Opportunistic Multipath Routing in Wireless Mesh Networks

Jack W. Tsai[1,2] and Tim Moors[1]

[1] University of New South Wales, Sydney, Australia
[2] NICTA, Australia[*]
jackwtsai@gmail.com, moors@ieee.org

Abstract. This paper investigates combining opportunistic routing techniques with multipath routing for achieving reliability and timeliness in fast-changing network conditions. We present two approaches, WIMOP and DOMR, based on source routing and distributed routing, respectively. Instead of using broadcast packets as in most opportunistic routing work, we use unicast with promiscuous listening so that the reliability at each hop can be increased through retransmissions, while maintaining the broadcasting property required by opportunistic routing. We evaluate our work in NS2 against single path routing and MORE. Our results show that using the same amount of redundant data, our approaches were able to achieve better reliability than MORE. In addition, DOMR also has the advantage over WIMOP that it requires significantly less computational time.

Keywords: Wireless, mesh, multipath, routing, opportunistic.

1 Introduction

Wireless mesh networks have been used to provide cheap network connectivity in a range of scenarios. The main challenge for mesh networks, especially in outdoor applications, is to overcome unstable and unreliable links caused by the environment such as fading, moving vehicles and external interference. Compared to WLANs, these problems associated with the wireless medium are magnified due the multi-hop nature of wireless mesh networks. It has been observed in some testbeds that many links have loss ratios as high as 50% [1]. Recently, a new breed of routing protocols, based on opportunistic routing, has emerged to address such issues.

In traditional routing protocols, a next-hop is selected before the packet is forwarded towards the destination. On the other hand, opportunistic routing exploits the broadcast nature of the wireless medium such that nodes which overheard a packet transmission can also participate in the packet forwarding process, therefore

---

[*] NICTA is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

N. Bartolini et al. (Eds.): QShine/AAA-IDEA 2009, LNICST 22, pp. 71–85, 2009.

increasing the probability of a packet transmission being successful at propagating the packet towards the destination.

To date, the main focus and application of opportunistic routing has been on improving traffic throughput by reducing the medium access time. In this paper we investigate using opportunistic routing to improve performance for those applications that require reliability and timeliness instead. In opportunistic routing, depending on the receptions at each hop, packets can, and usually do take different paths to the destination. This is in some ways similar to multipath routing, though the choice of multiple paths isn't predetermined by the source. Our work is to build on existing opportunistic routing but also explicitly require the delivery to use multiple paths in the process. We have previously proposed using multipath routing as a means to achieve these goals [2]. In this work, multiple paths are calculated from the source node such that the interference between different paths (inter-path) and also between nodes within each path (intra-path) is minimised. This method requires an exhaustive search of the path space and thus is not easily scalable, and also relies on the accuracy and freshness of the link metric at the source node. Any short term link quality changes at the intermediate nodes are not taken into account.

Since our proposed work needs to adapt to fast changing network conditions, we first investigate a new link quality estimation technique that rapidly reflects the link quality by complementing ETX [3] with the number of link layer retransmissions measured from the successfully transmitted packets.

In this paper we investigate two approaches of applying opportunistic routing to multipath routing. The first builds on our previous work in source-based multipath routing. At each node, the link quality of the next-hop link specified in the source-routed packet is constantly monitored, so that when it deteriorates below a certain threshold, the node can choose to bypass the link by electing a neighbour node to capture the transmission and forward the packet using the neighbour's best available path. The main benefit of this addition is that we can overcome the potential inaccuracy in metric calculation at the source due to the lag in time in the exchange of link quality information.

Our second approach is a distributed opportunistic multipath routing protocol. At each hop, a list of candidate forwarders is calculated using link quality and interference information. Each candidate forwarder has a probability of forwarding that depends on the quality of the paths it provides to the destination. Therefore the amount of redundant data in the network is controlled by adjusting the forwarding probabilities. The proposed routing protocol has the following properties:

- Distributed
- Opportunistic and multipath
- Interference aware

The rest of the paper is organized as follows. In §2 we present the related work. In §3 the new link quality metric and the two approaches to opportunistic routing are presented. Simulation results from NS2 are presented in §4, and we conclude the paper in §5.

