Effects of Loss Rate on Ad Hoc Wireless Routing

by

Daniel Aguayo

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

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2002

Author . . . .
Department of Electrical Engineering and Computer Science
May 24, 2002

Certified by . . . .
Robert Morris
Assistant Professor, MIT Laboratory for Computer Science
Thesis Supervisor

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

Effects of Loss Rate on Ad Hoc Wireless Routing

by

Daniel Aguayo

Submitted to the Department of Electrical Engineering and Computer Science
on May 24, 2002, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis uses measurements from a deployed wireless ad hoc network to illustrate the effects of link loss rates on routing protocol performance. Measurements of this network show that the radio links between the majority of nodes have substantial loss rates. These loss rates are high enough to decrease forwarding performance, but not high enough to prevent existing ad hoc routing protocols from using the links. Link-level retransmission can mask high loss rates, at the cost of substantial decreases in throughput. Simulations, driven by the observed loss rates, show that the shortest paths chosen by existing routing protocols tend to find routes with much less capacity than is available along the best route.

Thesis Supervisor: Robert Morris
Title: Assistant Professor, MIT Laboratory for Computer Science

Acknowledgments

I would like to thank the members of PDOS, and in particular the rest of the Grid project team, for their support and sense of humor. Frans Kaashoek, especially, provides both in abundance.

This thesis represents joint work with Douglas De Couto and Ben Chambers. I owe each of them my gratitude for their extensive contributions to the work presented here. In particular, Doug deserves all the credit for our extensive indoor network testbed.

I am very indebted to Robert Morris, who has been a great advisor from the beginning. Without his encouragement, enthusiasm and insight this work could not have been accomplished.

Most importantly I thank my parents and brother for their steady support throughout the years. Needless to say I could not have done this without them.

Contents

Chapter 1

Introduction

With widespread adoption of wireless networks and portable computing technologies comes the need for highly scalable networks supporting mobile nodes. One area of research that shows promise towards achieving this goal is that of *ad hoc networking*. Unlike existing networks today such as telephone networks and the Internet, ad hoc networks require no fixed, pre-existing infrastructure. Instead, nodes communicate directly among their nearest neighbors, and each forwards traffic cooperatively towards its destination.

For example, Figure 1-1 shows an ad hoc networking configuration. Radio links between nodes are depicted by dotted lines. There are a number of paths that traffic between A and D can follow, each consisting of at least three *forwarding hops*. For example, one multi-hop path is A-B-C-D.

Because an ad hoc network is self-organizing and requires no fixed infrastructure, it can be very rapidly and easily deployed. Furthermore, ad hoc networks can be robust in the face of failure: if node B in Figure 1-1 were to fail, there would still be other paths along which traffic could flow.

There are a number of potential benefits from such a system. An ad hoc network would be ideal for situations where the deployment of infrastructure would be too time-consuming or costly. In addition, such a network would support mobile nodes from its conception. As such, proposed applications include networks for developing nations, emergency disaster relief, pervasive computing, and military purposes.

Along with the numerous advantages to ad hoc networks, however, come challenges in implementation. Unlike static networks, the topologies of ad hoc mobile networks are likely

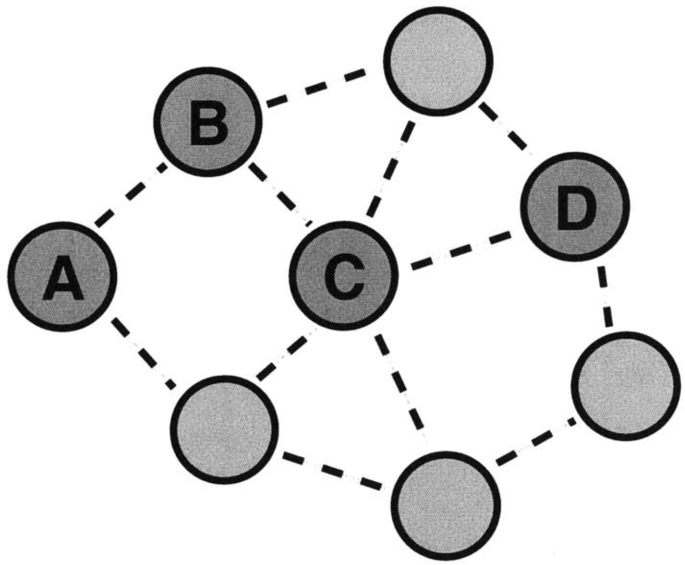


Figure 1-1: An example of an ad hoc network. Traffic from node A to node D can follow a number of multi-hop paths, including either of two three-hop routes via node B.

to be constantly changing, requiring a mechanism for quickly determining routes between hosts. There has been extensive research done towards devising dynamic routing protocols for wireless ad hoc networks. Despite this, however, there has been little done so far to implement such a system in practice.

In contrast, this thesis follows from research being done to construct a usable, eighteen-node ad hoc network [14]. Through the development and use of this testbed, we have determined that existing ad hoc routing protocols, which have been tested in simulation and detailed in publication, fail to find effective and efficient routes in real-world operation.

This failure comes from an implicit assumption which most ad hoc protocol designs have carried over from the design of routing protocols for ordinary wired networks. Protocols for wired networks usually assume that if a link delivers routing protocol packets, it will also deliver enough data packets to be useful. Put more simply, it is generally the case that wired links either work well or not at all. These designs form the basis for wireless routing protocols which favor shortest paths with no explicit attention paid to link quality, such as DSDV [18], DSR [10], and AODV [17].

