International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Title: **EX-OR OPPORTUNISTIC MULTI-HOP ROUTING FOR WIRELESS NETWORKS**

Volume 06, Issue 11, Pages: 452–467.

Paper Authors

CHILUKURI SWARNA LATHA
UNIVERSITY COLLEGE OF ENGINEERING AND TECHNOLOGY, MAHATMA GANDHI UNIVERSITY, NALGONDA, TS.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

EX-OR OPPORTUNISTIC MULTI-HOP ROUTING FOR WIRELESS NETWORKS

CHILUKURI SWARNA LATHA
ASSISTANT PROFESSOR,DEPT. OF CSE, UNIVERSITY COLLEGE OF ENGINEERING AND TECHNOLOGY, MAHATMA GANDHI UNIVERSITY, NALGONDA, TS.
Swarnalatha725@gmail.com

ABSTRACT

This paper describes ExOR, an integrated routing and MAC protocol that increases the throughput of large unicast trans- fers in multi-hop wireless networks. ExOR chooses each hop of a packet's route after the transmission for that hop, so that the choice can reflect which intermediate nodes actually received the transmission. This deferred choice gives each transmission multiple opportunities to make progress. As a result ExOR can use long radio links with high loss rates, which would be avoided by traditional routing. ExOR in- creases a connection's throughput while using no more net- work capacity than traditional routing. ExOR's design faces the following challenges. The nodes that receive each packet must agree on their identities and choose one forwarder. The agreement protocol must have low overhead, but must also be robust enough that it rarely forwards a packet zero times or more than once. Finally, ExOR must choose the forwarder with the lowest remaining cost to the ultimate destination. Measurements of an implementation on a 38-node 802.11b test-bed show that ExOR increases throughput for most node pairs when compared with traditional routing. For pairs between which traditional routing uses one or two hops, ExOR's robust acknowledgments prevent unnecessary retransmissions, increasing throughput by nearly 35%. For more distant pairs, ExOR takes advantage of the choice of forwarders to provide throughput gains of a factor of two to four.

Categories and Subject Descriptors

C.2.2 [Computer Communications Networks]: Network Protocols—Routing Protocols ; C.2.1 [Computer Commu- nications Networks]: Network Architecture and Design— Wireless Communication

General Terms

Algorithms, Experimentation, Performance

Keywords

wireless, mesh, 802.11.

1. INTRODUCTION

Multi-hop wireless networks typically use routing tech- niques similar to those in wired networks [15, 16, 9, 4, 5]. These traditional routing protocols choose the best sequence of nodes between the source and destination, and forward each packet through that sequence. In contrast, cooperative diversity schemes proposed by the information theory com- munity [20, 14] suggest that traditional routing may not be the best approach. Cooperative diversity takes advantage of broadcast transmission to send information through multi- ple relays concurrently. The destination can then choose the best of many relayed signals, or combine information from multiple signals. These schemes require radios capable of simultaneous, synchronized repeating of the signal [18], or additional radio channels for each relay [11]. This paper describes ExOR, an integrated routing and MAC technique that realizes some of the gains of cooper- ative diversity on standard radio hardware such as 802.11. ExOR broadcasts each packet, choosing a receiver to forward only after learning the set of nodes which actually received the packet. Delaying forwarding decisions until after recep- tion allows ExOR to try multiple long but radio lossy links concurrently, resulting in high expected progress per trans- mission. Unlike cooperative diversity schemes, only a single ExOR node forwards each packet, so that ExOR works with existing radios. The key challenge in realizing ExOR is ensuring that only the "best" receiver of each packet forwards it, in order to avoid duplication. ExOR operates on batches of packets in order to reduce the communication cost of agreement. The source node includes in each packet a list of candidate forwarders prioritized by closeness to the destination. Re- ceiving nodes buffer successfully received packets and await the end of the batch. The highest priority forwarder then broadcasts the packets in its buffer, including its copy of the "batch map" in each packet. The batch map contains the sender's best guess of the highest priority node to have re- ceived each packet. The remaining forwarders then transmit in order, but only send packets which were not acknowledged in the batch maps of higher priority nodes. The forwarders continue to cycle through the priority list until the desti- nation has 90% of the packets. The remaining packets are transferred with traditional routing. Measurements of an ExOR implementation on a 32-node 802.11b test-bed show that ExOR performs better than tra- ditional routing for almost all node pairs, typically boosting end-to-end throughput by a factor of two. The paper in- vestigates the conditions under which ExOR performs well, and the reasons for that performance. This paper contributes the first complete design and im- plementation of a link/network-layer diversity routing tech- nique that uses standard radio hardware. It demonstrates a substantial throughput improvement and provides insight into the sources of that improvement. The rest of the paper is organized as follows. Section 2 describes the basic idea behind ExOR. Section 3 presents ExOR's design, followed by Section 5, which presents an evaluation of ExOR's performance. Section 6 describes re- lated work, and Section 7 concludes.

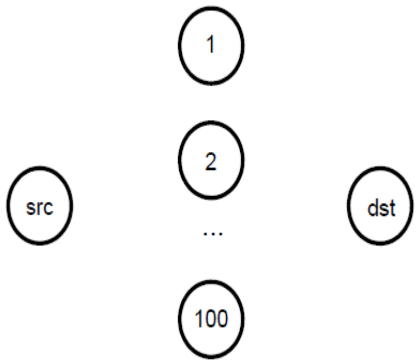

Figure 1: Example in which each of the source's transmissions has many independent chances of be- ing received by an intermediate node.