2 Related Work

Opportunistic routing exploits the broadcast nature of the wireless medium by allowing more than one neighbour to participate in the packet forwarding process. The concept of opportunistic routing was first proposed in ExOR [4]. The two main issues in opportunistic routing are forwarder selection and coordination rules. Since it is clearly detrimental to involve all neighbouring nodes in packet forwarding (for example, using nodes farther from the destination), forwarder selection is used to choose a set of neighbours that will provide the best forwarding performance. Coordination rules help decide which of the forwarders that have successfully received a transmission should forward the packet. This is to prevent unnecessary transmissions caused by forwarding packets that are likely to have been forwarded by other nodes already, which waste bandwidth and create interference.

In ExOR the sender computes a forwarder list by ranking the neighbours in terms of their ETX to the destination and picking those with smaller values than itself. It enforces coordination between the forwarders by using a strict packet scheduler in the MAC layer; each forwarder is given a time slot according to its priority and can only transmit during that slot. During a transmission, other forwarders listen in and record the packets that are being sent. This information is passed on where possible so that packets already transmitted by one forwarder are dropped by the rest.

The main problem with ExOR is that it requires a customised MAC in order to schedule packets, which increases hardware cost and restricts deployment. In contrast, MORE [5] proposed using randomness provided by network coding to eliminate the need for a scheduler and thus can be used with 802.11. This works by requiring nodes to transmit coded packets, which are linear combinations of multiple packets. A forwarder receives a coded packet and decides if it contains new information that it has not already received, if so it will forward a new linear combination of the received coded packets. When the destination receives enough coded packets to reconstruct the original packets, it immediately acknowledges the source to initiate the transmission of a new batch of packets. Because the data transmitted by each forwarder is a linear combination of received packets with random coefficients, the probability of nodes transmitting the same information and wasting bandwidth is greatly reduced, therefore a scheduler is no longer needed.

The forwarder selection in both ExOR and MORE considers only the ETX cost of each forwarder to the destination. While this is simple, it might not be optimal in achieving reliability. [6] addresses the least-cost opportunistic routing problem by proposing an algorithm that assigns and prioritises the set of candidate relays (forwarders) so that the cost of forwarding a packet to the destination is minimised. In our work we propose a similar forwarder selection that considers spatial diversity and also explicitly allow multiple copies of a packet to be forwarded.

The opportunistic protocols mentioned so far all focus on using broadcasts to forward packets to the destination, with some requiring link layer scheduling. However, unlike unicast, broadcast does not use retransmissions and collision detection (RTS/CTS). As a result, broadcast based routing might not perform well in terms of reliability when links are very unreliable. In contrast, we propose a unicast-based opportunistic routing that at least ensures reliability to a certain degree.

3 Protocol Description

3.1 Link Quality Estimation

ETX and similarly derived broadcast link quality metrics have recently been shown to be poor indicators of the real link quality experienced by data traffic in some cases. One problem is that ETX measures the performance at the receiver of a link without considering the exponential back-off time incurred by channel contention at the sender. The lack of RTS/CTS for broadcast traffic also causes inaccurate estimations due to the hidden terminal problem.

As a first step to improve link quality estimation, we note that the 802.11 MAC layer provides useful information about the current link condition, such as link rate, SINR, retransmission counts, etc. We are particularly interested in the number of retransmissions actually performed to forward a packet to a neighbouring node, which is closely related to the ETX metric concept. The advantage of using the retransmission counts to gauge the link quality is that it is measured on unicast traffic, which eliminates the problem with broadcasted probes as described before. In addition, it does not add to traffic overhead like probe-based metrics do. The main problem with retransmission counts, however, is that it requires active traffic over the link in order to be of any significance. To solve these problems, we propose to combine a probe-based metric with local MAC layer retransmission counts to get the best of both worlds.

The main use of the ETX base metric is when there is little or no traffic on the networks, it has been shown that under these conditions ETX base metrics give very accurate estimates of link qualities [7]. Thus the new metric should have a component of ETX whose weight is inversely proportional to the data traffic on the link:

$$ETX\text{-}R = \left(ETX \times \left(1 - \frac{t_{i,j}}{b_{i,j}}\right) + r_{ij}(T) \times \frac{t_{i,j}}{b_{i,j}}\right) \tag{1}$$

where

$r_{ij}(T)$ is the average number of link layer retransmissions required for a successful unicast transmission in the last T seconds