Unfortunately, this assumption of link quality distribution turns out to be far from true in real ad hoc networks. This paper presents measurements taken from a prototype network which show that many links can be expected to be of intermediate quality: sufficient to pass many routing protocol packets, but exhibiting high enough loss rates to be useless, or at least less than ideal, for user data. The reason for this is that, in an ad hoc network laid out with no goals other than convenience and basic connectivity, a node can expect to be in radio contact with other nodes at a wide range of distances and signal strengths.

In this context, simple shortest-path routing is not appropriate, since it does not distinguish between good links and bad links. A path with many forwarding hops may have better links and thus be of higher quality than a path with fewer but worse links. Furthermore, preferring paths with few hops may force a routing protocol to choose long distance links which may be operating at the edge of their reception ranges, and are thus more susceptible to noise and interference.

One approach to fixing this problem is to improve the effective performance of low-quality links. Forward error correction and MAC-level acknowledgment and retransmission take this approach. For example, the 802.11 ACK mechanism resends lost packets, making all but the lowest-quality 802.11 links appear virtually loss-free.

This link-level retransmission may mask the losses on low-quality links, but it does not make them desirable for use. The retransmissions limit path throughput and reduce overall system performance. In many cases there are longer but higher-quality paths that would afford substantially better end-to-end throughput as well as higher total system capacity.

As evidence of this, this thesis uses simulations to demonstrate that existing ad hoc routing protocols often choose routes that are substantially worse than the best available.

To summarize, this thesis makes two key contributions. First, it presents an extensive set of link-quality measurements from the network. Second, it identifies and evaluates the problem of intermediate link quality as a key obstacle to the practical use of existing ad hoc routing protocols.

Chapter 2

The network testbed

The measurements presented in this thesis were obtained from experiments conducted on a testbed network which exists for ad hoc networking research. This chapter briefly describes the testbed hardware and the distance vector routing protocol which runs on the network. The testbed, including the implementation of its routing protocol, are the result of a co-operative effort led by Douglas De Couto, and its development is not part of the research conducted for this thesis.

2.1 Testbed hardware

The testbed is a collection of eighteen PCs equipped with 802.11 wireless adapters, distributed around the fifth and sixth floors of the MIT Laboratory for Computer Science such that the resulting network is connected. Radio propagation was not considered when placing nodes, except that when possible we placed nodes further from the floor, to minimize obstruction by desks, monitors, computer cases, and people. For equipment security, the nodes are all placed in offices or enclosed lab spaces. One office has two nodes.

Offices are along the perimeter of the building, and are separated by sheet-rock partitions. The center of the building contains bathrooms, stairwells, and elevators, surrounded by concrete walls. Offices are occupied by three or four graduate students, or one professor. Most have all-metal Steelcase desks and bookshelves on one or more walls. The ceilings are drop-tile, with about two feet of space between the tiles and the next concrete floor. Lounges on each floor contain printers, photocopiers, microwaves, and refrigerators.

The lab runs a wireless network using 802.11 access points. The experiments described

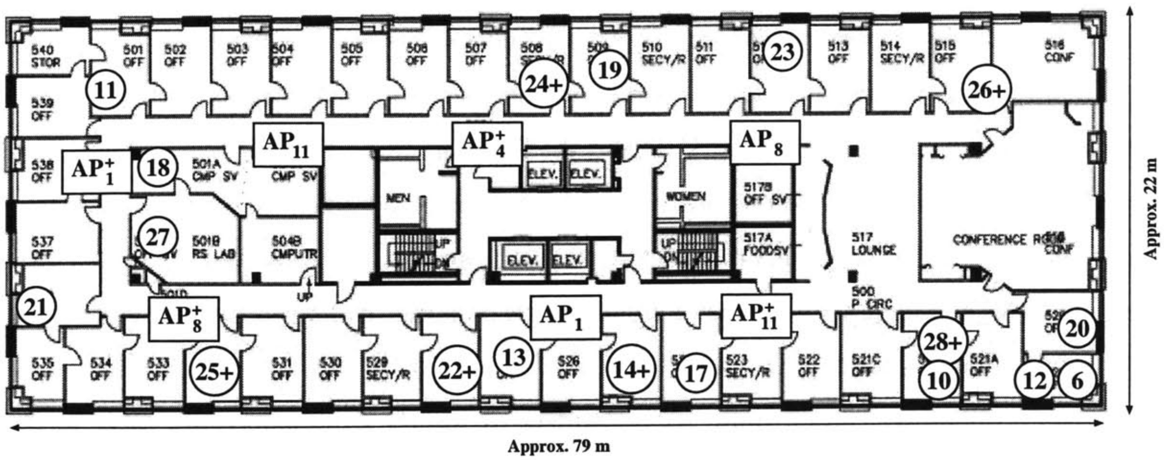


Figure 2-1: Node and wireless access point (AP) locations on the 5th and 6th floors. Nodes are circles labeled with their identifier; APs are squares labeled with 'AP' and the channel number. 6th floor nodes and APs are marked with '+'.

Firmware Version	4 (node 11: 3)
Hardware Revision	00:20
Software Revision	04:23 (node 11: 03:82)
Software Subrevision	00:00
Interface Subrevision	00:00
Bootblock Revision	01:50 or 01:43

Table 2.1: Details of the Cisco Aironet 340 cards used in the testbed network.

in this paper do not use the access points, but their presence may have affected the results. There are three on each floor, using 802.11 channels 1, 4, 8, and 11. Their locations are also shown in Figure 2-1.

All nodes in the network use the PCI version of the Cisco Aironet Model 340 wireless adapter [2], which implements the IEEE 802.11b Direct Sequence Spread-Spectrum protocol [3]. Table 2.1 shows detailed version information for the adapters we used.

Eight nodes in the network are additionally equipped with wired ethernet interfaces, in order to facilitate administration. However, all traffic between testbed nodes is forwarded wirelessly using the routing protocol discussed in the next section.