2. BASIC IDEA

A simplified version of ExOR might work as follows. A source node has a packet that it wishes to deliver to a distant destination. Between the source and destination are other wireless nodes willing to participate in ExOR. The source broadcasts the packet. Some sub-set of the nodes receive the packet. The nodes run a protocol to discover and agree on which nodes are in that sub-set. The node in the sub-set that is closest to the destination broadcasts the packet. Again, the nodes that receive this second transmission agree on the closest receiver, which broadcasts the packet. This process continues until the destination has received the packet. Why might ExOR provide more throughput than tradi- tional routing? One reason is that each transmission may have more independent chances of being received and for- warded. Consider the contrived scenario in Figure 1. The delivery probability from the source to each intermediate is only 10%. The delivery probability from each intermediate to the destination is 100%. Traditional routing would route all the data through the same intermediate; the high loss rate would require each packet to be sent an average of ten times before being received by the intermediate, once more to reach the destination, for a total throughput of 0.09 times the nominal radio speed. ExOR would achieve a through- put of roughly 0.5, since each of the source's transmissions is likely to be received by at least one intermediate. Another reason why ExOR might improve throughput is that it takes advantage of transmissions that reach unex- pectedly far, or fall unexpectedly short. Consider Figure 2, in which the source is separated by a chain of nodes leading towards the destination. Delivery probability decreases with distance. Traditional routing would forward data through some sub-sequence of the chain, for example src-B-D-dst. If a packet transmission from the source falls short of B, reaching only A, then that transmission is always wasted in traditional routing, and the source must re-send the packet. If a transmission reaches farther than B, for example all the way to D, traditional routing cannot make use of that luck. ExOR, in contrast, can often take advantage of both of these situations. In the former case, A will re-send the packet, al- lowing it to make some progress. In the latter case, D will forward the packet, eliminating one transmission. These situations are likely to be common, since traditional rout- ing must compromise when choosing hops: they must be long-distance enough to make good progress (and thus have noticeable loss rates), but short enough that the loss rate is low (thus leaving many more distant nodes with non-zero delivery probabilities).

Figure 2: Example in which the source's transmissions may make different amounts of progress towards the destination.

The above arguments assume that reception at different nodes is independent, and that there is a gradual falloff in delivery probability with distance. Whether these assump- tions hold depends on the particulars of the propagation and interference environment. For example, ExOR will work better with interference localized at each receiver than with global interference. The measurements in Section 5 suggest that the assumptions are sufficiently true in at least one real environment. ExOR is likely to increase total network capacity as well as individual connection throughput. It transmits each packet fewer times than traditional routing, which should cause less interference for other users of the network and of the same spectrum.

3. DESIGN

ExOR's design faces four key challenges. First, the nodes must agree on which sub-set of them received each packet. Since agreement involves communication, the agreement pro- tocol must have low enough overhead that it doesn't over- whelm ExOR's potential throughput gain. The protocol must also be robust enough in the face of packet loss that disagreement and thus duplicate forwarding are rare. Second, among the nodes that receive a packet, the node "closest" to the ultimate destination should be the one that forwards the packet. Thus ExOR must have a metric re- flecting the likely cost of moving a packet from any node to the destination. Third, in a large dense network there is a penalty to us- ing too many nodes as potential forwarders, since the costs of agreement grow with the number of participants. Thus ExOR must choose only the most useful nodes as partici- pants.The transmission tracker records the measured rate at which the currently sending node is sending, along with the expected number of packets it has left to send. The node uses this information to adjust the forwarding timer. The batch map indicates, for each packet in a batch, the highest-priority node known to have received a copy of that packet.

3.2 Packet Format

Figure 3 outlines ExOR's packet header format. The ExOR header follows the Ethernet header, and is followed by the packet's data. All ExOR packets are broadcasts. The Ver field indicates the current ExOR version, in case of fu- ture protocol changes. The HdrLen and PayloadLen fields indicate the size of the ExOR header and payload respec- tively. The BatchID field indicates which batch the packet belongs to. The PktNum is the current packet's offset in the batch. This offset corresponds to the batch map entry for the packet. The BatchSz indicates the total number of pack- ets in the batch. FragSz indicates the size of the currently sending node's fragment (in packets), and FragNum is the current packet's offset within the fragment. The FwdList- Size field specifies the number of forwarders in the list, and the ForwarderNum is the current sender's offset within the list. The Forwarder List is a copy of the sender's local for- warder list. The source and destination are specified in the forwarder list. The Batch Map is a copy of the sending node's batch map; in order to save space, each entry is an index into the Forwarder List rather than a full IP address.

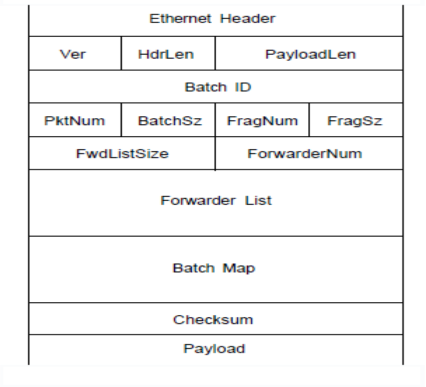

Figure 3: ExOR packet header format.

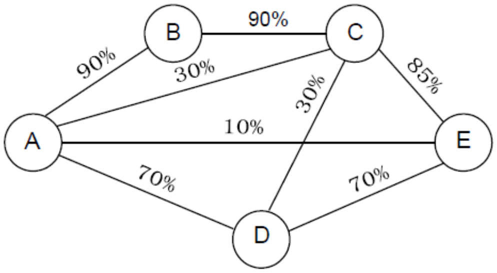

Figure 4: Example five node network with link delivery probabilities shown along the edges of the graph.

