

# Opportunistic Routing in Multi-Hop Wireless Networks

by

Sanjit Zubin Biswas

B.S., Stanford University (2002)

Submitted to the Department of Electrical Engineering and Computer Science  
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

March 2005

© Massachusetts Institute of Technology 2005. All rights reserved.

Author.....  
Department of Electrical Engineering and Computer Science  
March 14, 2005

Certified by .....  
Robert Morris  
Associate Professor  
Thesis Supervisor

Accepted by.....  
Arthur C. Smith  
Chairman, Department Committee on Graduate Students

# Opportunistic Routing in Multi-Hop Wireless Networks

by

Sanjit Zubin Biswas

Submitted to the Department of Electrical Engineering and Computer Science  
on March 14, 2005, in partial fulfillment of the  
requirements for the degree of  
Master of Science

## Abstract

This thesis describes ExOR, an integrated routing and MAC protocol for bulk transfers in multi-hop wireless networks. ExOR exploits the broadcast nature of radios by making forwarding decisions based on which nodes receive each transmission. The spatial diversity among receivers provides each transmission multiple opportunities to make progress in the face of packet losses. As a result ExOR can use long links with high loss rates, which would be avoided by unicast routing.

ExOR operates on batches of packets. The source node includes a list of candidate forwarders in each packet, prioritized by closeness to the destination. Receiving 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 received each packet. The remaining forwarders then transmit in order, sending only packets which were not acknowledged in the batch maps of higher priority nodes. The forwarders continue to cycle through the priority list until the destination has enough packets to recover the original data using forward error correction.

An evaluation of an implementation on a 38-node 802.11b test-bed shows that ExOR improves bulk data transfer throughput for most node pairs when compared with unicast routing. For pairs between which unicast uses one or two hops, ExOR’s robust batch maps prevent unnecessary retransmissions, increasing throughput by nearly 50%. For longer unicast routes, ExOR takes advantage of spatial diversity, providing gains of a factor of two to four when using a batch size of 10 packets.

Thesis Supervisor: Robert Morris

Title: Associate Professor

## Acknowledgments

This thesis is the result of joint work with Robert Morris, John Bicket and Dan Aguayo. I am indebted to Robert for his advice, high standards and for granting me the freedom to develop ExOR on my own. John and Dan deserve special thanks for their dedication and enthusiasm while building the Roofnet test-bed. Most of the ideas in this thesis came from late night discussions with them.

The members of PDOS provided an endless supply of entertainment and support, without which this thesis would never have been completed. Thanks to Frank, Jeremy, Jinyang, Athicha, Max, Thomer, Dave and Rodrigo, I looked forward to coming to lab every day.

Finally, thanks to Hope, Ma, Baba and Nina for their love and encouragement.

# Contents

<b>1</b>	<b>Introduction</b>	<b>8</b>
1.1	Properties of Multi-Hop Wireless Networks . . . . .	9
1.1.1	Causes of Packet Loss . . . . .	10
1.2	Measurements . . . . .	11
1.2.1	Multiple receivers per packet . . . . .	12
1.2.2	Gradual delivery ratio fall-off . . . . .	12
<b>2</b>	<b>Design</b>	<b>14</b>
2.1	ExOR Operation . . . . .	14
2.2	Forwarder Set Ordering and Selection . . . . .	20
2.2.1	Priority Ordering . . . . .	20
2.2.2	Forwarder Set Culling . . . . .	21
2.3	Batch Maps . . . . .	23
2.3.1	Batch Map . . . . .	23
2.4	Collision Avoidance . . . . .	25
2.5	Practical Considerations . . . . .	26
<b>3</b>	<b>Results</b>	<b>27</b>
3.1	Evaluation Methodology . . . . .	27
3.2	End-to-End Performance . . . . .	30
3.3	ExOR Node Pair Case Study . . . . .	33
3.4	Throughput Variation . . . . .	36
3.5	Batch Size . . . . .	37
3.6	Independent Loss Simulation . . . . .	38

<b>4</b>	<b>Related Work</b>	<b>40</b>
4.1	Opportunistic Channel Protocols . . . . .	40
4.2	Opportunistic Forwarding . . . . .	41
4.3	Multiple Path Routing . . . . .	42
4.4	Cooperative Diversity Routing . . . . .	42
<b>5</b>	<b>Conclusion</b>	<b>44</b>

# List of Figures

1-1	Simple network example, with delivery probabilities. . . . .	10
1-2	Histogram of number of neighbors . . . . .	12
1-3	Distance versus delivery ratios . . . . .	13
2-1	Example five node network. . . . .	15
2-2	ExOR packet header format. . . . .	17
2-3	Sample ExOR transmission timeline . . . . .	18
2-4	Sample unicast transmission timeline . . . . .	19
2-5	ETX for sample network . . . . .	21
2-6	ExOR transmission timeline for dense network . . . . .	22
2-7	ExOR transmission timeline for dense network, using culling . . . . .	23
2-8	Example of Batch Map update . . . . .	25
2-9	Estimator performance comparison, EWMA versus Moving Average . . . . .	26
3-1	Physical layout of the 38 Roofnet nodes which participated in the performance evaluation. . . . .	28
3-2	Overall end-to-end performance of ExOR and unicast routing . . . . .	31
3-3	The 25 highest throughput pairs . . . . .	32
3-4	The 25 lowest throughput pairs . . . . .	33
3-5	ExOR transmission timeline for transfer between N5 and N24 . . . . .	34
3-6	Aggregate view of all ExOR and unicast transmissions . . . . .	35
3-7	CDF of end-to-end ExOR throughput for various batch sizes . . . . .	37
3-8	CDF of simulated ExOR throughput for dependent and independent losses . . . . .	39

# List of Tables

3.1 Comparison of Unicast and ExOR throughputs and variation on a 20 node pair subset . . . . .	36
--	----

# Chapter 1

## Introduction

Multi-hop wireless networks typically use routing techniques similar to those found in wired networks [17, 18, 12, 5, 7]. A routing protocol chooses a path of nodes between the source and destination, and each packet is forwarded along the path through one node at a time. *Cooperative diversity* schemes proposed by the information theory community [21, 16] suggest an alternate approach that may yield higher throughput. Cooperative diversity takes advantage of broadcast transmission to send information through multiple 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 analog signal [20], or additional radio channels for each relay [14].

This thesis describes ExOR, an integrated routing and MAC technique that aims to realize some of the gains of cooperative diversity on standard radio hardware such as 802.11. ExOR broadcasts each packet to multiple potential forwarders, choosing one of them only after learning the set of nodes which actually received the packet. Delaying forwarding decisions until after reception allows ExOR to try multiple long but radio lossy links concurrently, resulting in high expected progress per transmission. Unlike cooperative diversity schemes, only a single ExOR node forwards each packet, making ExOR suitable for use 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. To help the receivers agree with as little communication overhead as possible ExOR operates on batches of packets. The source node includes in each packet a list of candidate forwarders prioritized by closeness to destination.

Receiving 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 received 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 destination has enough packets to recover the original data using forward error correction.

Measurements of an ExOR implementation running on an 802.11b test-bed with 38 nodes demonstrate that ExOR performs better than conventional unicast routing for almost all node pairs, typically boosting end-to-end throughput by a factor of two. The thesis presents an evaluation of each of ExOR’s major mechanisms to explore why the protocol works well. ExOR performs well when transferring batches as small as ten packets.

This thesis contributes the first complete design and implementation of a link/network-layer diversity routing technique that operates on standard radio hardware. It demonstrates a substantial throughput improvement and provides insight into the sources of that improvement.

The rest of the thesis is organized as follows. The remaining sections in this chapter describe the properties of dense wireless networks and present measurements from a test-bed. Chapter 2 presents an overview of ExOR’s design, followed by algorithm details. Chapter 3 presents an evaluation of ExOR’s performance. Chapter 4 describes related work, and Chapter 5 concludes.

## 1.1 Properties of Multi-Hop Wireless Networks

ExOR takes advantage of two properties that make multi-hop wireless networks different from wired networks: the broadcast nature of the wireless medium, and links with high loss rates. Figure 1-1 shows a simple network for purposes of illustration; suppose that node A wishes to route a packet to node D. Using conventional routing, and assuming the indicated delivery probabilities, the highest throughput route is  $A \rightarrow B \rightarrow C \rightarrow D$ ; this route minimizes the expected number of times that the packet will have to be re-sent to recover from loss. At a high level, ExOR takes advantage of the possibility that A’s transmissions might reach C or even D as well as (or instead of) B. Thus one would expect ExOR to be most effective when

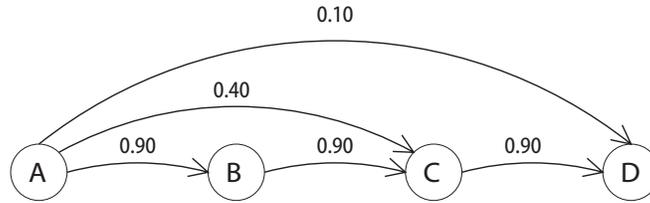


Figure 1-1: Simple network example, with delivery probabilities.

the following are true:

- Every node has non-zero delivery probability to multiple other nodes, so that there is a choice of forwarders. Whether this is true is mostly a function of the density and geometry of the network.
- Transmissions reach potential receivers with a wide range of delivery rates, so that packets occasionally travel a long way towards the ultimate destination. More specifically, ExOR depends on a gradual delivery probability fall-off; if there is a distance at which the probability abruptly drops from high to low, then conventional routing via a node just before that distance is the best strategy. Whether delivery probability decreases slowly depends on the interaction between modulation (and coding) and sources of error such as interference and multi-path fading.

### 1.1.1 Causes of Packet Loss

Wireless transmissions are severely degraded by the effects of *multipath propagation*. A signal emitted by a radio antenna propagates in all directions; multipath arises because the propagated signal reflects off, refracts through, and diffuses around scattering objects in the channel environment. Example obstructions include buildings, trees and cars in outdoor settings, and walls, furniture and people in indoor settings. Scattering and propagation over long distances increasingly attenuates signal power, an effect called *path loss*. Thus, a radio receiver observes multiple attenuated and time-delayed versions of the transmitted signal that are further corrupted by additive receiver thermal noise and other forms of interference. The copies of the transmitted signal might add constructively, thereby increasing the signal-to-noise ratio (SNR), or destructively, thereby decreasing the SNR. With the relative motion of radios and scatterers in the channel environment, SNR fluctuations occur across both time and space and are generally called *fading*.