2.2 Testbed routing protocol

A simple, proactive routing protocol runs on the testbed and routes all data traffic. The protocol is a somewhat simplified variant of destination-sequenced distance vector (DSDV) described by Perkins and Bhagwat in 1994 [18]. The implementation was done in Click [12], a modular software router which runs in userlevel on the testbed nodes.

The protocol operates as follows. Each node periodically broadcasts routing advertisements which indicate its own network ID and the list of nodes to which it has a route. Each broadcast is also tagged with a sequence number which is incremented for each advertisement. For each route entry, the broadcast also lists the number of "hops" required to forward traffic to that host and the sequence number cooresponding to the route advertisement on which that route is based.

When a node overhears a route advertisement, it inserts a one-hop route entry to the sender into its own route table. Each entry listed in the advertisement is then processed separately as follows. If the node's route table doesn't already contain a route to the entry's destination, or if the entry's sequence number is more recent, the entry is inserted into the route table. If the entry has the same sequence number, it is inserted into the node's route table if and only if it is shorter – that is, has a lower hopcount. In each case, when inserting or replacing a route, the hopcount listed in the advertised entry is incremented, and the node records the sender as the "next hop."

This use of hopcount as a metric for route selection seems intuitive. By minimizing the number of forwarding hops required, we might reason, we also minimize the latency for data traffic and the total system capacity consumed by that traffic. As we will see shortly, however, this is not actually the case.

Chapter 3

Loss Rate Experiments

With the testbed running the distance vector protocol described in section 2.2, it quickly became apparent that the performance of the network left a lot to be desired. This chapter summarizes some qualitative observations of network behavior and hypothesizes how links of intermediate quality could lead the distance vector protocol to make poor routing decisions.

This chapter also presents experiments aimed at validating this theory by investigating the underlying behavior of radio links in the network. The chapter will describe the experimental methodology, present the measurement results and summarize the lessons these data provide about link loss in a wireless network in an office environment.

3.1 Motivation for Loss Rate Measurements

The research conducted for this thesis followed from a desire to explain poor network performance. Specifically, network behavior was unsatisfactory in the following ways.

Low throughput. The most obvious way in which the distance vector protocol failed to meet expectations was in the throughput which it provided. When deploying software updates, TCP transfer rates to many nodes reached only on the order of 10 to 20 kilobytes per second, which is much slower than we expected on a network with 11 Mbps links and a three- or four-hop diameter.

Periods of disconnectivity. In addition to poor throughput, TCP transfers in the network also often stalled completely. Ping packets sent from one node to another, even between nodes that were relatively close together, indicated periods of complete disconnectivity. That is, not only did many routes exhibit high end-to-end loss rates, but often this loss occurred in large numbers of sequential packets (sometimes tens of seconds worth).

Routes changing too often. The first step in debugging the above problems was to look at the route tables of various nodes in the network. Even with only a strictly local view (the route table of only one node), it was immediately apparent that routes were changing more often than they should in a static network. Very often, for example, a two-hop route to a particular destination would flip between two or more possible forwarding hops.

One potential cause of the above symptoms was packet loss. High loss rates would obviously result in poor throughput. Moreover, if network links are of varying quality, unreliable delivery of advertisements could lead to overly frequent routing changes. Delivery of route advertisements over low-quality links could cause periods of complete disconnection.

In fact, it was possible that the problems of disconnectivity and low throughput were the result of nodes forwarding data over low-hopcount routes with poor-quality links, rather than longer routes which might yield better performance. By trying to minimize the number of forwarding hops in routes, nodes will attempt to communicate with nodes which are at the periphery of their radio ranges. These links might therefore be less reliable and more lossy than shorter links.

In ordinary unicast operation, 802.11 uses link-level acknowledgements and retransmissions to mask lost packets, causing almost any link to appear loss-free and making it difficult to recognize the link's underlying quality. However, retransmissions force us to pay the price of lost bandwidth and higher latency. Also, the route advertisements used in our distance vector protocol are broadcast packets, which use no such acknowledgements and might often be lost, even on high-quality links.

To summarize, poor network performance suggested that the distance vector routing protocol, which has no notion of link quality or loss rate, was forwarding traffic over links of low or intermediate quality. To determine how common such links were in the testbed, we performed a series of experiments to measure the loss rate between each pair of nodes in the testbed. The remainder of this chapter details the experimental procedure and results.

Transmit Rate	Auto (1, 2, 5.5, or 11 Mbps)
Channel	2 (2417 MHz)
Transmit Power	30 mW
Mode	Ad hoc
Antenna	2.14 dBi rubber duck

Table 3.1: 802.11 settings.

3.2 Experimental Procedure

During each loss experiment, one node tried to broadcast a series of equally-sized packets at a constant rate, and the other nodes recorded which packets they received. In a complete set of experiments, every node took a turn at broadcasting its share of packets. The broadcast periods did not overlap, so nodes did not interfere with each other.

By using broadcast packets instead of unicast packets, the experiments avoided the 802.11 ACK and RTS/CTS mechanisms. The measured loss rate therefore reflects the underlying quality of the link, which is what determines the delivery of both data and 802.11 control packets.

Each packet contained the sender's identifier and a sequence number. The transmitting node logged the transmission time and sequence number of every packet sent. Each receiving node logged the sender's identifier, sequence number, and reception time for every successfully received packet. Signal strength and quality, provided by the 802.11 interface on a per-packet basis, was also recorded.

No routing protocol was running during these experiments: only experiment packets were sent or received on each node's wireless interface. The interfaces were configured to use a unique 802.11 SSID (network name); other relevant 802.11 parameters are shown in Table 3.1.