3.3 Batch Preparation

The source begins by collecting a batch of packets all des- tined to the same host. The source chooses a unique batch ID and selects a forwarder list (Section 3.4). The source prepends an ExOR header to each packet of the batch, con- taining the batch ID and forwarder list. The batch map in each header indicates that the only source has received each packet. The source indicates how many packets it will send in both the BatchSz and FragSz fields. Finally the source broadcasts each packet in the batch.

3.4 Forwarder List

The source specifies the forwarder list in priority order based on the expected cost of delivering a packet from each node in the list to the destination. The cost metric is the number of transmissions required to move a packet along the best traditional route from the node to the destination, counting both hops and retransmissions. This metric is sim- ilar to ETX [4], differing in that ExOR uses only the forward delivery probability. ExOR uses knowledge of the complete set of inter-node loss rates to calculate these ETX values. Figure 5 shows the ETX values to node E from each node in the network of Figure 4. Each node's ETX value is the sum of the link ETX values along the lowest-ETX path to E. A link's ETX value is the inverse of the link's delivery probability in the forward direction. For example, B's ETX value to node E is the sum of the ETX of link B-C (1.11) and the ETX of link C-E (1.17). If the number of nodes in a network is too large, then the expected number of a batch's packets that any given node is responsible for forwarding might be close to zero. In that case ExOR's agreement and scheduling protocols will have high overhead, since they have costs proportional to the number of nodes. For this reason the ExOR source includes only a sub-set of the nodes in the forwarder list. The source runs an ExOR simulation based on the link loss probabilities and selects only the nodes which transmit at least 10% of the total transmissions in a batch. The source chooses

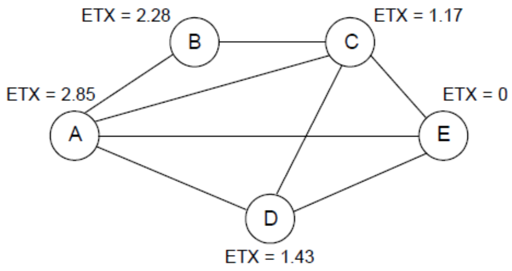

Figure 5: Estimated transmission count (ETX) to node E from each node in the sample network from Figure 4.

the forwarder list using network-wide knowledge of inter-node loss rates. The source can acquire this knowledge via periodic link-state flooding of per-node measurements. ExOR is relatively insensitive to inaccurate or out-of-date measurements, since a packet's actual path is determined by conditions at the time of transmission. In- correct measurements may degrade performance by causing the forwarder list order to be incorrect, or by causing nodes to be inappropriately included or excluded from the list.

3.5 Packet Reception

A node examines the header of every successfully decoded packet. If the forwarder list includes the node, the node adds the packet to the packet buffer for the corresponding batch. For each entry in the batch map contained in the packet, the node compares the entry with the corresponding entry in the local batch map, and replaces the latter if the packet's entry indicates a higher priority node. This batch map update algorithm and the inclusion of the sender's batch map in every transmission help the nodes ar- rive at nearly identical batch maps. The packet batch maps act as a gossip mechanism, carrying reception information from high priority nodes to lower priority nodes. The result is that a low priority node is unlikely to forward a packet that has already been received by a higher priority node.

3.6 Scheduling Transmissions

ExOR attempts to schedule the times at which nodes send their fragments so that only one node sends at a time. This scheduling allows higher-priority nodes to send first, which speeds completion and updates lower-priority nodes' batch maps. Scheduling also helps avoid collisions, which is particularly important since ExOR would like to use marginal links on which carrier sense often doesn't work. Each node waits its turn to transmit: after the source has sent the whole batch, the destination sends packets contain- ing just batch maps, then the participating nodes send in the order in which they appear in the forwarder list, high- est priority first. However, a node cannot rely on receiving the last transmission of the node just before it in the trans- mission order. Instead, a node starts sending at the time it predicts the previous fragment will finish, as indicated by the node's forwarding timer. mission tracker. The receiver remembers the last received fragment number, which it allows it to calculate how many packets must have been sent since the last received packet. This number, divided by the time since the last reception, yields the current transmission rate. The node passes the current transmission rate through an exponentially weighted moving average (EWMA) filter with α = 0.9, which deter- mines the estimated send rate. The EWMA filter smooths transient delays which would otherwise artificially inflate the estimated send rate. The node then sets the forwarding timer to be equal to the current time plus the estimated send rate times the number of packets remaining to be sent. The node determines the number of packets remaining from the fragment fields. Competing transmissions from other net- work protocols, or from other batches, inherit the fairness that the 802.11 MAC provides. The transmission tracker adapts to competing traffic because it observes a slower over- all rate of transmission from the current forwarding node.If a node has not yet heard any forwarded packets from higher priority nodes, and thus has no information in its batch map, it assumes that each higher priority node will send for five packet durations. When a node's forwarding timer elapses, it sends its batch fragment: the packets that it has received, but that its batch map indicates have not been received by any higher priority node. As a special case, when the ultimate destination's turn occurs to send, it sends ten packets that include its batch map but no data.When the lowest priority node has finished forwarding, the schedule starts again. The source re-sends any packets that its batch map indicates were not received by any node, the destination sends copies of its batch map, and each node in the forwarding set forwards any packets that have still not been received by any higher priority node.