In addition to channel effects such as path loss and fading, packet losses can be caused by collisions from simultaneous transmissions. Dense wireless networks using a shared channel follow a medium access (MAC) protocol, similar to their wired counterparts, through which nodes negotiate access to the shared resource. The radios used in evaluating ExOR follow a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) MAC protocol in which nodes sample the signal level of the shared channel prior to transmission. If the energy level is above a particular Clear Channel Assessment (CCA) threshold, transmissions are deferred. Occasionally, the MAC protocol fails to properly mediate access, resulting in simultaneous transmissions or *collisions* which result in cause packet losses, because the receiving radio cannot successfully decode the signal. Collisions may occur because the CCA threshold is too high or because of simultaneous transmission. Other losses may occur from sources of interference from nearby devices or networks which disrupt packet transmission.

## 1.2 Measurements

The rest of this chapter presents measurements of a test-bed that show that the assumptions of multiple receiver and intermediate links hold on that test-bed, and argues that they are likely to hold in other networks using similar technology. The reader should be warned, however, that it is easy to imagine modulation techniques, network geometries, and interference sources that cause the assumptions to be false.

The test-bed is an urban multi-hop network intended for Internet access. It consists of 38 nodes in apartment buildings scattered over an area about two kilometers on a side. Each node consists of a PC, an 802.11b radio, and a roof-mounted omni-directional antenna. The network primarily uses a conventional routing protocol, and was not constructed with ExOR in mind. The data presented in this section are derived from experiments in which each node in turn sends 1500-byte broadcast packets as fast as its 802.11 interface will let it for 30 seconds, using the 1-megabit/second transmit rate; each receiver records the fraction of the transmitted packets it receives from each sender. The radio transmit power level is 200 milliwatts. User traffic was disabled throughout the experiment, so it is assumed the majority of packet losses are due to signal attenuation caused by distance or multipath.

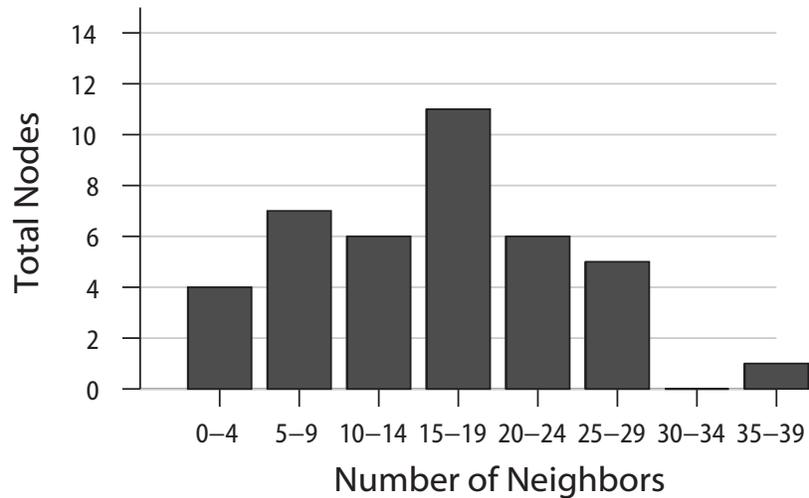


Figure 1-2: Histogram of the number of neighbors for each node where a neighbor decodes at least 20% of a node’s packets.

### 1.2.1 Multiple receivers per packet

Figure 1-2 shows a histogram of the number of neighbors per node, where a neighbor is defined to be a receiver that decodes at least 20% of the sender’s broadcast packets. In this particular test-bed, most nodes have at least on the order of ten neighbors, so that one might expect opportunistic routing to often have multiple receivers to choose from.

### 1.2.2 Gradual delivery ratio fall-off

Figure 1-3 shows the relationship between distance and delivery ratio (fraction of packets delivered), with two data points per pair of nodes in the test bed (one point for each direction). The graph shows only pairs that delivered at least one packet. The triangles denote the various receivers for a particular sender.

Most of the low loss-rate pairs in Figure 1-3 are separated by no more than a few hundred meters. However, there are many longer-distance pairs with reasonably high delivery ratios. Many of these ratios are low enough that conventional routing would probably avoid them in favor of multiple higher-quality links, but high enough that ExOR could profitably make use of the packets that they do deliver.

The data in Figure 1-3 are not what one might predict based on the inverse square law attenuation and the Prism 2.5 chip-set manufacturer’s specifications for the relationship between signal-to-noise ratio (S/N) and bit error rate [1]. The specifications show that there

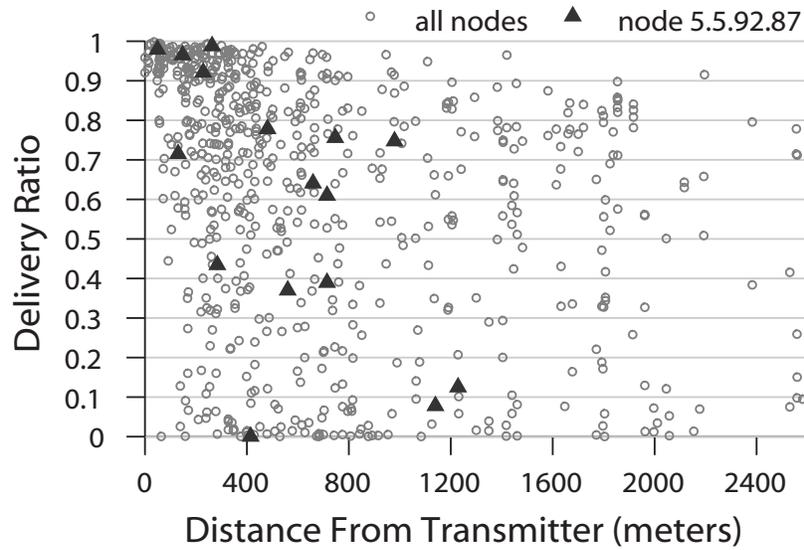


Figure 1-3: Comparison of distance versus delivery ratios. Highlighted points represent a single sender's results.

should be an abrupt transition from 0% packet error rate to 100% error rate; that is, only a very narrow range of S/N values should produce the kind of intermediate loss rates that dominate Figure 1-3. Thus it is not likely that losses on these intermediate links are predominantly caused by attenuation with distance and additive white Gaussian noise; perhaps multipath fading and intermittent interference are to blame. Other measurements of 802.11b and sensor-net test-beds show similar distributions [6, 9].

# Chapter 2

## Design

This chapter uses the observations about the underlying properties of wireless networks from Chapter 1 to develop ExOR as a protocol to take advantage of lossy links in large networks using cooperative diversity techniques. The key challenge in designing ExOR is coordination: in order to exploit the spatial diversity and independent error properties of wireless channels described earlier, many nodes must cooperatively schedule and order their transmissions. This chapter develops the ExOR protocol and explores several practical issues which influence its implementation and performance.

Section 2.1 begins with an overview of the protocol and an example of operation under ideal conditions. In the following sections, each of the three major components of ExOR is described in further detail. Selecting a small but effective forwarding set has the greatest effect on overall performance; Section 2.2 identifies the ideal forwarding set and a heuristic which achieves similar performance with significantly fewer nodes. Forwarding performance and protocol correctness is affected by frequent control packet losses. Section 2.3 describes how ExOR uses batched transmissions to cope with lost control messages. Section 2.4 then explains how nodes schedule transmissions in an effort to avoid collisions. Finally, since ExOR is intended for real-world workloads, Section 2.5 explains techniques for interaction with end-to-end protocols such as TCP.

### 2.1 ExOR Operation

This section provides an overview of ExOR and outlines how the protocol would operate in an example network. Later sections describe the individual components of ExOR in

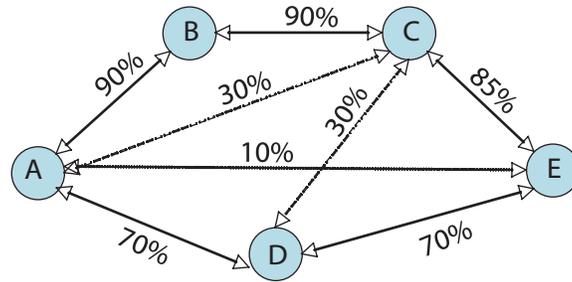


Figure 2-1: Example five node network with packet loss. Link delivery rates are shown along the edges of the graph.

greater detail and explain specific design decisions. For the sake of simplicity, the results in this section are taken from simulation. The simulator models losses using an independent random variable and thus does not account for interference or time-varied fading. Results from a working prototype are presented in Chapter 3.

ExOR operates on batches of packets. The source node includes a list of candidate forwarders in each packet, prioritized by closeness to the destination. Receiving 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 received each packet. Each node maintains an in-memory copy of the batch map, which is updated with information from received batch maps. The remaining forwarders then transmit in order, sending a *batch fragment* including only packets which were not acknowledged in the batch maps of higher priority nodes. The forwarders continue to cycle through the priority list until the destination has enough packets to recover the original data using forward error correction.

ExOR nodes schedule transmissions to avoid collisions. Forwarding nodes initially delay transmissions based on their priority order, waiting for higher priority nodes to forward packets first. As nodes forward, each node updates its local batch map information based on received packet headers and adjusts a transmission timer, which expires when it is the node's turn to forward. Forwarding nodes, including the source node, continue forwarding the packets of a particular batch until 90% of them have been received by higher priority nodes or the destination. Multiple iterations through the forwarding set may be required.