$t_{i,j}$ is the transmission rate (load) on link *i,j*

$b_{i,j}$ is the link rate of link *i,j*

3.2 Load-Aware Path Metric

In our prior work [2] we proposed a multipath routing algorithm based on interference minimisation. The source node computes the paths that are used for forwarding traffic to a destination using a WIM score. The WIM score reflects the degree of interference between different paths in a multipath set as well as individual path quality. The main component in the WIM score is the interference cost for a link *i,j* operating on channel *c* in a network *N*,

$$LI_{ij}(c,N) = ETT_{ij}(c) * \left|E_{ij}(c,N)\right| \tag{2}$$

where ETT [8] is the expected transmission time for a packet of size *S*, derived from ETX and the link bandwidth B,

$$ETT_{ij}(c) = ETX_{ij}(c) * S/B, \tag{3}$$

and $|E_{ij}(c, N)|$ is the number of mutual interference sets (set of links in which only one link can successfully transmit at a time) affected by transmission on link *i,j*.

The original *LI* calculation assumes that all links are active, while in many real-life application traffic are often bursty and links can be idle for periods of time. Consequently the original *LI* calculates the worst-case interference. Therefore in this paper we modify link interference estimation to include the load of a link. The load-aware link interference is:

$$LLI_{ij}(c, N) = ETT_{ij}(c) * LF_{ij} * S_{ij}(c, N), \tag{4}$$

where $S_{ij}(c, N)$ is the number of nodes that link *ij* interferes with.

The WIM score for a path set *P* is a weighted sum of the aggregated link interference of *P* and that of other nodes in the network (*N-P*), using a weight factor *β*.

$$PIC \quad = \sum_{ij \in p} LLI_{ij}(c, P) \tag{5}$$

$$NIC \quad = \sum_{ij \in p} LLI_{ij}(c, N - P) \tag{6}$$

$$WIM \quad = \beta * NIC \quad + (1 - \beta) * PIC \tag{7}$$

3.3 Unicast-Based Opportunistic Routing

Unlike the conventional approach to opportunistic routing, which uses broadcast to forward traffic towards the destination, our protocol adopts a unicast-based approach similar to that in [9]. Instead of sending packets in broadcast mode, the sender uses unicast to send packets to a primary recipient, while other nodes in the vicinity listen promiscuously and opportunistically capture packets. The main motivation behind our approach is to ensure reliability. In the 802.11 MAC broadcast mode there is no acknowledgement so the sender does not retransmit. Consequently reliability suffers. Broadcast-based opportunistic routing has been shown to improve data throughput and network capacity [4] [10], but link layer retransmission may be a better way to ensure reliability than having to implement acknowledgements above the link layer. Therefore in our proposed framework, each node listens in on wireless transmissions regardless of whether they are broadcasts or unicasts.

3.4 Approach One: Source Multipath Routing (WIMOP)

The source-based multipath routing algorithm we proposed in [2] relies on the accuracy of the metric at the time of the route computation to perform well. However in some networks even if the topology is static, link quality may vary greatly due to the environment. In these cases the time required for the link state algorithm to disseminate link quality information across the network may be longer than the coherence time of the channels, and therefore the link metrics do not reflect well the actual link qualities. To overcome this problem we propose using opportunistic routing at nodes along the paths when a link deteriorates.

Each node monitors the number of retransmissions used (r_{ij}) in the transmission of packets to each of its neighbour. A long term and a short term exponential moving average of the retransmission counts are computed for each link. These averages are used to help identify short term link deterioration. The source sends out data as before using source routing, at each intermediate node the quality of the next-hop link is examined. When the short term average r_{ij} drops below the long term average by more than a threshold value, indicating that the instantaneous link quality is below expectation, the node will find an alternative next-hop to collaborate in the forwarding of the packet. The node considers the total cost of delivering the packet via each of its neighbours without forming a loop. The neighbour with the best cost is flagged in the packet header as a *collaborator* to which data is forwarded using unicast transmissions until the short term r_{ij} of the next-hop link indicated in the source route moves back above the threshold. In addition, if the original next-hop node captures a packet sent to the collaborator, it will still need to forward the packet. In other words, the nodes in the source routes will opportunistically forward packets that are captured.

When the collaborator captures a packet, it calculates the best path to deliver the packet to the destination using the WIM calculation on the path candidate and existing multipath packets already indicated in the packet. This is done so the new path is selected to avoid interference with other paths delivering the same packet.

The detail is described as follows in Algorithm 1 and is illustrated in Fig. 1.