3.7 Completion

If a node's batch map indicates that over 90% of the batch has been received by higher priority nodes, the node sends nothing when its turn comes. The last few packets in a batch would be the most expensive to send, since they would require all the overhead of running the transmission sched- ule, but the overhead would be divided among relatively few packets. In addition, if fragments are small, there is a greater likelihood that nodes will set their forwarding timers incorrectly and collide. Because ExOR only guarantees to deliver 90% of a batch, the destination requests the remaining packets via tradi- tional routing. The destination sends its batch map to the source, which then sends the remaining packets with tra- ditional routing, which uses link-level acknowledgments to ensure reliable delivery.The source begins sending a new batch when 90% of the batch map entries from the current batch are filled with the IDs of higher priority nodes. The source waits until it stops hearing packets from the current batch.

3.8 Example

Figure 6 shows an example transmission time-line for the first six seconds of an ExOR transfer. The data is derived by monitoring a transfer from the experimental evaluation in Section 5. Node N5 is the source, and N24 is the destination. The figure was generated by aggregating traces of received packets. Because of delays within the radio hardware and operating system of the individual nodes, the timing reso- lution of the trace is on the order of tens of milliseconds, or roughly a single packet time. The figure shows a bar if more than four packets were sent by a node within a win- dow of 60ms, and the bar's horizontal length indicates the total duration of the transmission. The two shades of gray represent two successive batches. The source, N5, broadcasts the first batch of 100 packets with the forwarder list set to N24, N20, N18, N11, N8, N17, N13, N5 (ordered from highest to lowest priority). The destination (N24) and N20 happen to receive none of the source's packets, and thus are not yet aware of the batch's existence. N18 does receive about a dozen packets. N18 waits an interval that it thinks is long enough for the source to finish, plus 5 packet times each for N24 and N20 (which send nothing), though N18 under-estimates the time required because it misses most of N5's packets. Then N18 sends all the packets it received. Next, the remaining forwarding nodes transmit in reverse priority order. The batch maps they broadcast carry the set of received packets back towards N5, since N5 may not have heard any of N18's transmissions. When N13 is done, N5 starts a second round of transmissions, sending packets that were not received by any node in the first round. N24 starts the second round too early, at time 1.8, before N5 finishes. The reason is that N24 cannot hear N5, so it assumes N5 will send only five packets. N24 sends ten copies of its batch map. N8 does not transmit in the second round because over 90% of the packets in the batch have been acknowledged by higher priority nodes (though not yet by the destination). N17 does send a small fragment, possibly because it did not hear some of the earlier batch maps. The source sees at the end of the second round that 90% of the batch has made progress, and thus sends nothing; it delays starting the next batch while it hears packets from the first batch.Two smaller rounds ensue before the destination acknowl- edges receiving 90% of the packets. The source node begins transmission of a second batch at 3.4 seconds, since it hears from N11 that 90% of the batch has been received by higher priority nodes.Figure 6 illustrates why ExOR achieves higher throughput than traditional routing. Traditional routing must choose a first-hop node to which N5 will send. If the first hop is N18, then the loss rate will be about 90%, forcing N5 to re-send most packets many times even though N13 through N11 received most of them. If the first hop is N13, then most packets will be received at nodes closer to N24, but will be ignored. The choices in between suffer from both of these problems. ExOR performs better because it does not have to commit to a specific route.

4. IMPLEMENTATION

The ExOR implementation uses the Click toolkit [10] and runs as a user-space daemon on Linux. The daemon sends and receives raw Ethernet frames from the wireless device using a libpcap-like interface. The implementation accepts entire files, which it splits into batches of packets. ExOR does not guarantee reliable delivery, so the implementation uses traditional routing to transfer missing packets at the end of each batch.If ExOR were layered under TCP, ExOR's batches would likely interact badly with TCP's window mechanism. If the end-to-end loss rate were not very low, TCP would use a window size too small to allow ExOR to accumulate the 10 or more packets required for an efficient batch. For this reason the implementation includes a split web proxy. Browsers communicate with TCP to the proxy half on the client side of the wireless network. The proxy half on the Internet side of the wireless network fetches files with TCP from Internet web servers. The two halves use ExOR to transfer any files larger than 100 kilobytes over the wireless network between them.

5. EVALUATION

This section presents experimental results which show that ExOR delivers bulk data faster than traditional routing, for both long and short routes. It also examines some of the individual design decisions in the ExOR protocol, explores the consistency of ExOR's performance and identifies areas for improvement.

5.1 Network Description

The evaluation was performed on Roofnet [1], an outdoor roof-top 802.11b network. Roofnet consists of 38 nodes dis- tributed over roughly six square kilometers of Cambridge, Massachusetts. Each of the nodes is a PC with an 802.11b card connected to a roof mounted omni-directional antenna. The physical locations of all the nodes are show in in Fig- ure 7. The area is dominated by tightly-packed three and four story houses; most of the antennas are mounted about two or three feet above the chimneys of the houses. There are a number of taller buildings in the area and five of the Roofnet nodes are mounted on these buildings, which are located along the perimeter of the network. A handful of the nodes have antennas mounted in windows. All but four of the nodes use the Intersil Prism 2.5 802.11b chip-set. The transmit power level is 200mW. The other four nodes use Atheros AR5212 chip-sets with a transmit power level of 100mW. The radios operate in "pseudo-IBSS" mode, a simplified version of the 802.11 IBSS (ad-hoc) mode which does not use beacons.