## Batch Preparation

The source begins by preparing a *batch* of packets for transmission. First, the sender prepends an ExOR header shown in Figure 2-2 to each packet. The forwarding set in the header is the subset of nodes which may potentially relay packets to the destination, prioritized using a measure of closeness to the destination. Next, the sender specifies the forwarding set in priority order, with the lowest priority nodes first. In the case of the example in Figure 2-1, the list contains “B, D, C”. The forwarding nodes do not modify the forwarding set specified in the packet header. The source is implicitly part of the forwarding set, and uses the lowest priority member. Finally, the source generates a unique 32-bit *batch identifier* by combining its address with a local counter, sequentially numbers the packets in the batch, and initializes the *batch map*. The batch map describes the highest priority forwarder to have received each packet. Specifically, each packet’s entry in the batch map is the highest priority forwarder’s offset within the forwarding set. Initially only the source has received any packets, so all the entries are filled with zeroes, the index of the source in the forwarding set. Each node maintains an in-memory batch map and embeds a copy in each packet upon transmission, so other nodes can update their local state. Nodes will use the data in the batch map to make forwarding decisions, relaying packets only if higher priority nodes have not received them already. Since packet loss is common, the sender includes the same batch map in all the packets within a particular batch at the time of transmission. Nodes use the repeated batch map as a gossiping mechanism to convey which nodes have successfully received the packets in the batch.

## Fragment Transmission

At transmission time, forwarding nodes send a *batch fragment* containing the subset of the received packets which have not been acknowledged by higher priority nodes. Each packet in a batch fragment includes a *fragment number* and the *fragment size*, the total number of packets in the fragment. This metadata signals the end of a transmission. Nodes use fragment information to schedule medium access. If a transmission is underway, competing nodes defer until the predicted end of a fragment to avoid collisions.

Ver	HdrLen	PacketLen	
Source IP			
Destination IP			
Batch ID			
PktNum	BatchSz	FragNum	FragSz
FwdSetSize		ForwarderNum	
Forwarding Set			
Batch Map			
Checksum			

Figure 2-2: ExOR Packet Header format. ExOR uses IP addresses to specify forwarding nodes. The ultimate source and destination (outside the ExOR network) are specified in the encapsulated IP header, after the ExOR header. As a space-saving measure, the batch map is a packed array of offsets within the forwarding set.

### Packet Reception

All nodes examine the header of every successfully decoded packet. If the node is listed in the forwarder set, the packet is added to a buffer associated with the batch id, the in-memory batch map is updated with the sender's batch map contents, and an estimate of when the fragment will end is updated using the fragment information. If the received packet is from the next highest priority forwarder in the list, the local node sets the estimated forwarding time timer for immediately after the predicted end of the fragment. Otherwise, the forwarding timer is set to the minimum forwarding time (five packet durations in the implementation) times the number of higher priority nodes in the list. This estimate is intentionally aggressive; the estimate is revised if any other packets belonging to the batch are heard, but if they are not, then the medium is assumed to be clear.

### Deciding whether to forward and completion

When the forwarding timer expires, the node inspects its in-memory batch map, sending packets which have not been acknowledged by higher priority nodes. The node forwards only if less than 90% of the batch map has been acknowledged by a higher priority receiver.

If it is the destination's turn to forward packets, it sends a small number of packets (ten in

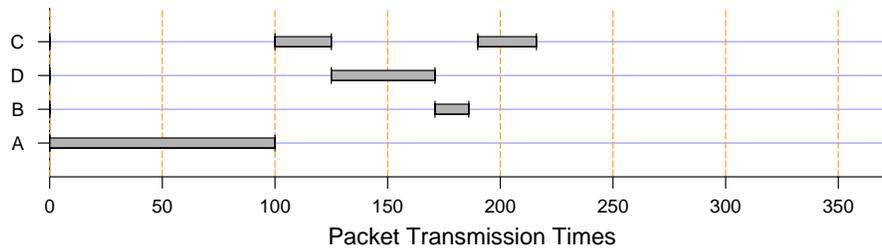


Figure 2-3: ExOR Transmission timeline for 100 packets sent from node A to node E in the example network depicted in Figure 2-1. ExOR delivers the packets in 216 transmissions and uses all the available nodes to forward.

the implementation) with complete ExOR headers but with no payload. The batch maps in these packets inform the other nodes of which packets the destination has received without consuming significant medium time.

If the batch map indicates that over 90% of the packets in a batch have been received by higher priority nodes, the batch transmission is finished from the local node’s perspective. The buffered packets are flushed, the batch map is cleared and newly received packets with the same source address and batch are ignored. If forwarders kept sending, ExOR would continue attempting to deliver the last few packets using small fragments. Small transmission fragments cause collisions, because it is difficult for nodes to estimate the end of fragment transmission based on a few packets.

### Example

Figure 2-3 shows the ExOR transmission timeline for 100 packets being sent from node A to node E in the example network from Figure 2-1. Simulated packet loss, based on the delivery rates shown in Figure 2-3, determined the number of received packets. Node A begins by sending a single batch of 100 packets encoded with the forwarder set “C, D, B”. After transmission, the destination sends null acknowledgment packets which informs the other nodes it has successfully received ten packets (not shown in the figure). Node C, which received 29 packets from A, then sends 25 packets, omitting packets which have already been acknowledged by E. Once C’s transmissions are complete, D similarly selects 45 packets of the 70 received from A (and the additional 7 it received from C) and broadcasts them. Next, B does the same and sends 15 packets. Node A observes that four packets have not yet

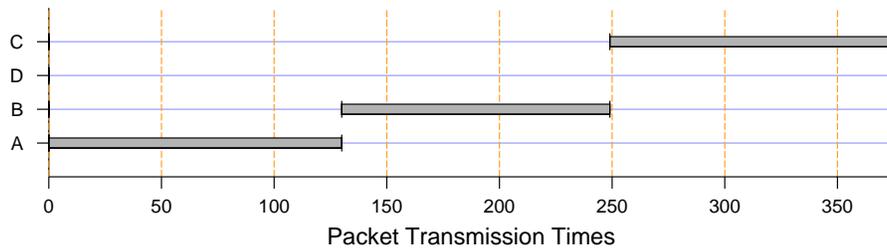


Figure 2-4: Unicast transmission timeline for 100 packets sent along the best route from node A to node E in the example network from Figure 2-1. A total of 339 transmissions are required to deliver packets to node E.

been received by higher priority nodes, but since four is less than 90% of the batch size, A suppresses its transmissions and decides it has completed its packet forwarding duties.

At this point, after a single iteration through the forwarding set, the destination has received 73 of the 100 packets. Node E begins a second round of forwarding by transmitting null packets with only batch maps after waiting for the minimum transmission time, since A did not forward any packets. Even if Node E's transmissions were not received by other nodes, a second round of transmissions would begin, since less than 90% of the packets had been received at the destination. Node D then transmits the 26 packets it has overheard which were not received by E. The remaining nodes do not transmit, as over 90% of the packets have been received. Finally, node E transmits another set of null packets after two minimum transmission times, informing all nodes that it has successfully received enough packets. In total, 216 packet transmissions were required to route the packets from A to E.

For comparison, Figure 2-4 shows a timeline for the best unicast route between A and E. In unicast routing, the link-layer uses per-packet acknowledgments to ensure each packet is successfully received by the next hop. As a result, each packet transmission takes  $1/r^2$  transmissions on average, where  $r$  is the delivery probability of the individual link. The simulation does not include acknowledgment packets in the transmission count, but does account for ACK losses.

The best route from A to E is via nodes B and C. The initial 100 packets sent by node A require 113 transmissions to reach B. Delivery from B to C requires an additional 119 transmissions, followed by 125 transmission to go from C to E. An alternative route via D would have needed roughly 205 transmissions on each hop or 410 in total.

## 2.2 Forwarder Set Ordering and Selection

This section explains how ExOR selects nodes to participate in packet forwarding. Ideally, ExOR would maximize cooperative diversity by potentially using all the nodes in the network to forward packets. Unfortunately, overall performance would likely suffer because many of the nodes would not have any packets to forward, causing unnecessary delays caused by estimation mistakes. For small networks, the best forwarding list may include all the nodes, but for larger networks, only the forwarders most likely to relay packets should be used. ExOR must also determine a priority ordering for the forwarding nodes. Simple heuristics which do not account for performance may yield unacceptably poor throughput.

### 2.2.1 Priority Ordering

The forwarding set is specified in priority order based on the cost of transmission to the destination. Ordering increases performance by decreasing the number of transmissions: nodes close to the destination are more likely to reach the destination with a fewer number of transmissions.

ExOR uses a modified version of the ETX metric [5] to prioritize the forwarder set. Because ExOR transmissions do not follow a specific path, the ETX metric over multiple hops only serves as a rough, conservative estimate of the total end-to-end transmission count. Nodes which are farther than a single hop away from the destination add ETX values along potential routes to the destination and use the minimum path metric to estimate closeness.

Nodes calculate ETX using measured packet delivery probabilities. The delivery ratio  $d$  is measured using dedicated link probe packets. Each node broadcasts link probes of a fixed size, at an average period of  $\tau$  (ten seconds in the implementation). To avoid accidental synchronization,  $\tau$  is jittered by up to 10% per probe. ETX probes are broadcast, so they are not acknowledged or retransmitted. Every node remembers the probes it receives during the last  $w$  seconds (100 seconds in the implementation), allowing it to calculate the delivery ratio from the sender at any time  $t$  as:

$$d(t) = \frac{\text{count}(t - w, t)}{w/\tau}$$

$\text{Count}(t - w, t)$  is the number of probes received during the window  $w$ , and  $w/\tau$  is the

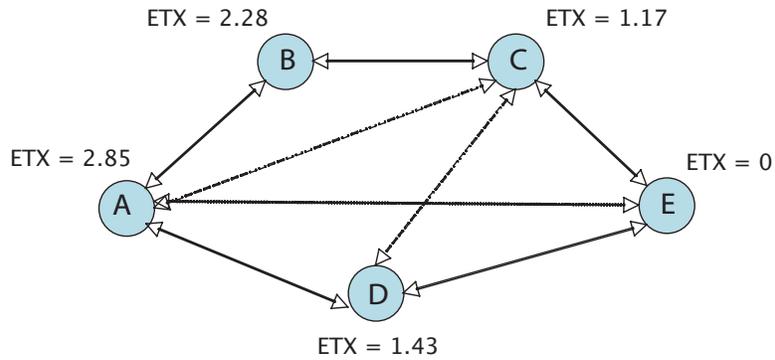


Figure 2-5: Estimated transmission count (ETX) to node E for each node in the sample network from Figure 2-1.

number of probes that should have been received. Each node then maintains a table with the ETX to all the other nodes in the network based on its knowledge of the network link-graph.