Algorithm 1

```
$C_{i,d}(j)$ is the path cost from node i to destination via node j
$T_l > T_s$, $T_s$ $T_l$ are the intervals for retransmission counts collection
If $r_{ij}(T_s) > r_{ij}(T_l)$
  do for all node n in neighbours
      find nl such that $C_{i,d}(nl)$ = min($C_{i,d}(n)$)
      do flag packets to s to indicate nl as collaborator
        unicast packet to collaborator
 continue until If $r_{ij}(T_s) < r_{ij}(T_l)$
```

Limiting the Number of Duplicates in the Network

When a collaborator is used, if the original next-hop also captures the packet successfully, the number of duplicates of the packet in the network is increased by one. Therefore the maximum number of duplicates that may be generated along a path is the length of that path. In order to limit the number of duplicates in the network and therefore save bandwidth, a packet that has been already forwarded by a node is dropped. In addition, if the packet has not been forwarded by a node but has been seen transmitting by the next-hop node, i.e., the next-hop node has forwarded the packet already, the packet is also dropped.

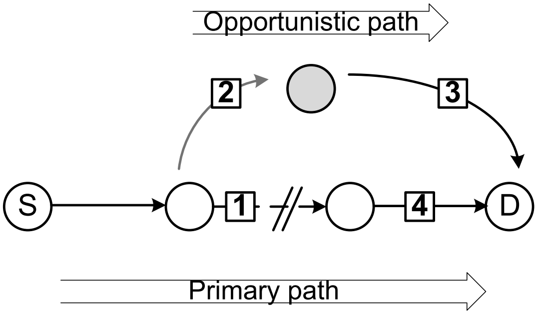

Fig. 1. Illustration of opportunistic routing when link deteriorates. 1. The local sending node detects short term deterioration of next-hop link and tags a collaborator in the packet header 2. The collaborator receives the packet via opportunistic listening 3. The packet is forwarded towards the destination by the collaborator 4. If the next-hop node on the primary path receives the unicast packet it will forward the packet as per usual.

3.5 Approach Two: Distributed Opportunistic Multipath Routing (DOMR)

Compared to WIMOP, DOMR provides a distributed approach to opportunistic multipath routing. The omission of source routing and exhaustive search of paths means that DOMR has a lower requirement for computational power and thus better scalability.

In DOMR each node keeps information about every flow it has participated in during the last *Tout* seconds. The timeout value is used to purge information of stale flows. Two parameters are set by the source: N_p, the redundancy factor, and N_f, the number of potential paths.

We classify forwarders into *primary* and *opportunistic*, each with a different forwarding behavior.

Forwarder – Primary

The *primary forwarders* are the forwarders along the primary path computed by the source. At each primary forwarder, the best next-hop may not be the same as the one identified by the source due to fluctuation in link qualities. Therefore a primary forwarder should dynamically switch to a better link, this is done using algorithm 1.

Forwarder – Opportunistic

Opportunistic forwarders are identified by the primary forwarder in the packet header. When an opportunistic forwarder captures a packet it will forward the packet using its best metric path with a probability given in the forwarder list. The forwarded packet will contain a new forwarder list and forwarding probabilities computed using local information. By changing N_p and N_f as a packet travels downstream, we can control the degree of redundancy and whether to further branch out packet transfer ($N_f > 1$).

Each node prepares routing by listing forwarders whose path costs to the destination are less than that of itself, similar to ExOR. The path with the best aggregate ETT is identified as the primary path. A WIM score is then computed between each forwarder's best ETT path and the primary path. The forwarders list is then ranked according to the WIM scores and truncated to leave the top N_f forwarders. Each forwarder i is given a forwarding probability according to the proportion of its WIM score (WIM_i) with regard to the total WIM score:

$$p_i = \frac{1/_{WIM_i}}{\sum 1/_{WIM_i}} \times N_p, \tag{8}$$

In other words, a forwarder is more likely to forward a packet if it has a lower interference path to the destination when compared to the primary path. Assuming every forwarder successfully captures the packet, the expected number of forwarders that will forward the packet is N_p, which satisfies the redundancy requirement. In order to reduce computational time, the probabilities are recomputed only if the metric of a link has changed by more than a threshold. To ensure reliability on the primary path, if p_i of the primary forwarder is less than 1, it is adjusted to 1 and then p_i of the opportunistic forwarders re-calculated using $N_p - 1$ instead of N_p. Also if N_p = N_f then some forwards may have a p_i of greater than 1. In this case we could either allow forwarders to perform more than 1 transmission of the same packet, or setting the forwarding probabilities of all forwarders to 1.