Prior to each experiment, each node set its card's maximum transmit rate to the lowest available setting, 1 Mbps, to prevent it from automatically changing speeds in response to link conditions. However, further investigation has shown that the cards do not honor explicit rate settings, and may have transmitted at higher rates.

Two types of experiments were conducted. Small packet tests approximated the size of 802.11 RTS/CTS and ACK packets as closely as possible. Large packet tests were more representative of typical data transfers. Small packets were 50 bytes (8 bytes data plus

UDP, IP, and Ethernet headers), and were sent at 1024 packets per second. Large packets were 1024 bytes, and were sent at 50 packets per second. The result is a send rate of just over 0.4 megabits per second (Mbps) after including 802.11 headers.

This should be well below the minimum 802.11 capacity of 1 Mbps. However, on some occasions nodes were not able to broadcast at the desired rate, perhaps because of 802.11 traffic outside our control, or interference triggering a card's carrier sense mechanism. During a typical experiment, for example, actual transmit rates varied from 0.330 to 0.389 Mbps for seventeen of the nodes. The eighteenth node, node 27, is located in a machine room and had transmit rates varying from 0 to 0.106 Mbps. We hypothesize that electrical interference in the machine room repeatedly triggered the node's carrier detection mechanism.

3.3 Results

3.3.1 Distribution of Link Qualities

Two sets of experiments were run on the testbed in the afternoon of Friday 8 February 2002, one for small packets (8-Feb-13:30-50-byte) and one for large packets (8-Feb-15:15-1024-byte). Each node transmitted for 300 seconds during each set of tests.

Figure 3-1 shows the cumulative distribution of delivery rates across all links for each of the two packet sizes. The two directions between each node pair are considered to be separate links.

The figure shows that about 50% of the links deliver no packets, while the best 20% of links deliver more than 95% of their packets. The delivery rates of the remaining 30% of links are approximately evenly distributed. Other experiments on different days and at different times confirm that there are always many links in the network which provide intermediate delivery rates.

The small packet tests exhibit higher delivery rates than the large packet tests because there is less chance of radio interference at the receivers during shorter transmissions. Regardless, the number of links with intermediate loss rates and the distribution of those loss rates is what is relevant, and nearly identical between the two types of tests.

These loss rate distributions support the hypothesis presented in section 3.1 for poor network performance. Shortest-path routing works well if all links have similar character-

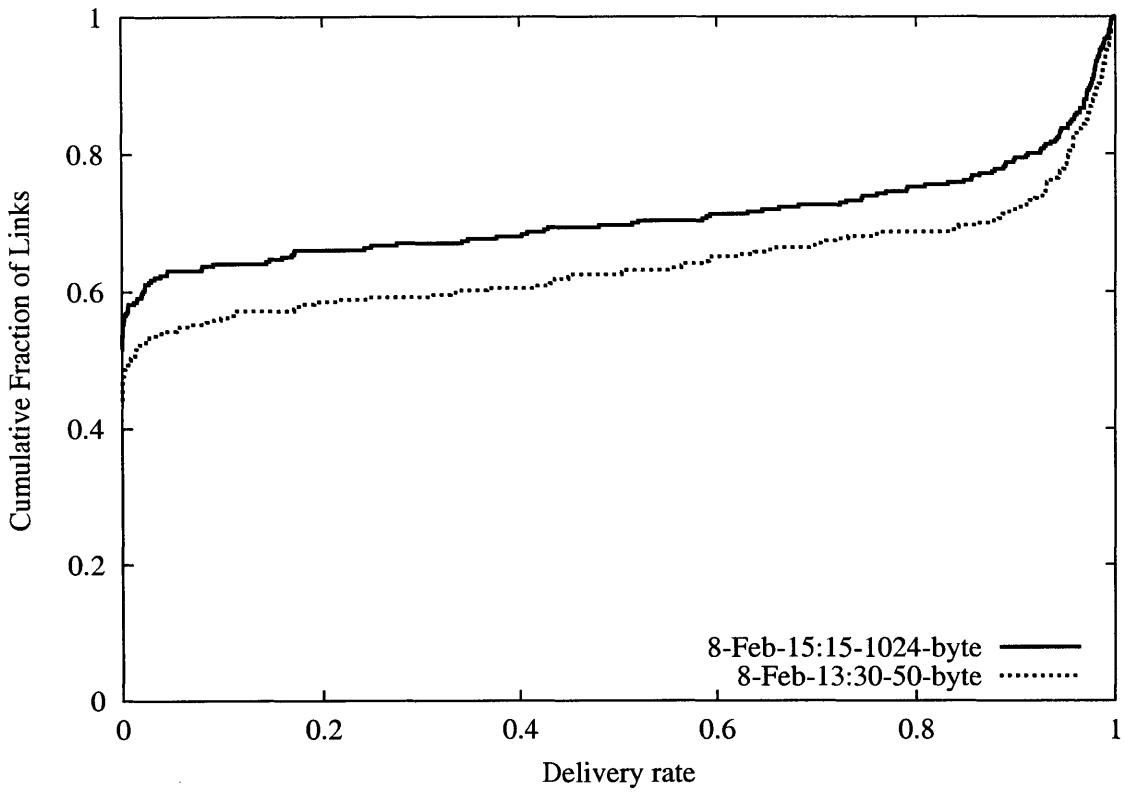


Figure 3-1: Cumulative distribution of per-link delivery rates on the network. Many links are of intermediate quality.

istics, because a longer route won't provide better end-to-end performance. But Figure 3-1 indicates that in the testbed longer routes might sometimes be preferable.

In the following sections, further analysis of the experimental results will highlight other characteristics of link loss in the testbed which could adversely affect routing protocol performance.