Figure 2-5 shows the ETX values to the destination node E in the network shown in Figure 2-1. The ETX value for node B is the sum of C’s ETX and the ETX of the link to C. Similarly, the ETX for node A is dependent on node D’s ETX, since it is lower than its direct ETX to the destination.

### 2.2.2 Forwarder Set Culling

As mentioned earlier, large forwarder sets result in nodes with no packets to forward, degrading overall throughput. This becomes particularly relevant in large, dense networks where many of the nodes do not find an opportunity to contribute. This problem is illustrated in Figure 2-6 which depicts the ExOR transmission timeline for transmission between two distant nodes in a Manhattan topology. Nodes are evenly laid out in an infinitely sized grid and delivery probabilities are proportional to  $1/d^2$  where  $d$  is the distance.

Because of the high forwarder density, many of the nodes make fewer than five transmissions because other nodes have already forwarded the packets. In simulation this has little negative effect, but in practice packet loss would cause multiple nodes to attempt to forward at the same time, as the nodes would not defer transmissions based on received packet headers. Even worse, extended periods of time would be spent waiting for transmissions if multiple forwarding iterations were required, as many nodes would have no packets to relay on the later rounds. Even if these forwarders sent a null packet indicating they had nothing

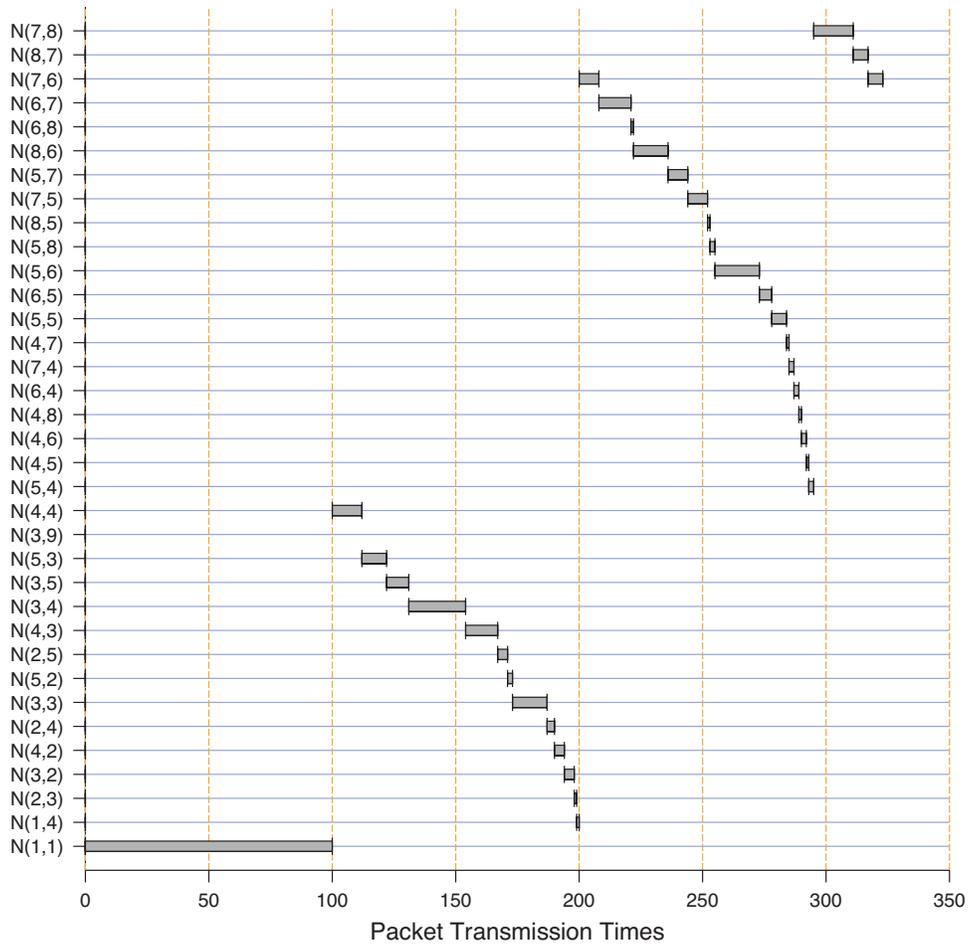


Figure 2-6: ExOR transmission timeline for routing 100 packets between two distant nodes in a dense, uniform network using all available nodes. The network addresses are sorted in priority order by ETX to the destination and nodes which did not transmit any packets have been omitted. A total of 323 transmissions are needed to deliver packets to the destination.

to send, channel losses might cause other nodes to not receive it.

Any small forwarding set is likely to avoid forwarders with no packets to send, but may not include the best forwarding nodes. ExOR uses only those forwarding nodes which are likely to have the greatest positive impact on performance. The source runs a rough simulation of delivery using a complete prioritized forwarder set and selects the nodes which transmit at least 10% of the total transmissions in a batch. This approach filters out the nodes which only forward a handful of packets and are therefore most likely to decrease throughput.

Figure 2-7 demonstrates the effect of forwarder culling on the transmission timeline. The

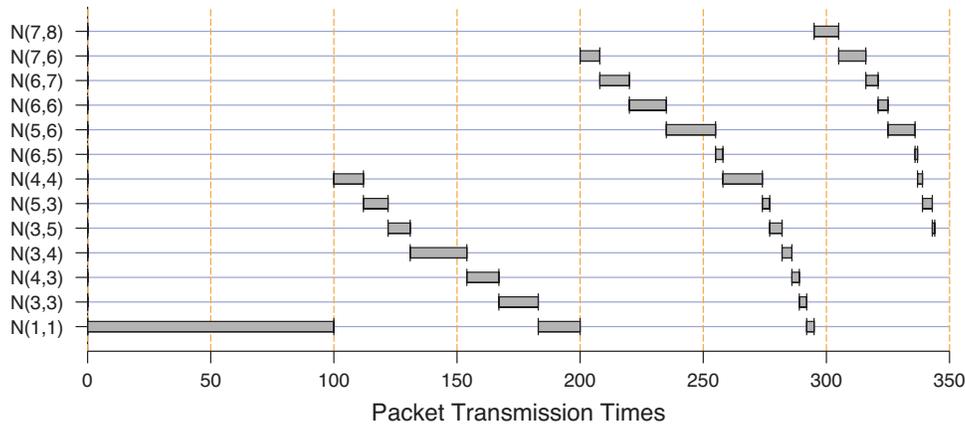


Figure 2-7: ExOR transmission timeline for routing 100 packets between two distant nodes in a dense, uniform network using forwarder culling. A total of 344 transmissions are required to route at least 90% of the packets, despite the large reduction in forwarder set size.

smaller forwarder set needs more transmissions to successfully forward packets to the destination, but only 344 compared to 323 for the full forwarder set demonstrated in Figure 2-6. However, the culled set only uses thirteen forwarding nodes, which is significantly fewer than the 81 used in the full set (nodes which did not transmit any packets were omitted from the initial figure). Using fewer nodes is likely to generate higher throughput, since there will be fewer delays caused by nodes with no packets to send.

## 2.3 Batch Maps

The batch map serves two specific purposes in ExOR : to eliminate the need for the per-packet acknowledgments used in unicast routing, and to enable nodes to schedule transmissions. The batch map acts as a gossiping mechanism, indirectly informing forwarders of which packets were successfully received. This section describes how ExOR batches transmissions.

### 2.3.1 Batch Map

Instead of acknowledging each packet individually, ExOR sends batches of acknowledgments using a *batch map* embedded in packet headers. The batch map indicates, for each packet in the batch, the highest priority forwarder known to have a copy of the packet. Because

the batch map contains information about all the packets in a batch, only a single packet must be successfully decoded to determine which of the packets were successfully received, according to the sender. Each element in the batch map is encoded as the *forwarder id*, the position of the node in the forwarder set, which is specified elsewhere in the packet header. This optimization makes it economical to embed a complete batch map in every transmitted packet. For a 100 packet batch with 30 forwarding nodes, the total space required is  $\log_2(30) * 100$  bits, or 63 bytes.

### Updating the Batch Map

Nodes listed in the forwarding set of a batch maintain an in-memory batch map which is updated upon packet reception. As packets arrive, each node compares the contents of the received map embedded in the packet header with the local map. Each element in the map is updated with the higher of the received and local values, which represents the highest priority forwarder known to have a copy of the packet referred to by the map entry. If the receiving node is the highest priority forwarder thus far, the packet's entry is updated with its forwarder id. Nodes embed a copy of their in-memory map in the packet header when forwarding packets. Each node frees the local batch map, along with buffered packets, when more than 90% of the map elements indicate that higher priority nodes have successfully received the packets.

### Example

Figure 2-8 graphically describes a batch map update, highlighting indirect updates due to gossiping. The received map shown on the right is received in a packet from forwarder three, and differs in 13 locations which are highlighted in shades of gray. For ten of the entries, which are shown in dark gray, the map is updated to show forwarder four as the highest priority forwarder responsible for the packet, even though information came via forwarder three. For the other three entries, shown in light gray, the map is updated to list forwarder three. This gossiping mechanism helps ExOR recover from batch map losses.