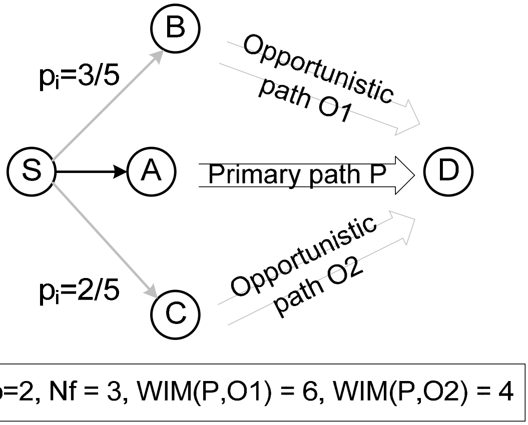

Fig. 2. DOMR forwarding probability calculation. Node A is the next-hop node on the primary path from S to D. S calculates the WIM scores for path sets (P,O1) and (P,O2) and then determines the forwarding probability for node B and C. A packet is sent from S to A and identifies B and C as opportunistic forwarders, each has a forwarding probability of 3/5 and 2/5, respectively.

By using different N_p and N_f, the source can control the number of paths a packet will take and also how frequently each path is used. For example, setting N_p and N_f both to 3 will allow 3 paths each with forwarding probability of 1 (if p_i cannot be greater than 1). While using a smaller N_p of 2 results in the same number of paths, but a lower volume of redundant traffic.

The forwarder list is included in the packet header along with other information such as N_p and N_f, and the estimated path metric to the destination from the next hop. The estimated path metric is calculated during the computation of the routing table and therefore does not incur additional computation time. The packet is then sent using unicast to the next hop.

Fig. 2 gives an example on the forwarding probability calculation in DOMR.

4 Evaluation

We evaluate our opportunistic multipath routing protocol using the NS-2 simulator. In this section we present the simulation results.

4.1 Simulation Setup

We used a modified version of NS 2.33 simulator. We have modified the link layer component to support multiple radios and multiple orthogonal channels. We use the Optimized Link State Routing (OLSR) protocol [11] to provide the basic link-state exchange framework, the Topology Control (TC) messages in OLSR now carry link information of all the interfaces in each node, rather than that of the main interface only. This allows each node to have full information of the connectivity in the network.

The network topology is generated by placing 100 nodes randomly over a 2km x 2km area. The distance between any two nodes is at least half the transmission range, and every node has at least one neighbour with which it can communicate. The topology is fully connected. The minimum distance requirement between nodes ensures that the topology is evenly spaced and there is an upper bound on the node density. Each node is equipped with two 802.11b radios tuned to orthogonal channels such that we can assume that there is no interference between the channels. The channel assignment is static and redundant, i.e. every node operates on the same pair of channels. Instead of the frame capture model available in older versions of NS, we use the new physical interface extension that supports additive SINR. The transmission rate on each link is set using the distance/rate relationship defined in Table 1. To model loss and fading in NS2 we apply the Gilbert-Elliot loss model with an average loss ratio of 0.5.

Table 1. Distance/Rate Relationship of Radios

Distance (m)	60	120	180	250
Rate (Mbps)	11	5.5	2	1

4.2 Link Quality Estimation

First we evaluate ETX-R against ETX. We performed simulations to test the performance of the metrics under interference from neighbouring nodes. 5 topologies of 50 nodes each are randomly generated. For each topology we performed 5 simulations, each consists of one main data flow and three interference flows. The traffic rate of the main flow is increased until the maximum throughput can be established. The sending rate of the interference flows were increased in 50 pkt/s increments from 0 pkt/s to 100 pkt/s. Two sets of simulations were performed, one using ETX and the other using ETX-R as the link metric. The results in Fig. 3 show that ETX-R was able to select higher-reliability paths. As discussed before, the performance of ETX depends on the broadcast interval of probe packets as well as the rate of link state information exchange. ETX-R is able to reflect quicker the change in link quality if there are enough transmissions taking place. In order for ETX to reach the same responsiveness in our scenario, the intervals would need to be much smaller than 1 second, which would result in an unacceptable increase in overhead traffic.