5.2 Method

Each experiment measures throughput between 65 ran- domly selected node pairs. First, the nodes broadcast 1500- byte packets every ten seconds for ten minutes and report the measured delivery probabilities from all other nodes to a central server. The server distributes this information to all the nodes. The measurements are used to compute ETX metrics and traditional routes. Next, the server con- tacts each of the 65 node pairs in sequence, telling the pair to measure the time required to transfer a 1.0 megabyte file using traditional routing, then to wait 15 seconds, then

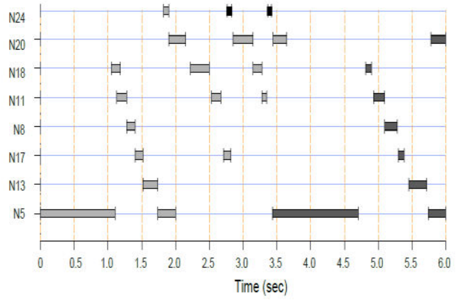

Figure 6: Transmission time-line for an ExOR transfer from N5 to N24. Nodes higher on the Y axis have lower ETX metrics to N24. The light gray bars show the transmissions in the first batch. The darker gray bars show part of the second batch.

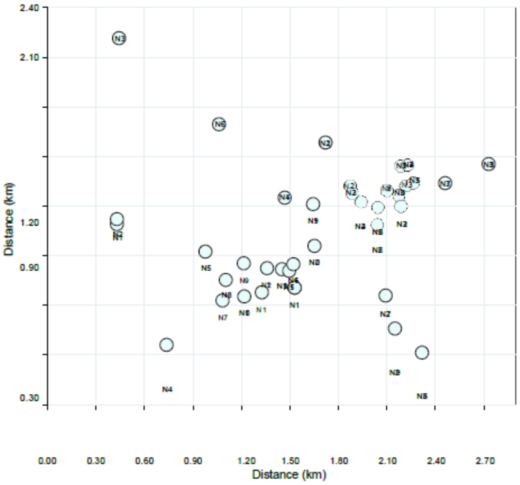

Figure 7: Physical layout of the 38 Roofnet nodes which participated in the performance evaluation.

to measure the time required to transfer 1.1 megabytes us- ing ExOR. The evaluation does not use the combination of ExOR and traditional routing, so the extra 0.1 megabyte is to compensate for the 10% of packets which may not have been delivered ordinarily. The reported throughput is one megabyte divided by the total time required to transfer the data. Every twenty minutes, the central server suspends the experimental runs to recollect the link loss measurements. During the experiment, existing Roofnet routing and user traffic are present.The ExOR batch size is 100 packets, except for the exper- iments in Section 5.5 which consider batch sizes of 10 and250 packets. Each packet contains 1024 byte of payload data and either a traditional routing header or an ExOR header. Traditional headers vary between 24 and 48 bytes, depend- ing on the number of hops. ExOR headers vary between 44 and 114 bytes, depending on the forwarder list size. All the packets are sent with the 802.11b one megabit/second bit-rate. It's likely that a higher bit-rate would provide higher throughput, but selection of the best rate remains as future work. The traditional route is chosen using the ETX metric, which has been shown to find the best routes [4, 5] when the link loss measurements are accurate. The traditional routing traffic is sent along pre-computed source-routed paths. Each node in the route sends the entire file to the next node before the next node starts sending, so that only one node sends at a time. This is done to make a fair comparison of the two protocols, since traditional multi-hop traffic tends to traditional routing data packet is sent using 802.11 unicast, so that 802.11 keeps re-sending a lost packet until the sender gets an 802.11 acknowledgment from the next hop. To reduce the effect of interference from Roofnet user traf- fic and other sources, the reported values are the median of nine experimental runs. The exception is Section 5.7, which studies the variations between experimental iterations.In addition to throughput measurements, the nodes col-lect the received headers and arrival times of all packets of a single iteration of the experiment. These traces are centrally processed to reconstruct the state of the wireless channel, providing the trace data for the case studies in Sections 3.8 and 5.4.

5.3 End-to-End Performance

Figure 8 compares the throughput CDFs of ExOR and traditional routing for the 65 node pairs. ExOR's through- put is 33 KBytes/sec for the median pair, whereas tradi- tional routing achieved 11 KBytes/sec for the median pair.

5.3.1 The 25 Highest Throughput Pairs

ExOR's throughput advantage varies with the number of nodes between the source and destination. Figure 9 com- pares the 25 highest throughput pairs. The top five pairs and the pair N13-N7, near the center of the figure, correspond to single hop traditional routes. For these pairs ExOR provides higher throughput even though it too sends most packets di- rectly from source to destination. Traditional routing relies

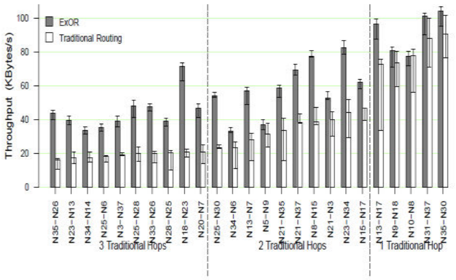

Figure 9: The 25 highest throughput pairs, sorted by traditional routing throughput. The bars show each pair's median throughput, and the error bars show the lowest and highest of the nine experiments. ExOR's improvement in throughput is smaller for shorter routes.

