimizing the expected number of packet transmissions at the source. The numbers over the links denote the forwarding probability ($d_{f\,i} = 1 - p_{err\,i}$) of each link.

3.1. Incorporating link-quality metrics

To incorporate the new link-quality metrics into ODMRP, we modified ODMRP as follows. Each node maintains a 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 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 [18].

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 [22]. To limit the overhead of queries, 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 of $\alpha < \delta$ seconds when it receives the first JOIN QUERY with a particular sequence number. Each node forwards duplicate queries only until the timer of α seconds expires. It is important to choose α carefully as a very small value will lead to minimal path diversity, and a very high value may lead to a high query processing overhead.

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.

4. Simulation Results

4.1. Simulation setup

We used the Glomosim [31] 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 ten members each. The sources sent CBR traffic, consisting of 512-byte packets sent at a rate of 20 packets/second[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. 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. We simulated each protocol on 10 different randomly generated topologies and the results for the average over all topologies are presented.

4.2. Results for Single Source per Group

In this section, we present the performance results of the various versions of ODMRP with single source per group. Unless otherwise stated, we show the results of ODMRP using various link-quality metrics normalized with respect to that of the original ODMRP.

4.2.1. Throughput

Figure 2, column "Throughput-simulations" shows the relative throughput results for the different ODMRP versions. 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. Note that we also did simulations under lower load and found similar qualitative results, but are not shown due to space limitation.

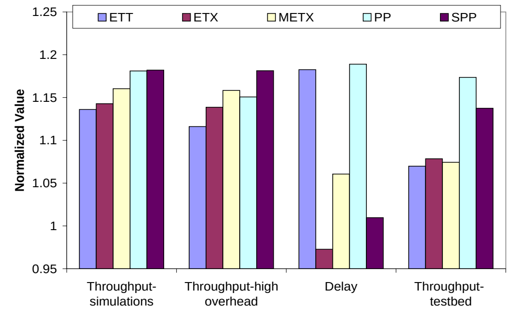


Figure 2. The relative performance of the different routing metrics in terms of throughput and delay normalized with respect to ODMRP.

ODMRP performs poorly because of fading. Fading is defined as a random change in the attenuation of a communication 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 2 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 (Section 4.2.2), which was confirmed by running ODMRP_ETX with ODMRP_ETT's overhead and getting similar results for both of them.

Figure 2 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, since 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 for such a

[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 traffic load within a transmission range is on average 600 Kbps.

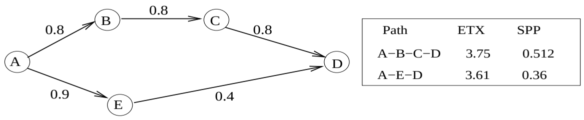


Figure 3. SPP can choose longer and higher-throughput paths than ETX by avoiding a path containing even a single lossy link. The numbers over the links denote the forwarding probability of each link.

difference between PP and ETT is the way in which a packet loss is penalized. 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.

Figure 2 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 the 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 3 illustrates how ODMRP_SPP is capable of choosing better throughput paths than ODMRP_ETX using an example network.

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, hence it performs poorly in terms of throughput.

Table 1. Comparative percentage overhead for the different routing metrics.

Metric	ETT	ETX	METX	PP	SPP
% Overhead	3.03	0.66	0.61	2.54	0.53

4.2.2. Probing overhead

In this section, we compare the probing overhead of various protocol versions that use link-quality metrics. Table 1 shows the percentage of bytes from probe packets out of the total number of data bytes received. 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 4.2.3.

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 2 the column “Throughput-high overhead” shows the throughput gains for all versions using link-quality metrics when the probing rate is increased by 5 times. Compared to Figure 2, column “Throughput-simulations”, 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. 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.

4.2.3. Delay

We also measured the normalized average end-to-end delay for ODMRP under each of the metrics with respect to ODMRP. The results, shown in Figure 2, column “Delay”, show that in most cases, ODMRP_SPP and ODMRP_ETX achieve lower end-to-end delays than the rest of the 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.

4.3. Results for Multiple Sources per Group

Since ODMRP creates forwarding group members per group and not per source[2], it builds a more redundant mesh-structure when there are multiple sources per group than when there is a single source per group. This increased redundancy of data delivery paths compensates the original ODMRP's inability to choose high-throughput paths and reduces the throughput improvement from using high-throughput routing metrics. Our simulation results show that the relative throughput gain is reduced by around 10-15% for the different link metrics (The details are omitted due to space limitation and can be found in [26].) 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 [27]. Second, when the network is relatively large and 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 unnecessary data traffic in the network.

5. Testbed experiments

To validate 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 implemented ODMRP using all the different routing metrics and experimentally compared them to the original ODMRP on this testbed.

5.1. Setup

Our testbed [21] consists of 8 wireless mesh routers (small form factor PCs 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 4. 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.

5.2. 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 [19, 6, 23] as user-level daemons with loadable kernel modules for packet capturing and routing.