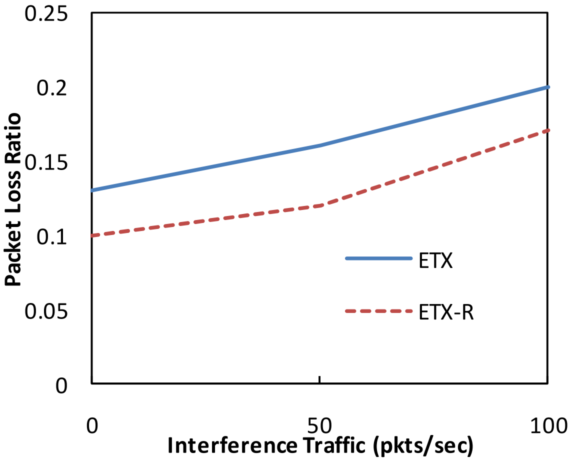

Fig. 3. Performance comparison between ETX and ETX-R

4.3 WIMOP

In this section we examine the performance improvement achieved as we add dynamic routing and opportunistic receiving to source-based multipath routing.

Using the opportunistic routing techniques described in §3.3, we modified our prior work in WIM [2] and compared it with the original version. The intervals, $T_s$and T_l, for calculating the transmission count moving averages in WIMOP-S, are set to 2 seconds and 0.5 seconds, respectively. We performed simulations using 2 and 3 paths.

Fig. 4 shows the resulting packet loss rate and end-to-end delay with a range of link layer retransmission limits. The addition of opportunistic routing to source multipath routing improved the packet loss ratio and lowered end-to-end delay. The lowered delay, in particular, is due to the fact that redirecting packets avoids incurring a large number of retransmissions at the problem link. Table 2 shows the proportion of packets that were transmitted using alternate hops due to temporary link deterioration on the primary hop.