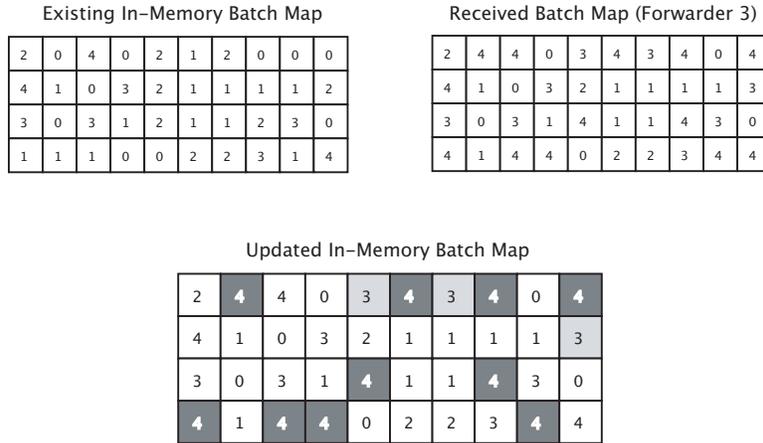


Figure 2-8: Example of a batch map update. Each entry in the batch map represents a packet. The number is the highest priority forwarder known to have a copy of the packet. In this example, the received batch map from node three contains updates about node three's packets and node four's packets. The direct updates from three are shown in light gray, the indirect updates from four are highlighted in dark gray.

## 2.4 Collision Avoidance

ExOR nodes schedule transmissions in an effort to minimize collisions. Low-priority forwarders wait for the fragment transmissions of higher-priority nodes to complete before sending. Because the last packet in a fragment may be lost, each node estimates the fragment finish time, so it can begin forwarding quickly. The estimate must be robust enough to deal with lost packets, deferred packets due to channel contention and sender-side delays.

ExOR uses an exponentially weighted moving average (EWMA) filter to estimate the end of a fragment transmission based on received packets. The estimator takes the packet sequence number within the fragment and the received time and predicts the end of the fragment based on historical measurements. EWMA computes a linear combination of infinite history, weighted exponentially. The implementation chooses a smoothing value  $\alpha$  of 0.9, which allows the estimator to identify the average pace of transmissions based on a handful of packets, but filters out errors caused by transient delays.

Figure 2-9 illustrates the accuracy of the Moving Average and EWMA estimators when compared with a packet trace with competing traffic present. The Moving Average estimator used a window size of  $n = 12$  and the EWMA estimator uses an  $\alpha = 0.9$ . Both parameters produced the smallest error on the trace data. From the trace, it is clear EWMA is better at predicting transmission times than moving average.

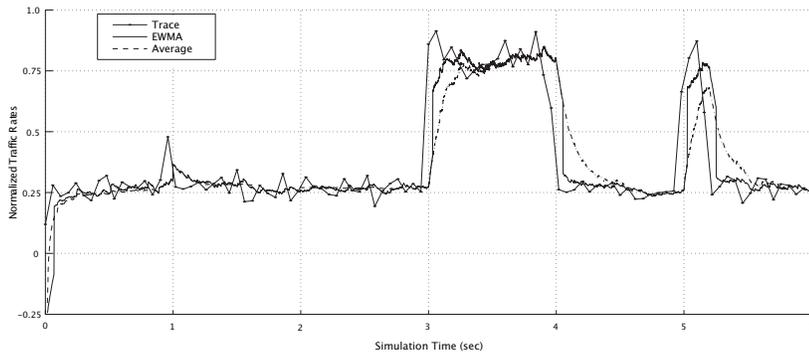


Figure 2-9: Simulated performance of Exponentially Weighted Moving Average (EWMA) and Moving Average estimator on packet trace data. The EWMA with ( $\alpha = 0.9$ ) proves to be more agile than the Moving Average filter ( $n = 12$ ).

## 2.5 Practical Considerations

ExOR loses efficiency when the destination has received all but a handful of the packets. No one forwarder is likely to have more than one of the required packets, so batching does not act to reduce overhead. Worse, the nodes that hold the last few packets may know that they have been forwarded by a higher priority node, and may not be aware that those transmissions were lost. For these reasons the receiver should use a separate mechanism for the last few packets: either asking the source to re-send them with ordinary unicast routing, or expecting the batch to include erasure-coded redundant data so that the last few packets are not needed. The ExOR implementation uses the latter technique.

If ExOR is used to carry TCP traffic, ExOR's batches are likely to interact badly with TCP's window mechanism. If the loss rate is not very low, TCP may use a window size too small to allow ExOR to accumulate the 10 more more packets required for an efficient batch. For this reason ExOR may best be combined with a web proxy running at the edge of the wireless network, allowing the proxy to accumulate an entire file to present to ExOR. This would have to be coupled with a retransmission system on top of ExOR, since ExOR does not guarantee reliable delivery.

# Chapter 3

## Results

This chapter presents experimental results which show ExOR is able to deliver bulk data faster than the best unicast route, 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.

Following a description of the experimental test-bed and methodology, Section 3.2 begins with a comparison of end-to-end throughput for ExOR and traditional unicast routing. The results are then examined separately for short and long routes, with an emphasis on understanding the types of routes which benefit the most from ExOR. Section 3.3 considers a representative route, detailing ExOR's operation and verifying some of the protocol assumptions. Section 3.4 examines if ExOR produces consistent results. Finally, Sections 3.5 and 3.6 study the effects of batch size and correlated losses.

### 3.1 Evaluation Methodology

This section describes the framework used to benchmark ExOR and unicast routing performance. Experiments measuring end-to-end file transfer throughput were run on a 38-node outdoor 802.11b network over three days. Testing of all the possible node pairs was not feasible, so a randomly selected set of 65 node pairs were considered to provide a full range of results. The experiments were repeated multiple times, to reduce the impact of background interference caused by user traffic on the network.

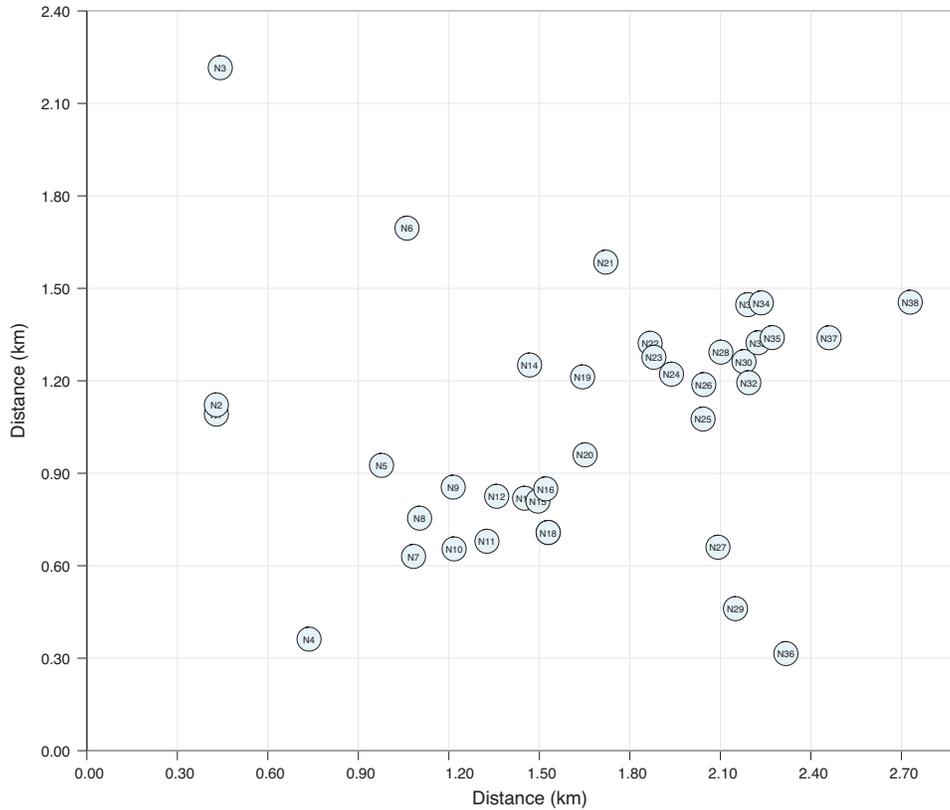


Figure 3-1: Physical layout of the 38 Roofnet nodes which participated in the performance evaluation.

### Network Description

All the results in this chapter were generated on Roofnet [2], an outdoor roof-top wireless network based on 802.11b radios. Roofnet consists of 38 nodes distributed 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 Figure 3-1. 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 identical 802.11b cards based on the Intersil Prism 2.5 chip-set. The cards transmit at 2.422 GHz (802.11b channel 3) with the transmission power level set to +23 dBm (200mW). The other four nodes use radios based on the Atheros AR5212

chip-set, with a transmission power level of +20 dBm (100mW) on the same 2.422GHz channel. The omni-directional antennas provide 8dBi of gain with a 20-degree -3dB vertical beam-width. Cabling and lightning arrestors introduce an attenuation of 6 to 10 dB depending on the length of the cable.

The network provides Internet service to the volunteers hosting the nodes. User traffic is routed to three gateway nodes located in the lower right corner of network via a unicast routing protocol similar to DSR [12]. The radios operate in the Prism 2.5 “pseudo-IBSS” mode, a simplified version of the 802.11 IBSS (ad-hoc) mode which does not use beacons. The nodes generate link measurement probes every ten seconds which are used by the existing routing protocol and ExOR to measure link quality. An in-depth study of the network’s link-level characteristics can be found in [2], which details inter-node loss rates in the absence of traffic. The radios are capable of transmitting packets at 1, 2, 5.5 or 11Mbit/s rates and send unicast acknowledgments at 1Mbit/s.

## Measurements

Each experiment measured routing performance over 65 randomly selected node pairs and consisted of nine iterations. First, the nodes broadcast 1500 byte packets every ten seconds for ten minutes and reported the measured delivery probabilities to all the other nodes to a central server. These measurements provided the complete link graph of the network, used to compute ETX metrics and routes. The server then distributed the link graph to all the nodes. Next, the server contacted each node pair in sequence, telling them to measure the time required to transfer a 1.0 megabyte file using ExOR and then unicast, padded with 15 second wait intervals. Since ExOR is expected to only deliver 90% of the packets, an additional 110KBytes of redundant data was added to the file when transferred using ExOR. The ExOR batch size was fixed at 100 packets. The exception are the experiments in Section 3.5, in which considers multiple batch sizes ranging from 10 to 250 packets. Each packet contained 1024 byte of payload data and either a unicast or ExOR header. Unicast routing headers varied between 24 and 48 bytes, depending on the number of hops. ExOR headers varied between 44 and 114 bytes, depending on the forwarding set size. In both cases, throughput was measured using the total time required to transfer 1.0 megabyte of data (1.1 megabytes for ExOR). Every twenty minutes, the central server suspended the experimental runs to recollect the link graph measurements, to ensure the best unicast routes and forwarding sets

were being chosen. During the experiment, existing Roofnet routing and user traffic were present.