3.3.2 Link Quality Asymmetry

Figure 3-2 shows the delivery rates for each link pair (the two links in each direction) between two nodes for the 8-Feb-13:30-50-byte experiment, excluding pairs where neither node received any packets. Link pairs that are very good in one direction tend to be good in both directions, and pairs that are very bad in one direction tend to be bad in both directions. However, roughly 10% of the link pairs shown have asymmetric delivery rates, if we define asymmetry as a difference of more than 20% between the rates in each direction.

Because of these asymmetric links, low-loss delivery of broadcast advertisements in one

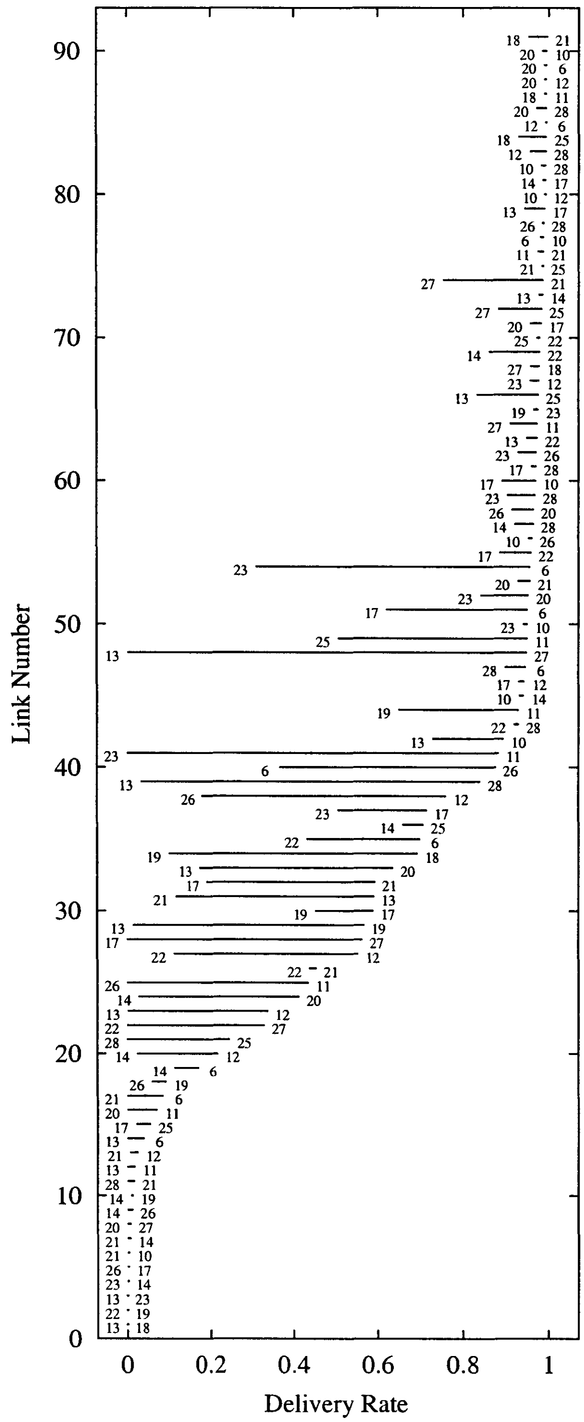
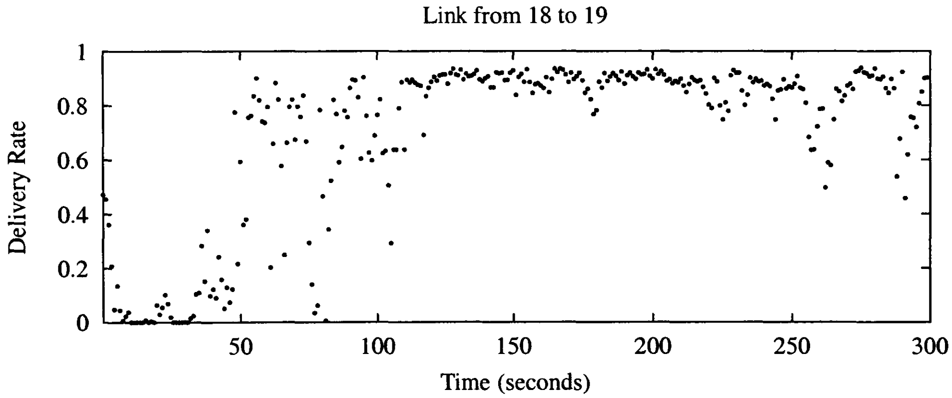


Figure 3-2: Delivery rates on each link pair during 8-Feb-13:30-50-byte, sorted by the larger delivery rate of each pair. The x values of the two ends of each line indicate the deliver rate in each direction; the numeric labels indicate the node IDs. Pairs with zero delivery rate in both directions are omitted. While most links are symmetric, a few are high quality in one direction and low quality in the other.



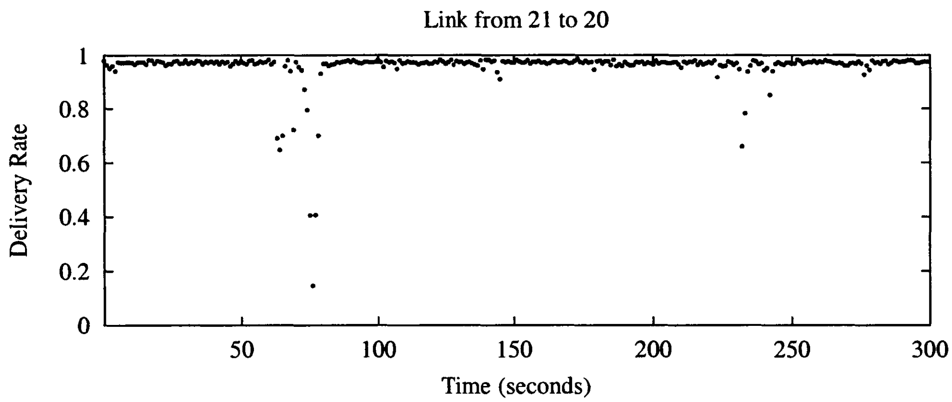


Figure 3-3: Example per-second variation in link delivery rates. Each point is the delivery rate over one second during 8-Feb-13:30-50-byte. The delivery rate of the 18→19 link fluctuates on a time-scale of seconds, while the 21→20 link is comparatively stable.