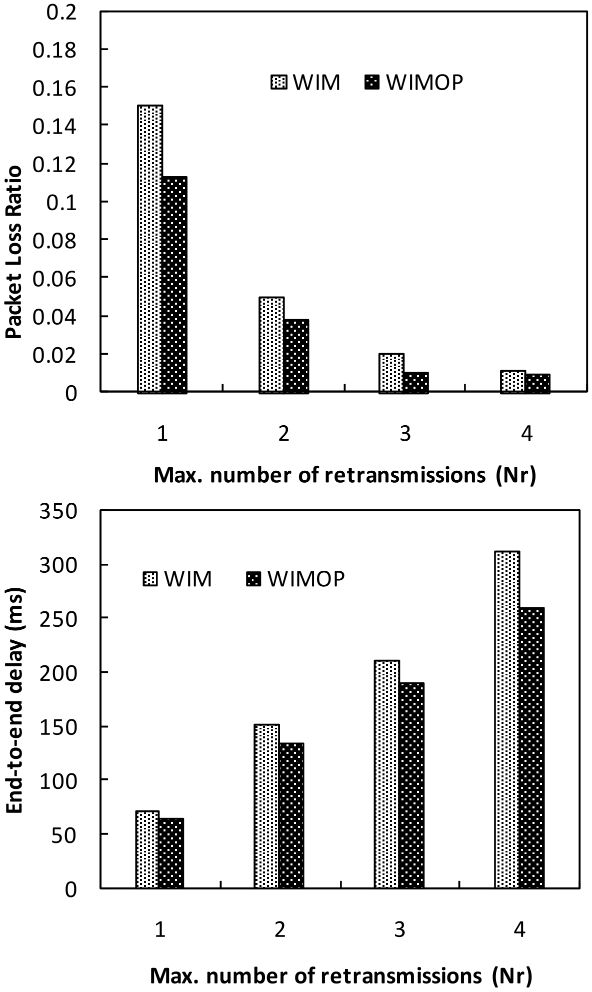

Fig. 4. Performance improvement of opportunistic source routing using WIM

Table 2. Proportion of packets rerouted

Nr	1	2	3	4
Rerouted %	7.1	9.6	14.4	21.3

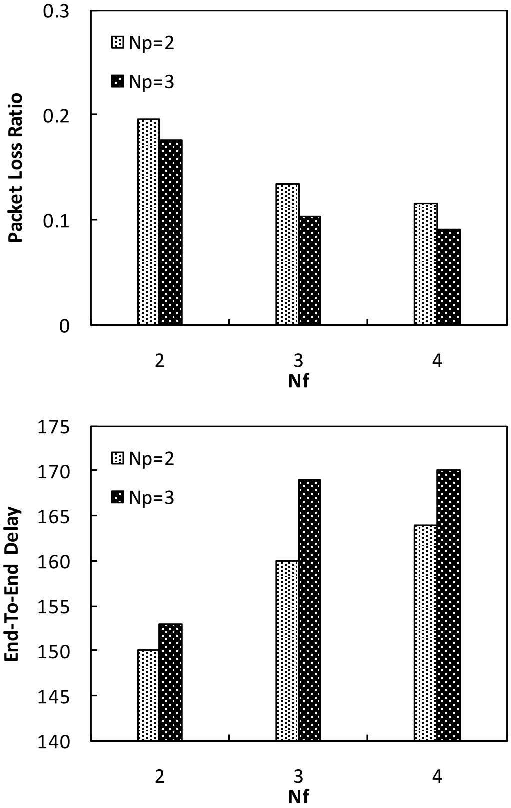

Fig. 5. Performance using different redundancy factor, Np, and number of paths, Nf

4.4 DOMR

In this section we evaluate the performance of distributed opportunistic multipath routing based on our algorithm described in §3.4. The performance of DOMR is evaluated against single path routing using WCETT [8], MORE, and sourced-based opportunistic multipath routing (WIMOP).

Since MORE is a reliable file transfer protocol, it continues to resend packets to the destination until all packets in a batch are received and acknowledged. In order to evaluate its general performance as a routing protocol, we set a limit to the number of packets the source can send for a batch of packets to *Np* times the number of packets, the redundancy factor used in DOMR. After the limit is reached, the source will move on to the next batch.

First we investigate the effect of varying the degree of redundancy, N_p, and the path diversity, N_f. Fig. 5 shows that an increased redundancy improves the packet loss ratio of the data transfer at the cost of a slight increase in delay. Fig. 5 also shows that, provided with the same degree of redundancy, increasing path diversity allows more opportunistic listening and forwarding to take place and results in better reliability.

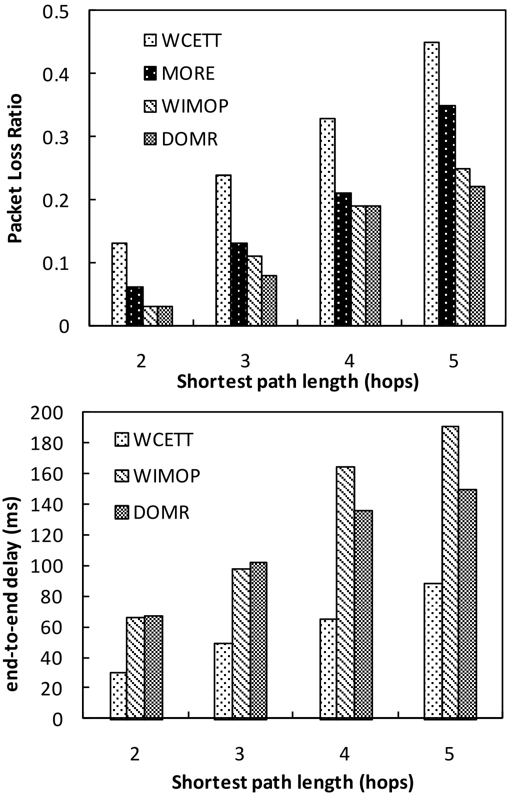

Fig. 6. Opportunistic routing performance comparison

Fig. 6 and Fig. 7 show the result of the simulations. Both WIMOP and DOMR were able to achieve better reliability than MORE. DOMR also has a slight advantage over WIMOP in both reliability and end-to-end delay. The low delay achieved by using WCETT compared to others is due to the fact that traffic is forwarded along the best metric path, however the lack of redundancy makes the packet loss ratio significantly higher. The delay for MORE is not included in the comparison since packets are coded and transmitted in batches, resulting in large overall delay and making calculation and comparison of per-packet delay difficult.