To avoid measurement biases due to interference from Roofnet user traffic, the results are based on the median of the nine measurements for a given node pair for both ExOR and unicast. The exception is Section 3.4, which studies the variations between experimental iterations.

The best unicast route was chosen using the ETX metric. It is logistically difficult to test all the available routes to verify that ETX chose the best route. However, ETX has been shown to determine the best routes [5, 7] when the link measurements are accurate.

All traffic in these experiments (ExOR and unicast) is sent at 1Mbit/s. Unicast traffic is sent in batches on each hop, so that only one node sends at a time. This was done to make a fair comparison of the two protocols, since multi-hop unicast traffic tends to suffer losses due to collisions between successive hops. Batching improved unicast throughput in all cases.

In addition to throughput measurements, the nodes collected the received headers and arrival times of all packets of a single iteration of the experiment. These traces were centrally processed to reconstruct the state of the wireless channel, providing the trace data for the case study in Section 3.3.

## 3.2 End-to-End Performance

Figure 3-2(a) compares the throughput CDFs of ExOR and the best unicast route for all 65 randomly chosen node pairs. ExOR's throughput is 33 KBytes/sec for the median pair, whereas the best unicast route achieved 11 KBytes/sec for the median pair. The following sections explore why ExOR outperforms the best conventional unicast route.

There are three regions in Figure 3-2, which correspond to route length. The top 10% of the CDF corresponds to node pairs which have a direct link with less than a 50% loss rate. In these cases, ExOR outperforms traditional unicast routing because it eliminates per-packet acknowledgments, using repeated transmissions and gossip to ensure reception of batch maps. In these cases, ExOR improves overall performance by an average of 26%. The next 20% of of pairs corresponds to two hop routes. In these cases, ExOR improves performance by between 50% and 100%. Finally, the remainder of the CDF compares routes

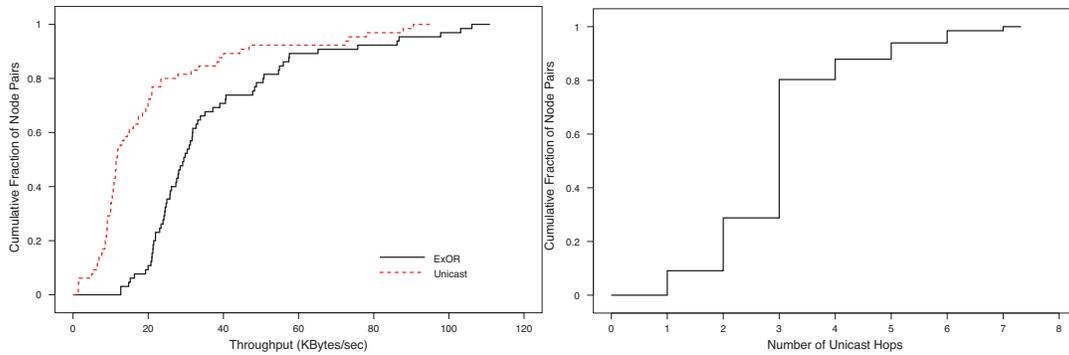


Figure 3-2: Overall end-to-end performance of ExOR and unicast routing over the 65 randomly selected node pairs. The plot on the left shows the median throughput for each of the 1.0MByte transfers. ExOR provides three times as much throughput as unicast for the median pair. The plot on the right shows the number of hops taken by the unicast route.

with three or more hops. In these cases, ExOR generally improves performance by a factor of two.

### The 25 Highest Throughput Pairs

ExOR's relative performance improvement varies in number of hops in the unicast route. To better understand why ExOR improves performance for particular node pairs, it is helpful to consider them separately based on the number of hops in the unicast route. Figure 3-3 compares the 25 highest throughput pairs, organized as bar graphs, which allows for closer examination of individual node pairs. The top five pairs and the pair N13-N7, near the center of the figure, correspond to single hop unicast routes. Asymmetric links produce unnecessary retransmissions in conventional unicast routing because of lost ACK frames, even on single hop routes. In such cases, ExOR outperforms unicast routing because it recovers from lost acknowledgments using the redundant batch map information. On single hop routes, the destination node sends packets containing only headers to inform the source of the batch status.

The next nine pairs to the left are routes with two unicast hops. Some of the pairs do not see a large improvement due to ExOR, largely because there are often a small number of forwarding node choices for the two hop routes. For these pairs, the performance gains come from the batch map's ability to avoid needless resends if packets are lost. However, for more than half of the pairs, ExOR outperforms unicast routing by 50% or more. In these

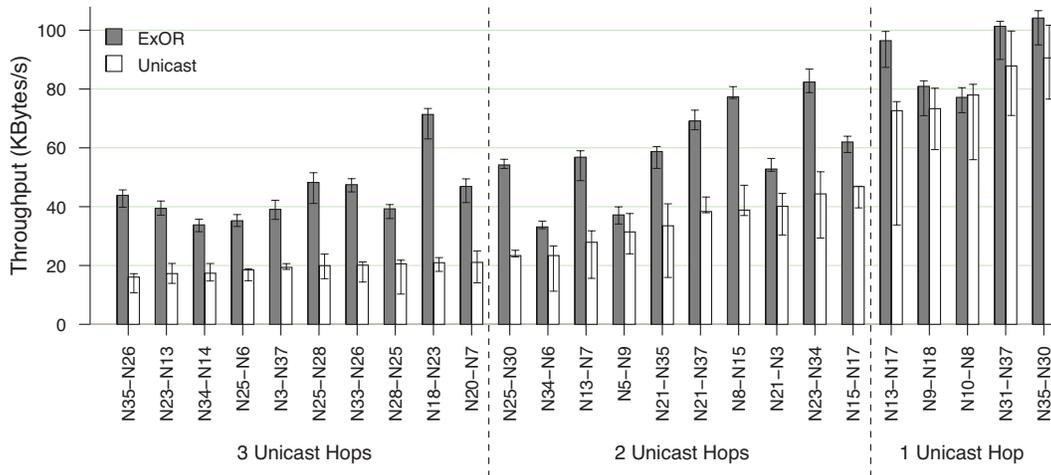


Figure 3-3: The 25 highest throughput pairs, when sorted by unicast throughput. The bars show each pair’s median throughput, and the error bars show the lowest and highest of the nine experiments. ExOR’s improvements in throughput are smaller for shorter routes.

cases, ExOR was able to find additional forwarding nodes, or perhaps use a direct link to the destination which was avoided by ETX due to asymmetry.

The remaining pairs correspond to three hop routes. On most of the pairs, ExOR outperforms unicast 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 avoid retransmission.

### The 25 Lowest Throughput Pairs

Figure 3-4 compares the throughputs of the 25 node pairs at the bottom of the unicast throughput distribution. These node pairs are separated by three unicast hops or more, so are likely to benefit the most from ExOR. The graph, which is sorted by unicast throughput, does not cluster the pairs by number of hops. Most of the pairs which see a factor of two or more improvement in throughput have four or five hops.

The most dramatic performance gains occur over the longest routes, which have between five and seven hops. In these cases, ETX tends to choose long routes because of unicast acknowledgment losses. ExOR’s performance is mostly affected by forward delivery rates, which allows it to find much shorter paths to the destination through a large set of forwarding nodes. The batch maps allow information to flow backwards if the sender can’t hear the next forwarder, which makes it possible to use links with low delivery rates.

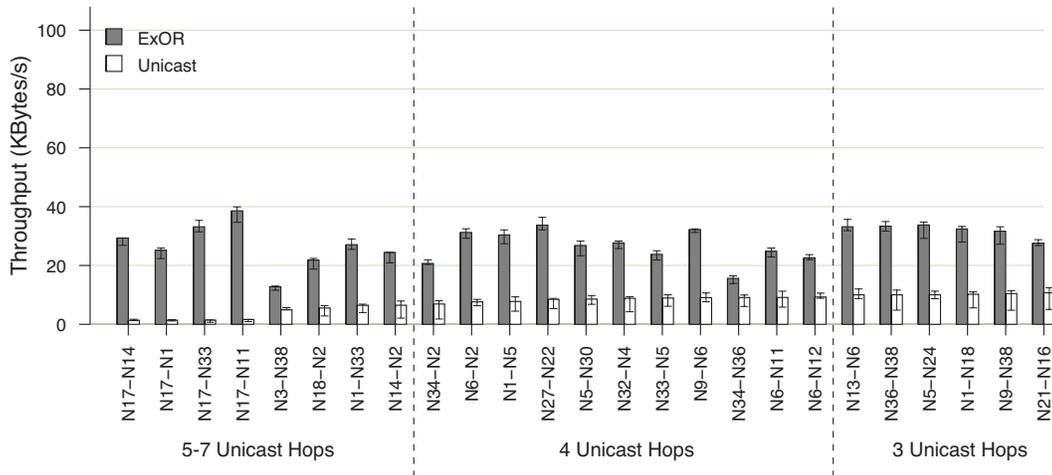


Figure 3-4: The 25 lowest unicast throughput pairs. The bars show each pair’s median throughput, and the error bars show the lowest and the highest of the nine experiments. ExOR tends to outperform unicast routing by a factor of two or more.

### 3.3 ExOR Node Pair Case Study

To better understand why ExOR works well, it is helpful to closely examine the performance of an individual node pair. Figure 3-5 shows the transmission timeline for the first six seconds of an ExOR transfer between the node pair N5-N24, which is near the middle of the overall throughput distribution.

The figure was generated by aggregating traces of overheard packets headers in promiscuous mode from the forwarding nodes. Because of delays within the radio hardware and operating system of the individual nodes, the timing resolution of the trace is on the order of tens of milliseconds, or roughly a single packet time. The transmission timeline provides a rough snapshot of the wireless channel between the forwarding nodes for the six seconds shown, demonstrating the order and timing of each transmission. The plots show a bar if more than four packets were sent by a node within a window of 60ms, the horizontal length indicates the total duration of successive unique packets. The shades of gray represent different batches, and the Background line at the bottom monitors non-ExOR packets which were received during the timeline window.