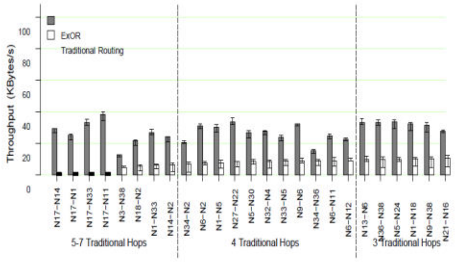

Figure 10: The 25 lowest throughput pairs. The bars show each pair's median throughput, and the error bars show the lowest and the highest of the nine experiments. ExOR outperforms traditional routing by a factor of two or more.

on 802.11 ACKs to trigger re-sends, but this mechanism in- correctly re-sends the data packet if the ACK is lost. In contrast, the ExOR destination sends each batch map ten times, so that the source is unlikely to re-send incorrectly. In addition, in some cases ExOR re-sends packets from nodes that have lower-loss links to the destination than the source has.The next nine pairs to the left are routes with two tra- ditional routing hops. For most of the pairs ExOR outper- forms traditional routing by 50% or more. In these cases, ExOR has a choice of forwarding nodes as well as multi- ple ways in which batch map information can be gossiped back to the source. Some of the pairs see little improvement because of a limited number of forwarding node choices.The remaining pairs correspond to three hop routes. For most of these pairs, ExOR outperforms traditional routing by a factor of two or more. As route length increases, the likelihood of finding additional forwarding nodes increases, which gives ExOR more opportunities to make progress.

5.3.2 The 25 Lowest Throughput Pairs

Figure 10 compares the throughputs of the 25 pairs with the lowest traditional routing throughputs. These pairs are separated by three traditional hops or more, and tend to have many potential forwarders and thus benefit the most from ExOR.The biggest performance gains occur on the longest routes, which have between five and seven hops. In many cases ExOR is able to use long links that are asymmetric: that deliver many data packets but few packets in the reverse di- rection. In these cases the ExOR batch maps flow back to- wards the source by a different path than the data packets. Traditional routing, in contrast, avoids these asymmetric links because ETX takes link-level ACK losses into account.

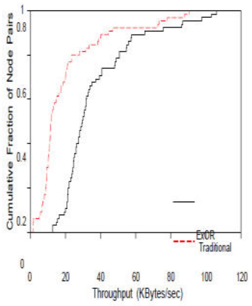

Figure 8: The distribution of throughputs of ExOR and traditional routing between the 65 node pairs. The plots shows the median throughput achieved for each pair over nine experimental runs. ExOR provides three times as much throughput as traditional routing for the median pair.

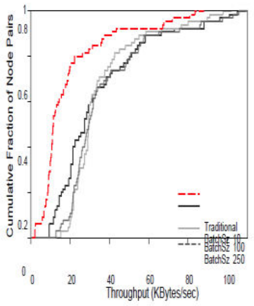

Figure 13: CDF of ExOR throughput for three batch sizes. All of the sizes outperform traditional routing. Large batches work well for low-throughput pairs because of redundant batch map transmissions, while smaller batches work well for high-throughput pairs due to lower header overhead.

5.4 Distance per Transmission

One of the potential reasons that ExOR works well is that delivery probability may decrease gradually with distance, so that significant numbers of packets travel less or more far than expected. Figures 11 and 12 illustrate the extent to which this is true for a transfer from node N5 to node N24. This is the same transfer explored in Section 3.8.

Each bar in Figures 11 shows the number of transmissions a particular node makes, with the node's ETX metric to N24 plotted on the X axis. Bars higher than 1000 indicate nodes that had to transmit some packets more than once. For ex- ample, the traditional routing source node had to send each packet an average of over three times. The ExOR source node sent each packet about half as many times. The rea- son is that many of the ExOR transmissions that failed to reach the first traditional hop were received by nodes closer to the source (i.e. with X positions to the left of 5); ExOR al- lows these packets to make some progress, while traditional routing does not.Figure 12 shows that ExOR makes good use of packets that travel farther than expected. The graph indicates the distance that each transmission traveled, as measured in difference in ETX metric to N24 between the transmitter and receiver. The white bars indicate that the four hops in the traditional route traveled ETX distances of up to about2.6. Many ExOR transmissions traveled farther than 2.6. While the number of packets carried by each individual long- distance link is small, the sum is substantial. The resulting decrease in total transmissions required helps explain why ExOR increases throughput.

5.5 Batch Size

ExOR's batches increase the chances that repeated in- formation such as batch maps and fragment counters reach all the nodes, so that they make consistent scheduling and transmission decisions. However, the ExOR header grows with the batch size, and many transfers may only have a few packets. This section examines how ExOR throughput varies with batch sizes of 10, 100 and 250 packets.The batch size affects the per-packet overhead due to the batch map embedded in the ExOR header. Given a rel- atively large forwarder list of 14 nodes, each entry in the batch map occupies 4 bits. Thus, for a 10 packet batch the total batch map requires five bytes of space. For a batch size of 100, the batch map occupies 50 bytes, and for a 250 packet batch, the map occupies 125 bytes. In the experiment, the per-packet payload is fixed at 1024 bytes, so the 250-batch map creates 12% more per-packet overhead. The through- put is calculated as the total amount of unique payload data delivered divided by the total transfer time.

Figure 13 shows the throughput distribution for each of the three batch sizes. The evaluation method is similar to the experiments presented earlier in the section, except that each experiment was repeated only three times. The graph plots the median for each node pair.For low-throughput pairs the batch size of ten performs about 20% worse than 100 or 250. Smaller batches reduce the chance that all nodes hear previous senders' batch maps, and thus increase the chance of needless transmissions. For high-throughput node pairs with direct communication, the ExOR header overhead penalizes large batches. 250-packet batches result in roughly 15% slower performance when com- pared to 10 or 100 packet batches. The fact that 10 and 100 perform equally well suggests that the best batch size is somewhere between those two values.

5.6 Independent Loss Simulation

ExOR’s design assumes that the majority of packet loss is uncorrelated among receivers. If all losses were due to low signal-to-noise ratio or multi-path fading, such an assumption would be true. Packet losses caused by background traffic or interference, in contrast, might be correlated among the receivers. This section evaluates the impact on ExOR throughput of correlated losses.