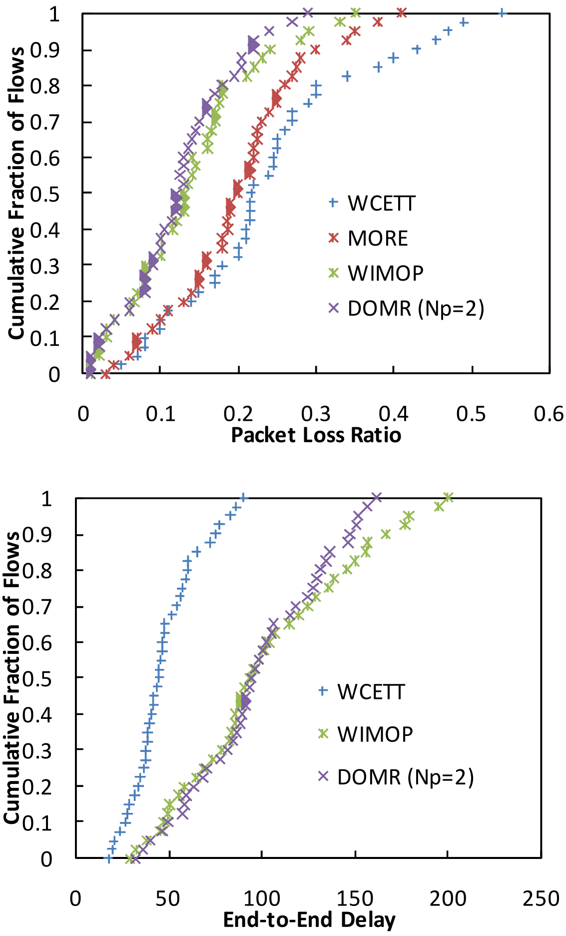

Fig. 7. Cumulative Distribution Function of Traffic Flows Performance

5 Conclusion

This paper presented two opportunistic multipath routing algorithms. The first is a modification to our prior work using source-based multipath routing. The addition of opportunistic routing techniques enables the routing to dynamically bypass links which have deteriorated since the calculation of routes at the source. The second of our proposed algorithms is completely distributed, and relies on varying each potential forwarder's forwarding probability to limit the number of duplicates in the network. We also evaluated using local link layer retransmission count to improve the accuracy of link quality estimation. Simulation results in NS2 showed that the addition of opportunistic routing further improved the reliability of source-based multipath routing. Furthermore, both approaches were able to achieve better reliability than MORE.

References

[1] Aguayo, D., Bicket, J., Biswas, S., Judd, G., Morris, R.: Link-level measurements from an 802.11b mesh network. SIGCOMM Comput. Commun. Rev. 34, 121–132 (2004)

[2] Tsai, J.W., Moors, T.: Interference-aware Multipath Selection for Reliable Routing in Wireless Mesh Networks. In: MeshTech 2007, pp. 1–6 (2007)

[3] De Couto, D.S.J., Aguayo, D., Bicket, J., Morris, R.: A high-throughput path metric for multi-hop wireless routing. In: 9th Ann. Int'l Conf. on Mobile Computing and Networking, San Diego, CA, USA (2003)

[4] Biswas, S., Morris, R.: ExOR: opportunistic multi-hop routing for wireless networks. SIGCOMM Comput. Commun. Rev. 35, 133–144 (2005)

[5] Chachulski, S., Jennings, M., Katti, S., Katabi, D.: Trading structure for randomness in wireless opportunistic routing. In: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, Kyoto, Japan. ACM, New York (2007)

[6] Dubois-Ferriere, H., Grossglauser, M., Vetterli, M.: Least-Cost Opportunistic Routing. In: Forty-Fifth Annual Allerton Conferences. Allerton House, UIUC, Illinois, USA (2007)

[7] Subramanian, A.P., Buddhikot, M.M., Miller, S.: Interference aware routing in multi-radio wireless mesh networks. In: 2nd IEEE Workshop on Wireless Mesh Networks, WiMesh 2006, pp. 55–63 (2006)

[8] Draves, R., Padhye, J., Zill, B.: Routing in multi-radio, multi-hop wireless mesh networks. In: Int'l Conf. on Mobile Computing and Networking, Philadelphia, PA, USA, pp. 114–128 (2004)

[9] Katti, S., Katabi, D., Hu, W., Hariharan, R., Médard, M.: The Importance of Being Opportunistic: Practical Network Coding For Wireless Environments. In: Forty-Third Annual Allerton Conference on Communication, Control, and Computing (2005)

[10] Katti, S., Rahul, H., Hu, W., Katabi, D., Medard, M., Crowcroft, J.: XORs in the air: practical wireless network coding. SIGCOMM Comput. Commun. Rev. 36, 243–254 (2006)

[11] Optimized Link State Routing Protocol (OLSR), `http://hipercom.inria.fr/olsr/`