In the experiment, the coding expands the 1.0 megabyte file to 1.1 megabytes, before it is broken into 11 batches, each containing 100 1KByte packets. The first batch of 100 packets are broadcast as a single fragment by the source node, N5, with the culled forwarder list

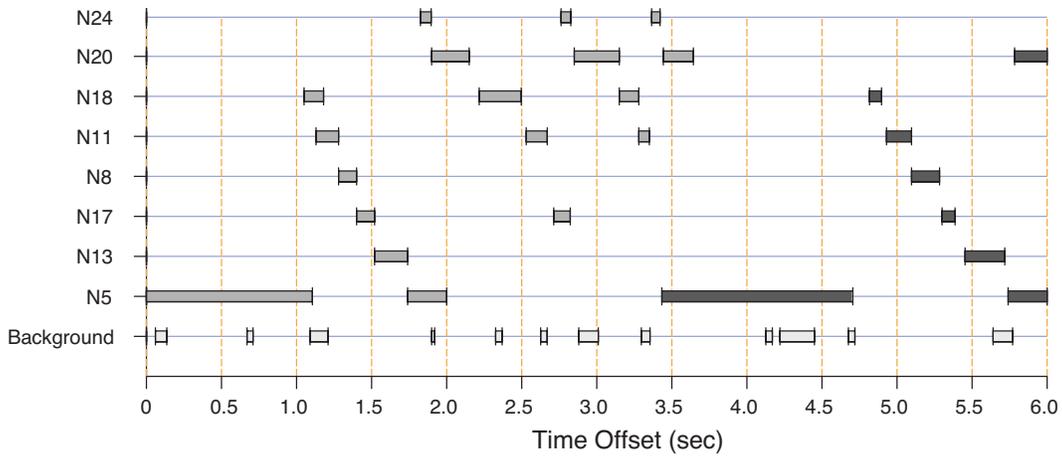


Figure 3-5: Transmission timeline for an ExOR transfer from N5 to N24. The nodes are listed in priority order, based on ETX, to the destination. The light gray bars show all the transmissions from the first batch. The darker gray bars show part of the second batch. Non-ExOR packets which were recorded during the trace are shown in the Background line.

generated from the link graph. In this case, the specified list in priority order is: “N20, N18, N11, N8, N17, N13”.

After the initial 100 packet transmission by the source, the highest priority forwarder which received any packets, N18, sends all the packets it received. N18 and the source transmissions overlapped for 100ms, most likely because of an estimation error. The forwarder, which received roughly ten percent of the packets, did not anticipate the source would continue transmitting past the 1.0 second offset. The source may have delayed transmissions at the MAC because of non-ExOR background traffic.

Next, the remaining forwarding nodes transmit packet fragments in succession. Forwarding nodes typically send fewer packets than they receive, using the information gossiped in the batch maps to determine if higher priority nodes successfully received each packet. The destination node, N24, sends a small fragment containing only batch maps at 1.8 seconds, because it did not hear any transmissions after N13’s fragment. After the destination’s acknowledgments, the highest priority node begins the second batch transmission, immediately followed by the next two priority forwarders. The node N8 does not transmit, because over 90% of the packets in the batch have been acknowledged by higher priority nodes. However, the node N17 does send a small fragment, possibly because it did not overhear some of the earlier batch maps indicating which packets have been acknowledged by the destination.

At 2.7 seconds, during N17’s transmissions, the destination node sends acknowledgments

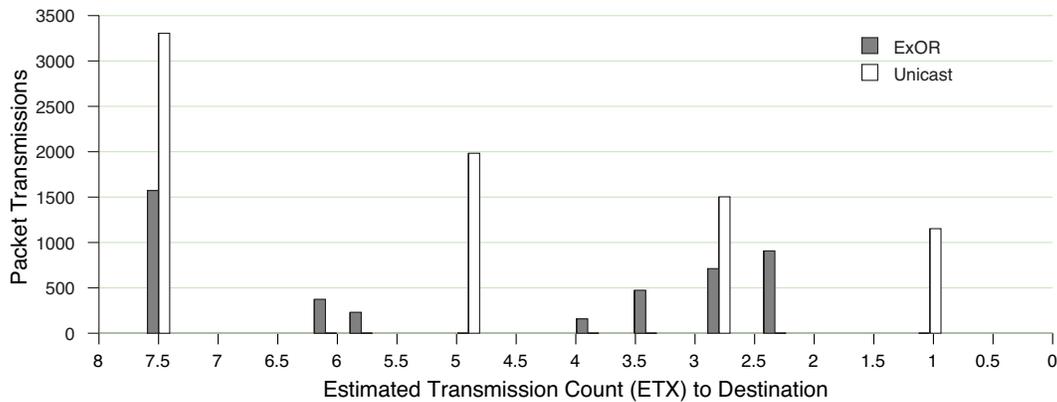


Figure 3-6: Aggregate view of all ExOR and Unicast packet transmissions needed to transfer 1.0MByte of data between N5 and N24. The plot shows the number of transmissions made by each forwarding node, shown in terms of the node's ETX metric to the destination. Packets tend to travel farther using ExOR, since the majority of the source's transmissions were relayed by nodes closer to the destination when compared with unicast.

for the second time, because it did not receive any packets within the timeout window after N11's transmission. Forwarding continues for the three highest priority forwarders, which each send fragments in turn until 90% of the packets in the batch are acknowledged. During the last round, the source node begins transmission of a second batch at 3.4 seconds, since the batch map indicates 90% of the batch has been acknowledged by higher priority nodes and it did not overhear any other ExOR transmissions in progress.

After an additional nine batches, which are not shown in the figure, the data is delivered to the destination after 36.6 seconds, producing an ExOR throughput of 30KB/s.

The best unicast route forwards data over four hops. The source transmits the one megabyte of data for 26.4 seconds to N9. Once all the packets have been acknowledged, N9 transmits for 15.8 seconds to N18, which then transmits for 12.0 seconds. Finally, N23 completes the last hop in 9.2 seconds. The unicast throughput was 15.8KB/s. The unicast route metric was accurate within 4%, since each packet took an average of 7.8 transmissions to reach the destination, implying the route metrics were accurate.

An alternative view of the trace in Figure 3-6 shows ExOR packets tend to travel farther compared to unicast. The majority of the source's transmissions were relayed by nodes closer to the destination than the first unicast forwarder.

Node Pair	ExOR (KB/s)	ExOR Range	Unicast (KB/s)	Unicast 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 3.1: Comparison of Unicast and ExOR throughputs and variation on a 20 node pair subset

### 3.4 Throughput Variation

An unanticipated side-effect of using additional forwarding nodes is a reduction in variation between per-transfer throughput. Table 3.1 demonstrates that ExOR throughput shows less variation between trials when compared with unicast. It is surprising that any variations in overall throughput would exist when transferring thousands of packets, as the throughput should converge towards the average over many packets. However, wireless link delivery rates tend to change over time, due to slow fading on the scale of tens of seconds or greater [2]. In a dense network, unicast routing could change paths in response to fading, but in the test-bed there are often few routes to choose from. ExOR is not as severely affected by individual link fluctuations as long as slow fading is uncorrelated between links. When individual link delivery rates go down, ExOR makes heavier use of other available links within the forwarding set.

Table 3.1 shows the amount of variation between the nine experimental iterations for 20 randomly selected node pairs of the 65 total. The throughput columns are the median of the

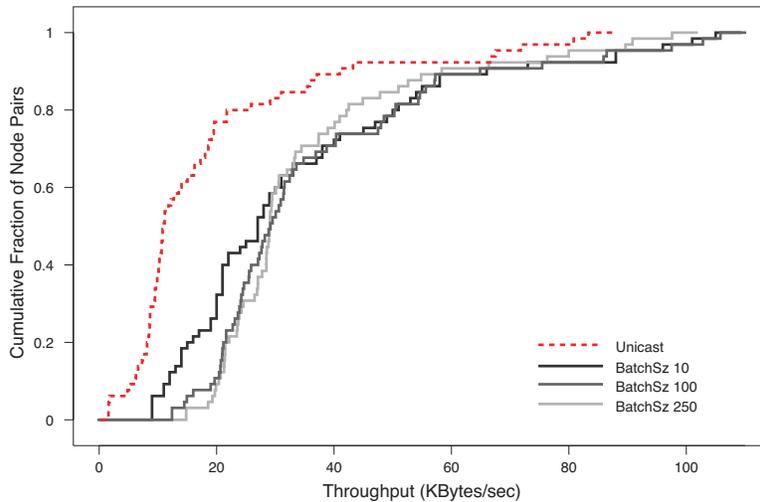


Figure 3-7: CDF of end-to-end ExOR throughput for three batch sizes. All of the sizes outperform unicast routing, but ten-packet batches perform half as well as 100 or 250 packet batches for some pairs.

nine runs and the range column is the difference between the highest and lowest performing run, expressed as a percentage of the median. In general, unicast route performance tends to vary by eight to ten times as much as ExOR.

In unicast routing, the throughput is dependent on the individual performance of each of the hops along a route. ExOR is able to consider multiple links simultaneously, making it less prone to individual fluctuations.

### 3.5 Batch Size

ExOR operates on batches of packets, allowing the use of batch maps to gossip packet reception information between forwarding nodes. As the batch size increases, the likelihood of successfully receiving the batch map also increases, as it is embedded in each packet. However, many transfers may only have a few packets. This section examines how ExOR throughput varies in with batch size, comparing 10, 100 and 250 packet batch sizes on the 65 node pairs studied earlier in the chapter.

The batch size affects the per-packet overhead due to the batch map embedded in the ExOR header. Given a relatively large forwarding set 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 throughput is calculated using the total transfer time, including packet headers.

Figure 3-7 shows the throughput distribution for each of the three batch sizes, calculated based on total transfer time. The results are taken from a single experimental run which began in the afternoon of December 15, 2004. The evaluation methodology is similar to the experiments presented earlier in the chapter, except instead of benchmarking unicast traffic, an additional forms of ExOR were run with a batch size of 10 and 250. Also, only three iterations were done. The results for each node pair are based on the median throughput.