To study the effects of shared interference, it is necessary to use a simulator, as such interference sources are difficult to produce on the test-bed. The simulator takes the measured link loss rates as input. When a packet is broadcast with independent loss, the simulator models each link (each receiver) with a separate random variable. For dependent loss, all the links are conditioned on the same random variable. For example, given two links with a 50% and a 75% loss rate, the first receiver will receive a strict superset of the packets received by the second receiver. The simulator does not model contention or any other MAC-related delays, so it produces an optimistic result.

Figure 14 illustrates the simulation results. For single hop routes, there is no difference between dependent and independent losses, as there is only a single link. As the pairs become distant, a performance gap develops, in which the dependent curve lags the independent curve by 20% for the median pair.ExOR forwarder lists contain nodes at various distances between the source and destination, producing a wide range of inter-node loss rates. Even if all losses were correlated, some transmissions would deliver packets farther than others, allowing ExOR to exploit the lucky transmissions. Thus, ExOR does not require independent losses, but does take advantage of them when available.

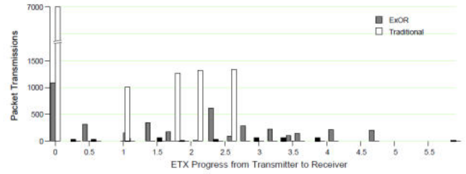

Figure 12: Distance traveled towards N24 in ETX space by each transmission. The X axis indicates the difference in ETX metric between the sending and receiving nodes; the receiver is the next hop for traditional routing, and the highest-priority receiving node for ExOR. The Y axis indicates the number of transmissions that travel the corresponding distance. Packets with zero progress are not received by the next hop (for traditional routing) or by any higher-priority node (for ExOR).

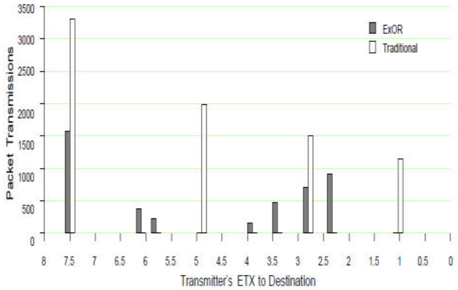

Figure 11: The number of transmissions made by each node during a 1000-packet transfer from N5 to N24. The X axis indicates the sender’s ETX metric to N24. The Y axis indicates the number of packet transmissions that node performs. Bars higher than 1000 indicate nodes that had to re-send packets due to losses.

5.7 Throughput Variation

A side-effect of ExOR’s use of multiple forwarding nodes is a reduction in variation between per-transfer throughputs. Table 1 shows the variation among the nine experimental iterations for 20 randomly selected node pairs. The throughput columns are the median of the nine runs and the range column is the difference between the highest and lowest throughputs, expressed as a percentage of the median. Traditional routing throughput tends to vary by eight to ten times as much as ExOR.It might seem surprising that the throughputs of successive transfers should vary substantially, since each transfer effectively averages thousands of individual packet transmissions. The variation is caused by changes in link delivery rates at time scales comparable to a single transfer.

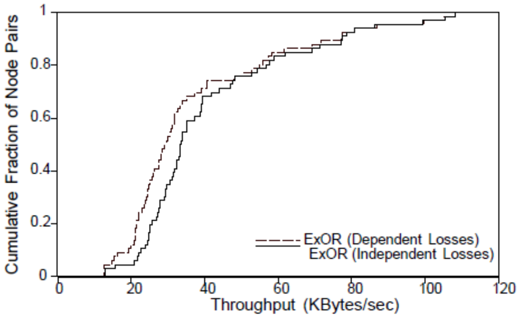

Figure 14: CDF of simulated end-to-end ExOR throughput for independent and dependent losses. Overall throughput is 20% less when losses are dependent.

Node Pair	ExOR (KB/s)	ExOR Range	Traditional (KB/s)	Traditional Range
N17-N33	33.1	3.9%	1.4	55.6%
N18-N2	21.8	3.6%	5.5	63.8%
N34-N2	20.7	1.6%	6.9	90.4%
N27-N22	33.7	4.3%	8.5	41.2%
N33-N5	23.8	3.2%	8.9	44.0%
N6-N11	24.8	3.0%	9.1	59.6%
N36-N38	33.4	3.3%	10.0	68.1%
N9-N38	31.7	5.9%	10.4	63.9%
N4-N33	24.6	4.0%	10.9	38.1%
N11-N30	31.5	3.1%	11.3	44.2%
N19-N16	29.3	3.8%	11.7	71.4%
N13-N34	35.1	4.1%	12.6	22.7%
N33-N20	29.7	1.8%	14.4	65.1%
N23-N13	39.4	4.8%	17.2	39.4%
N3-N37	39.1	6.5%	19.3	10.2%
N28-N25	39.3	4.8%	20.6	56.1%
N25-N30	54.2	3.1%	23.4	9.7%
N5-N9	37.2	5.9%	31.4	43.7%
N8-N15	77.3	4.0%	38.8	26.5%
N15-N17	62.0	5.5%	46.8	15.7%
N10-N8	77.2	8.5%	78.0	32.9%

Table 1: Comparison of traditional and ExOR throughputs and variation for 20 node pairs.

in a single link can have a large effect on a traditional route that happens to use that link. ExOR's throughput is less sensitive to single links, so that ExOR's experiment-to-experiment throughput would only vary if many links simultaneously changed quality.