direction does not mean that unicast data delivery will be efficient: 802.11 acknowledgements require bidirectional communication. It is therefore hard to take advantage of asymmetric links, at least while still relying on link-level ACKs. In this case there is limited value in the use of such links, but it is important to recognize their occurrence and avoid them. Because the protocol running on the network does not do so, asymmetry could contribute to poor network performance.

3.3.3 Link Quality Variation Over Time

The loss rates shown in Figure 3-1 do not tell a story complete enough to verify our hypothesis. The graph shows that the *average* loss rates over the course of five-minute experiments have a wide-ranging distribution. We must, however, assure ourselves that link loss is not

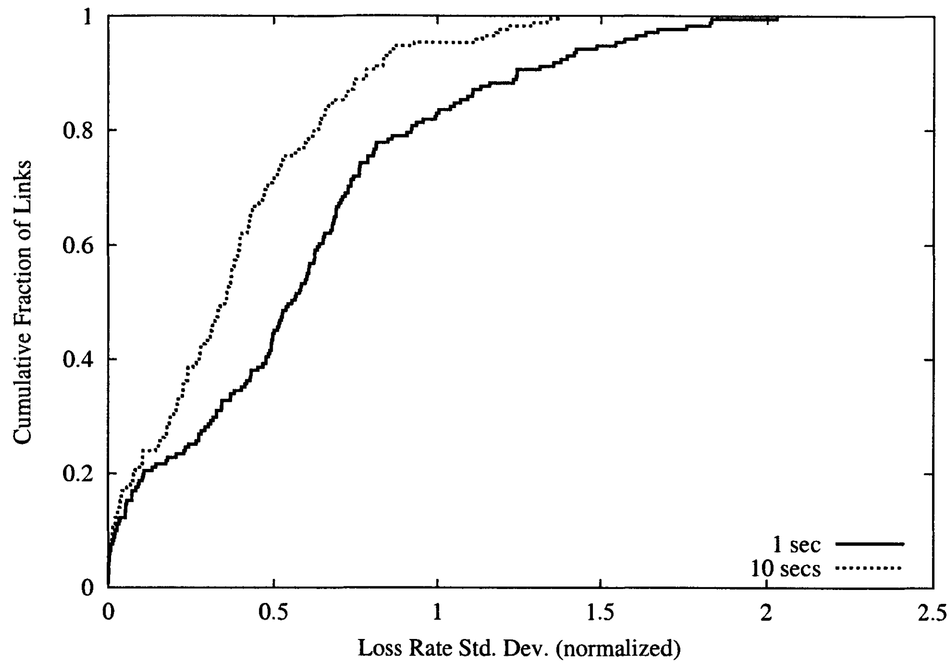


Figure 3-4: The cumulative distribution of the normalized standard deviation of short-term link *loss* rates calculated over 1 and 10 second intervals on the network (8-Feb-13:30-50-byte). Many links show significant variation in short-term loss rates over time.

simply alternating between very high and very low loss on a time scale longer than the advertisement period of our routing protocol. If this were the case, then loss might not be a concern after all, because the all-or-nothing delivery assumption would hold in the short-term and routes would be adjusted appropriately.

Figure 3-3 shows the second-by-second delivery rates for two sample links from the 8-Feb-13:30-50-byte experiment. The graphs show that delivery rates can indeed be stable at intermediate values. Link quality can also sometimes change quickly.

Figure 3-4 summarizes loss rate variation over time for all links. The graph shows the standard deviation of the loss rate among 1- and 10-second periods for each link. The standard deviations are normalized by their respective means, and then plotted as a cumulative distribution. Loss rates are shown rather than delivery rates for this analysis because quality changes on links with low loss are most important: links with high loss are useless for data traffic regardless of their variation.

These distributions show that quite a few links vary greatly on these times scales. For example, half of the links had standard deviations in their 1-second loss rates that exceeded half of the mean.

3.4 Reported signal strength and quality

The results presented in sections 3.3.1 and 3.3.2 showed that a practical ad hoc routing protocol must estimate link loss rates and adjust routing decisions appropriately. The graphs shown in section 3.3.3 indicate that any such estimator must be responsive to rapid changes in loss rate.

Unfortunately, it is unclear how to estimate loss rate quickly and accurately. One method would be to send periodic broadcast pings and accumulate loss statistics over a window of time. A long window, however, might limit responsiveness to link changes and mobility, while shortening the measurement period would make the estimate more coarse and unreliable. Broadcast pings also limit system capacity.