For node pairs with direct communication, the overhead dominates the performance results. As expected, the 250 packet batch maps result in roughly 15% slower performance when compared to 10 or 100 packet batches. On the longer routes, however, the batch size of ten performs 20% worse than its larger counterparts. Because there are fewer packets in each batch, and thus each transmission fragment, it is less likely that the batch map is received by all the forwarding nodes. Without the batch map repetition provided by each packet in a fragment, forwarding nodes are unable to determine which packets have been successfully received.

### 3.6 Independent Loss Simulation

ExOR's design assumes that the majority of packet loss is uncorrelated among receivers. If all losses were due to attenuation or multipath 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-graph as an input and models packet losses using a random variable for each link. When a packet is broadcast, a weighted coin is flipped for each of the links connected to the transmitting node. In the case of independent losses, each of the links has its own random variable. To model losses due to shared interference however, all the links are conditioned on the same random variable. For example, given two links with a 50% and a 75% loss rate,

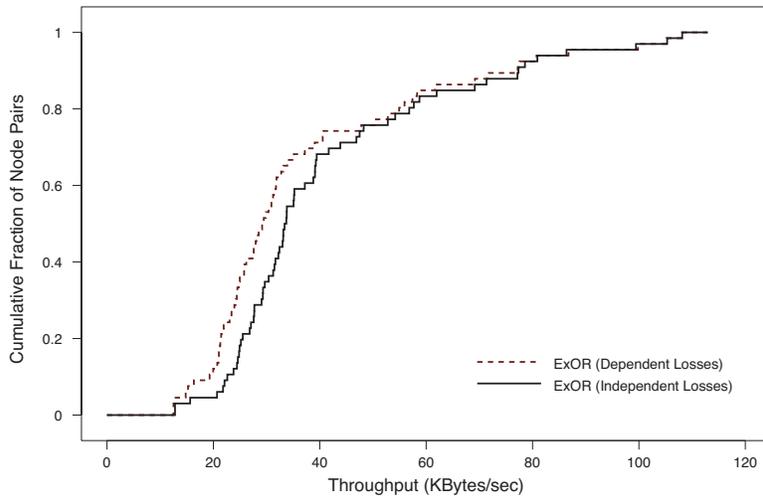


Figure 3-8: CDF of simulated end-to-end ExOR throughput for independent and dependent losses. Overall throughput is 20% less when losses are caused by shared interference.

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 3-8 illustrates the simulation results for the link graph and node pairs studied in the earlier experimental sections. 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 forwarding sets 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 it when available.

## Chapter 4

# Related Work

This chapter summarizes existing work related to ExOR. Because ExOR addresses a network problem, its design and performance are affected by characteristics of the wireless medium at the physical layer, routing strategies and medium access control. Section 4.1 begins by describing opportunistic protocols which exploit instantaneous channel conditions between specific pairs of nodes. Next, Section 4.2 describes opportunistic forwarding protocols. Section 4.3 discusses routing protocols which simultaneously use multiple paths for added reliability. Finally, Section 4.4 describes cooperative diversity routing, which combines many of the aforementioned techniques.

### 4.1 Opportunistic Channel Protocols

ExOR's opportunistic forwarding and fragment transmission mechanism share many techniques with the Opportunistic Auto Rate (OAR) protocol [19], developed to exploit 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. The signal strength of mobile links tends to fade due to changing multipath conditions, though the coherence times during which packets are likely to be received without error are tens of milliseconds long. OAR attempts to send multiple packets during those coherence windows, but also tries to share the channel fairly. 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 [10]. 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 during fragment transmission. However, because ExOR nodes are static, channel conditions are not likely to change rapidly and are instead considered to lose packets probabilistically. As a result, the outcome of the RTS/CTS exchange is not likely to predict the success of data packet transmission. Also, OAR uses opportunistic transmissions to increase throughput between fixed source-destination pairs but does not consider multi-hop routing. For example, there may be a third node to which the source has a coherent channel, which would not be discovered in the RTS/CTS exchange.

## 4.2 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” [15], 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 forwarding nodes are likely to send CTS frames simultaneously, potentially causing collisions. Jain, *et al.* [11], propose an improvement to the protocol in which the forwarders respond in a priority order specified in the initial RTS. Upon receiving the first reply, the source immediately begins transmission to that node (regardless of SNR), reducing the overhead associated with waiting for multiple replies. Roy Chowdhury presents an alternate protocol [4] which avoids the control packet overhead by using historical observations of channel conditions.

In all three protocols, it is assumed that channel measurements 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 [2] have shown delivery cannot be predicted by signal to noise ratio measurements.

### 4.3 Multiple Path Routing

Multi-path routing techniques, such as the braided routing protocol proposed by Ganesan [8], explore multiple routes simultaneously to increase robustness or performance. Such techniques typically first choose multiple path, then use them either in parallel or as a primary/backup for reliability. Other protocols, such as Opportunistic Multipath Scheduling (OMS) [3], schedule traffic over multiple paths to reduce delay by avoiding areas of high channel contention.

In both protocols, the source packet is duplicated and sent along multiple paths, which leads to additional spectrum usage. ExOR explores multiple paths simultaneously, but relies on the broadcast nature of radio to avoid extra transmissions. Furthermore, ExOR does not determine routes for each packet in advance, which allows it to take advantage of transmissions which travel long distances but occur with low probability.

### 4.4 Cooperative Diversity Routing

Laneman and Wornell [13] develop and analyze a series of information theoretic cooperative diversity techniques which, like ExOR, exploit nearby nodes which overhear transmission. In their protocols, all nodes “closer” to the destination relay a copy of the packet, increasing the likelihood of successful reception under varying channel conditions and probabilistic delivery. Closeness is determined by SNR; nodes which are closer to the destination are likely to have higher SNR than the source. By not choosing a specific route in advance, cooperative diversity is able to make use of multiple nodes which overhear the source’s transmission. 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.

ExOR fills in some of the details needed to make cooperative diversity efficient, such as scheduling transmissions and determining which nodes will relay packets. ExOR forwarding nodes use the batch map to gossip which nodes are responsible for forwarding packets. For node pairs separated by multiple hops, ExOR is likely to make fewer transmissions, because it explicitly limits the number of forwarding nodes. Laneman’s protocols, in comparison, are likely to invoke many of the nodes between the source and destination. Laneman’s protocols also assume orthogonal channels or fixed time slots, which are difficult to implement with commodity radio hardware. ExOR nodes share a single channel, and dynamically schedule

transmissions based on packet reception.

## Chapter 5

# Conclusion

This thesis presented ExOR, an integrated routing and MAC protocol for multi-hop wireless networks in which the “best” of multiple receivers forwards packets. ExOR improves performance by taking advantage of lossy links which would otherwise have been avoided by conventional unicast routing protocols. The result is a factor of two to four improvement in throughput between distant pairs of nodes in a real testbed.

ExOR operates on batches of packets. ExOR uses redundant batch maps, embedded in every packet header, to recover from lost control packets. The batch maps act as a gossiping mechanism, indirectly informing forwarding nodes of which packets were received. The batch size affects ExOR performance, but batches as small as ten packets still outperform traditional unicast routing by a factor of two or more.

Future work on ExOR includes extending the protocol to use multiple available bit rates, implementing ExOR within the context of TCP, and taking advantage of received frames which contain bit-errors.

# Bibliography

- [1] HFA3863 *Direct Sequence Spread Spectrum Baseband Processor with Rake Receiver and Equalizer Data Sheet*. Intersil Americas Inc., December 2001.
- [2] 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.
- [3] C. Cetinkaya and E. Knightly. Opportunistic traffic scheduling over multiple network paths. In *Proc. IEEE Infocom*. IEEE, September 2004.
- [4] R. Roy Choudhury and N. Vaidya. MAC layer anycasting in wireless networks. In *Second workshop on Hot Topics in Networks (HotNets II)*, November 2003.
- [5] 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.
- [6] D. De Couto, D. Aguayo, B. Chambers, and R. Morris. Performance of multihop wireless networks: Shortest path is not enough. In *First Workshop on Hot Topics in Networks (HotNets-I)*, October 2002.
- [7] 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.
- [8] 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.

- [9] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks. Technical report UCLA/CSD-TR 02-0013, UCLA CS Department, 2002.
- [10] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive mac protocol for multi-hop wireless networks. In *ACM Mobicom 2001*, September 2001.
- [11] S. Jain, Y. Ly, and S. Das. Exploiting path diversity in link layer wireless ad-hoc networks. In *preparation*, September 2004.
- [12] 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.
- [13] J. N. Laneman and G. Wornell. Exploiting distributed spatial diversity in wireless networks. In *Proc. Allerton Conference on Communications, Control, and Computing*, October 2000.
- [14] N. Laneman, D. Tse, and G. Wornell. Cooperative diversity in wireless networks: Efficient protocols and outage behavior. In *IEEE Transactions on Information Theory*, December 2004.
- [15] P. Larsson. Selection diversity forwarding in a multihop packet radio network with fading channel and capture. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(4):47–54, 2001.
- [16] R. U. Nabar, H. Bölcskei, and F. W. Kneubühler. Fading relay channels: Performance limits and space-time signal design. *IEEE Journal on Selected Areas in Communications*, June 2004.
- [17] C. E. Perkins and P. Bhagwat. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. In *Proc. ACM SIGCOMM Conference (SIGCOMM '94)*, pages 234–244, August 1993.
- [18] C. E. Perkins and E. M. Royer. Ad hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999.

- [19] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic media access for multirate ad hoc networks. In *Proceedings of ACM Mobicom 2002*, September 2002.
- [20] A. Scaglione and Y.W. Hong. Opportunistic large arrays: Cooperative transmission in wireless multihop ad hoc networks to reach far distances. In *IEEE transactions on Signal Processing*, volume 8, pages 2082–2092, August 2003.
- [21] E.C. Van der Meulen. A survey of multi-way channels in information theory. IT-23:1–37, 1977.