6. RELATED WORK

6.1 Opportunistic Channel Protocols

The Opportunistic Auto Rate (OAR) protocol [17] exploits durations of high quality channel conditions caused by movement. Wireless channels between pairs of nodes tend to change over time, but if there are multiple nodes waiting to send there is likely to be one good channel at any given time. OAR attempts to identify the best channel and let that pair of nodes send without interruption, while maintaining fairness between contending senders. OAR works best if channels are stable enough that it can send many packets between changes in channel conditions. OAR determines the channel conditions by measuring the received signal strength of 802.11 control packets. The source node sends an RTS packet when it wishes to initiate a transmission. The returned CTS packet includes the received signal strength of the RTS, which can be used to determine the best bit rate to send at, as in the RBAR protocol [7]. OAR then sends a burst of packets, monitoring the signal strength information of the link-level acknowledgments to ensure the channel is still good.Like OAR, ExOR uses a channel reservation scheme to avoid collisions. However, ExOR requires less channel stability, since it has no RTS/CTS exchange, and ExOR does not require signal strength measurements to predict reception. Finally, ExOR takes advantage of intermediate nodes to relay packets.

6.2 Opportunistic Forwarding

A number of protocols have considered the problem of choosing forwarding hops based on channel conditions. Larsson presents the idea of "selection diversity forwarding" [13], in which the source includes a list of potential forwarders' node addresses in the RTS packet. Neighboring nodes which successfully receive the packet respond with CTS packets containing the signal to noise ratio of the RTS, and the source node chooses a forwarder based on guidelines from the routing layer and reported RTS S/N. The candidate for- warding nodes are likely to send CTS frames simultaneously, potentially causing collisions. Jain, et al. [8]. propose an im- provement to the protocol in which the forwarders respond in a priority order specified in the initial RTS. Upon receiv- ing the first reply, the source immediately begins transmis- sion to that node (regardless of S/N), reducing the overhead associated with waiting for multiple replies. Roy Chowdhury presents an alternate protocol [3] which avoids the control packet overhead by using historical observations of channel conditions. Similarly, GeRaF [21] uses an RTS/CTS-based receiver contention scheme to select the best of many po- tential forwarders, but prioritizes forwarders based on geo- graphic distance instead of S/N.In all four protocols, it is assumed that channel measure- ments or distance accurately predict whether packets are likely to be delivered. ExOR, in contrast, determines the forwarding node based on reception of data packets rather than preceding control packets. Link-level measurements in [1] have shown that delivery probability is hard to predict by signal to noise ratio or distance measurements.

6.3 Multiple Path Routing

Ganesan's braided multi-path routing [6] identifies multi- ple routes, using one as a primary and switching if the pri- mary fails. Opportunistic Multipath Scheduling (OMS) [2] splits traffic over multiple paths, adaptively favoring paths that provide low delay or high throughput. Tsirigos and Haas [19] propose sending erasure-coded fragments of each packet over disjoint paths in a mobile ad-hoc network, in order to tolerate loss of some fragments due to fading or node movement. ExOR also exploits multiple paths, but uses broadcast to explore them simultaneously and to use long and high-loss radio links. ExOR does not need to iden- tify specific paths in advance, nor must it ensure that the paths are disjoint.

6.4 Cooperative Diversity Routing

Laneman and Wornell develop and analyze a series of information theoretic cooperative diversity techniques to exploit nearby nodes which overhear transmissions. In their protocols, all nodes "closer" to the destination relay a copy of the packet. Closeness is determined by S/N; nodes which are closer to the destination are likely to have higher S/N than the source. However, because the relay nodes do not communicate with each other before forwarding, duplicate transmissions are likely to occur in dense networks, potentially wasting spectrum. Laneman's protocols assume or- thogonal channels or fixed time slots, which are difficult to implement with commodity radio hardware.

ExOR fills in some of the details needed to make coopera- tive diversity efficient and practical on commodity hardware, such as scheduling transmissions and agreeing on the best node to relay each packet.

7. CONCLUSION

This paper presents ExOR, an integrated routing and MAC protocol for multi-hop wireless networks in which the "best" of multiple receivers forwards each packet. ExOR improves performance by taking advantage of long-distance but lossy links which would otherwise have been avoided by traditional routing protocols. The result is a factor of two to four improvement in throughput between distant pairs of nodes in a real test-bed.Potential areas of future work include extending the pro- tocol to use multiple transmit bit rates, better integration with TCP, and taking advantage of received frames which contain bit-errors.

8. REFERENCES

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In ACM SIGCOMM 2004, August 2004.

[2] C. Cetinkaya and E. Knightly. Opportunistic traffic scheduling over multiple network paths. In Proc. IEEE Infocom. IEEE, September 2004.

[3] R. Roy Choudhury and N. Vaidya. MAC layer anycasting in wireless networks. In Second workshop on Hot Topics in Networks (HotNets II), November 2003.

[4] D. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In Proc. ACM/IEEE MobiCom, September 2003.

[5] R. Draves, J. Padhye, and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. In Proc. ACM SIGCOMM Conference (SIGCOMM 2004), September 2004.

[6] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. ACM Mobile Computing and Communications Review, 5(4), October 2001.

[7] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In ACM Mobicom 2001, September 2001.

[8] S. Jain and S. Das. Exploiting path diversity in the link layer in wireless ad hoc networks. In Proc. of the 6th IEEE WoWMoM Symposium, June 2005.

[9] D. B. Johnson. Routing in ad hoc networks of mobile hosts. In Proc. of the IEEE Workshop on Mobile Computing Systems and Applications, pages 158–163, December 1994.

[10] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. The Click modular router. ACM Transactions on Computer Systems, 18(3):263–297, August 2000.