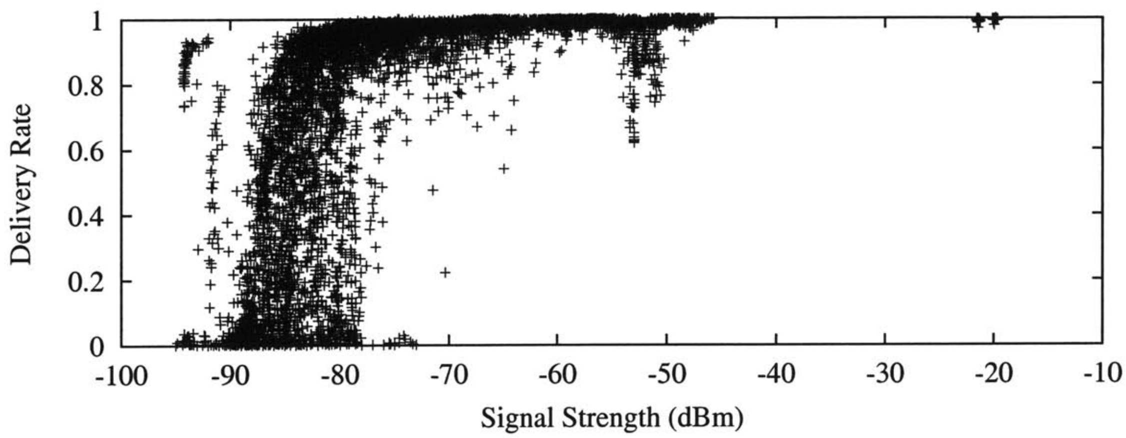
One appealing alternative to the accumulation of link loss rate statistics is the use of measured signal strength, which is reported by the 802.11 hardware on a per-packet basis. The 802.11 hardware reports two numbers, "signal strength" and "signal quality." The actual physical meaning of these numbers is unclear, but documentation from the Aironet hardware and its underlying chipset suggests "strength" represents signal-to-noise ratio (SNR) in the received signal, and "quality" is some aggregate measure which accounts for SNR as well as missed beacons. (In ad hoc mode, all 802.11 cards transmit beacons, by default once every 100 milliseconds.)

We might expect the received power or SNR to provide an instantaneous measure of link quality, which would be suitable for loss rate estimation. The use of such a measure is proposed in [7] and [5].

Figure 3-5 shows a plot of delivery rate versus each of the two metrics provided by the card firmware, from the 8-Feb-13:30-50-byte experiment.

It is clear that there is no simple, general relationship between loss rate and either of these two values. Few links seem to have a strong correlation. These values seem to be suitable only for thresholding links into one of two or three wide "bins" of link qualities, but it is questionable whether that alone would be a sufficient loss rate estimator.

The quality metric seems to be a slightly better predictor in the case of links starting at node 11. The data from these links form the outlying band of points to the right of most others. The fact that those points come from links that *originate* at node 11 suggests that the quality metric is indeed counting missed beacons. Node 11 is the only node with an earlier firmware revision than the others, and it is possible that between versions the send rate of beacons was changed. Presumably if the same firmware ran on every node, the plot would consist entirely of a band similar to the one on the left.



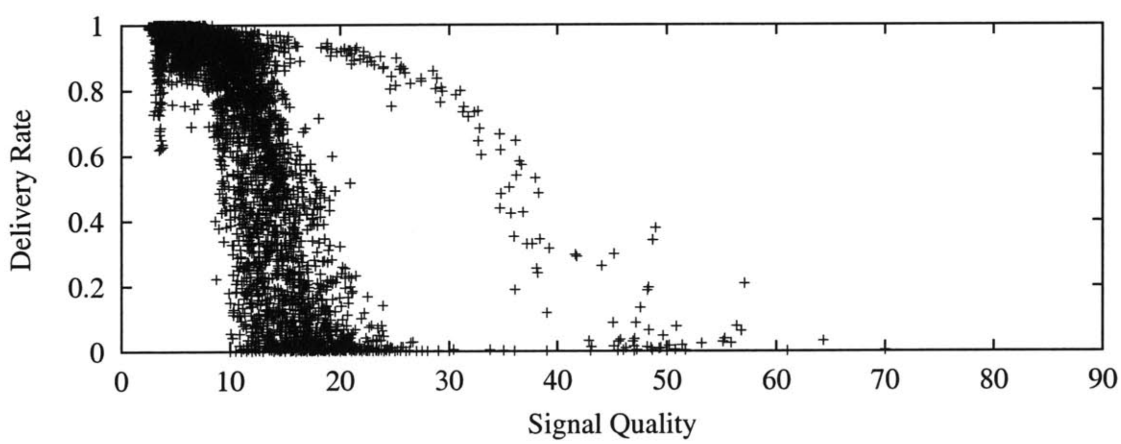


Figure 3-5: Delivery rate versus quality and signal strength as reported by the Aironet card. Each point is the delivery rate over 1 second for some link, plotted against the link's average signal strength or quality during that interval, from 8-Feb-13:30-50-byte. There is no simple relationship between signal strength and delivery rate; quality is a slightly better predictor.

Chapter 4

Simulations

To better understand the effects of the observed link characteristics on the performance of existing routing protocols, we evaluated DSDV and DSR on a simulated network with loss characteristics based on measurements of our testbed. The evaluation is based on end-to-end throughput achieved on the routes selected by these protocols, compared to the best achievable throughput.

4.1 Simulation setup

These simulations use version 2.1b3 of the *ns* simulator [6] with the CMU wireless extensions [8]. All the simulations used the 2 Mbps 802.11 implementation included with the CMU extensions. RTS/CTS was enabled for all unicast transmissions. We used the included implementations of DSDV and DSR with the default parameters, except that ARP was disabled.

The simulated network consisted of eighteen nodes, as in our network testbed. To model our observed loss rates, we replaced the default radio propagation model with a simple lookup which refers to a table of average loss rates from the 8-Feb-13:30-50-byte experiment. Each radio transmission (including each phase of the four-way RTS/CTS/Data/ACK exchange) is dropped with a probability cooresponding to this loss rate.