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 throughout 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.

5.3. Results

Figure 4 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 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

[2] 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 of the same group.

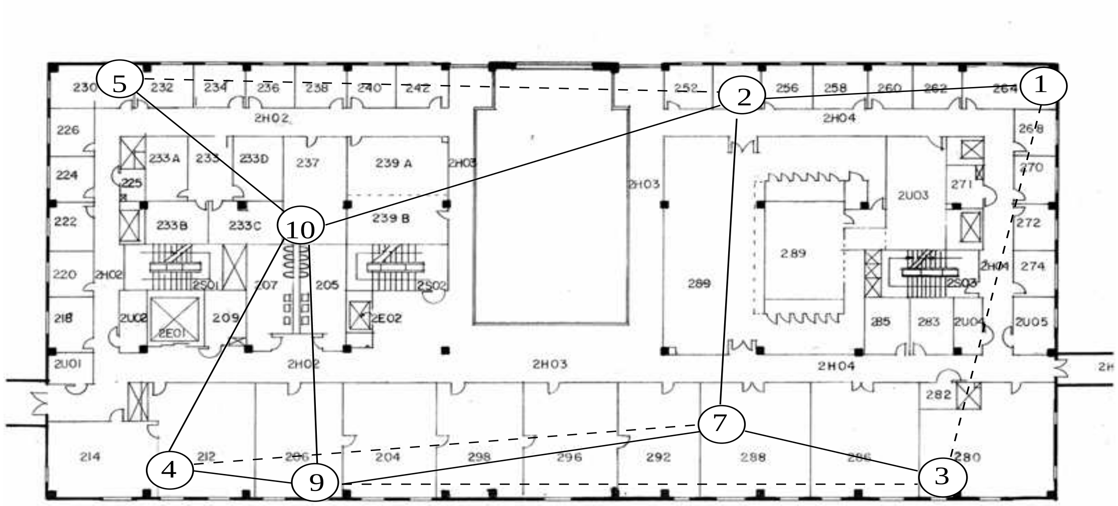


Figure 4. The floor map of our eight-node mesh network testbed deployed in an office building.

lines show the links that are lossy and the solid lines show the links that have low or almost no loss. The pairs of nodes with no lines between them cannot 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 2, column "Throughput-testbed" 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. Under high loss-rates, PP has the property of causing the cost of a path to blow up exponentially. But under 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 those 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 because PP uses a long history based on EWMA. 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, they are chosen again under these metrics because such metrics have a small history window.

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 5 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 4). Similarly, node 4 chooses a one-hop path to 7 which is lossy. In contrast, ODMRP_PP 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 between 3 and 1, or 4 and 7, or 9 and 3. ODMRP_PP is again able to figure out the lossy links and avoid them.

6. Conclusions and Future Work

In this paper, we have studied various link-quality routing metrics for high-throughput multicast in mesh networks.

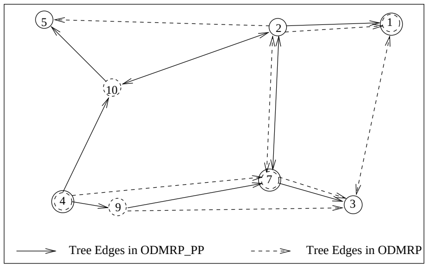


Figure 5. The trees constructed by ODMRP and ODMRP_SPP. The dashed circles denote nodes that do not belong to any group. The solid and concentric circles denote nodes of two different multicast groups.

We first discussed the fundamental difference between unicast and multicast routing in how data packets are transmitted at the link layer, and then showed how to adapt unicast routing metrics for use in multicast. We studied the performance of different metrics via extensive simulations and experiments on a mesh network testbed, using ODMRP as a representative multicast protocol. Our simulation 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. Finally, our experimental results on an eight-node mesh network testbed validate the results obtained in the simulation study.

In our future work, we plan to investigate more about the optimal probing rate, and to extend the high-throughput link-quality metrics studied in this paper for multicast routing in multi-radio/multi-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] M. Cagalj, J.-P. Hubaux, and C. Enz. Minimum-energy broadcast in all-wireless networks: NP-Completeness and distribution. In *Proc. of ACM MobiCom*, September 2002.

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

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

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

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

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

[13] 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.

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

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

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

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

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

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

[20] 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.

[21] Mesh@Purdue. `www.engineering.purdue.edu/MESH`.

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

[23] E. Nordstrom. DSR implementation code (Uppsala University), 2005. `http://core.it.uu.se/AdHoc/DsrUUImpl`.

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

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

[26] S. Roy, D. Koutsonikolas, S. Das, and Y. C. Hu. High-throughput multicast routing metrics in wireless mesh networks. Technical Report TR-ECE-05-17, Purdue University, Sept. 2005.

[27] 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.

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

[29] C. Wu and Y. Tay. AMRIS: A Multicast Protocol for Ad hoc Wireless Networks. In *Proc. of MILCOM*, November 1999.

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

[31] 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.