Because we replaced the radio propagation model, these loss rates also determine interference and carrier sense between nodes. The CMU model features two power thresholds: one at which a node can receive an incoming packet, and one at which it cannot receive the packet, but can still sense and be interfered by it. In our simplified model, reception and interference both occur with a probability equal to our observed average delivery rate. In reality, the probability of interference should be higher than the delivery probability, so our model overestimates delivery rates. Using a single probability leads to the delivery of some packets which should have been delayed because of carrier sense or lost due to radio interference.

Each simulation begins with one node in the network sending 1024-byte packets to another node at a rate of one packet per second. This affords the routing protocol time to establish routes and settle into steady-state operation. After one minute, the source node begins sending a steady stream of 2 Mbps traffic, in 1024-byte packets, for five simulated minutes. In a complete set of experiments, two simulations are conducted for every pair of nodes in the network — one for each sending direction.

We evaluate the routing protocols by the end-to-end delivery rate of the 2 Mbps traffic. This measure reflects the underlying quality of the links along the selected paths, because retransmissions reduce available capacity. It also penalizes longer routes, which have reduced available capacity because of interference between successive nodes in the route. [13]

For comparison, we also estimated the throughput of the "best" route between each pair of nodes. For each pair, we generated a list of all routes fewer than six hops in length and ranked them based on the expected total number of data and ACK transmissions required for the successful delivery of a single data packet. For each of the top ten routes, we ran the same simulation described above and took the route with highest throughput as "best."

In reality, to definitively determine the true "best" would require an analysis which accounts for probability of interference between each pair of nodes in the route, or one which simply tests all possible routes. Nevertheless, our "best" routes can only underestimate the optimal route for the given conditions.

4.2 Results

The results of the simulations are shown in Figures 4-1 and 4-2. In each graph, one vertical line is shown for each communicating pair of nodes, with each direction plotted as a separate line. The data is sorted along the horizontal axis by the throughput given by the best route we found. Throughput in Mbps is shown in the top graphs, and average route length is in the bottom. Since we used only one static route for each "best" test, the lengths of those

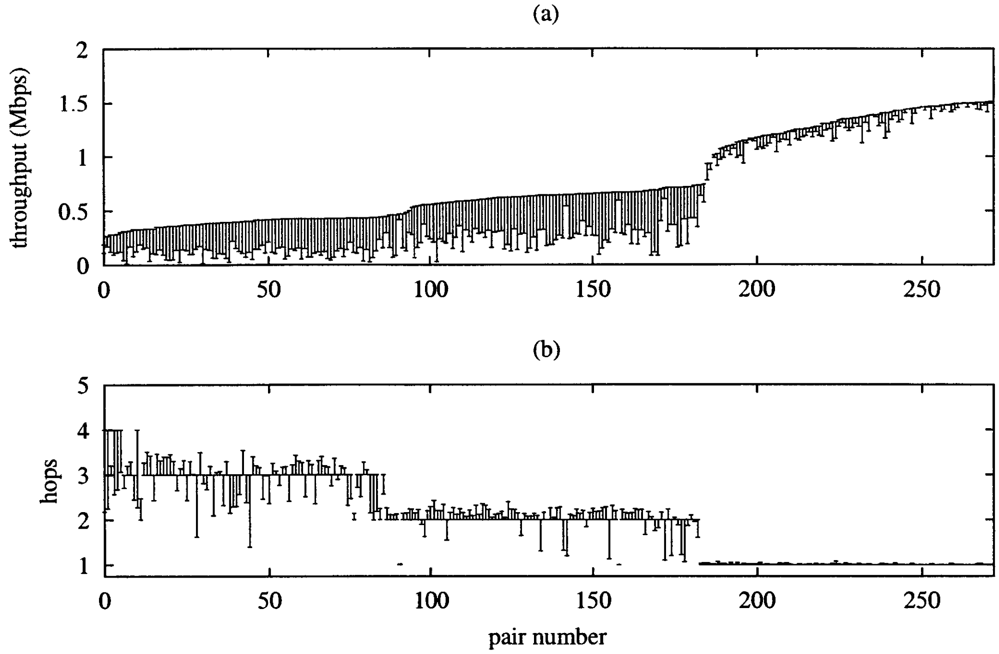


Figure 4-1: (a) A comparison between the simulated end-to-end throughput made available using DSDV (marked by the bottom of each vertical line) and the "best" route. One line is plotted for each pair of nodes. (b) The corresponding average route length used by successfully delivered packets for each of the pairs in (a).

routes are seen as horizontal lines at integer values in the bottom graphs.

The highest throughput shown for any route is roughly 80% of the total available 2 Mbps. This is about what is expected after accounting for bandwidth consumed by link-level headers and RTS, CTS and ACK packets. The throughput plots have three regions, corresponding to the lengths of the best routes. Longer routes have lower throughput because of interference between the successive hops of the route. Thus a two-hop path can deliver no more than 50% of the available one-hop throughput, and a three-hop path can do no more than 33%.

4.2.1 DSDV

The DSDV results in Figure 4-1 are particularly striking. End-to-end throughput for DSDV's multiple-hop routes falls far short of the best possible, averaging just 41% of best among two-hop routes and 24% of best in three-hop routes. Even among one-hop routes, performance averages 5% less than the best possible.

DSDV's low throughputs result directly from the effects described in Section 3. These lead to poor performance in several ways:

Missed updates on high-quality links. A missed route update on a link between two nodes will cause those nodes to use an alternate, multi-hop, route between them, even if the link is otherwise high-quality. This alternate route will stay in use until the next routing period, which in the *ns* implementation is 15 seconds long. Failures of this nature are seen in the graph as pairs where the average route length used by DSDV is higher than optimal. These failures become more likely the longer the ideal route, so one can expect the performance of DSDV to degrade further in larger networks.

Updates received over poor quality links. If a link exists between sender and receiver with a 50% delivery rate, then those nodes will use that link 50% of the time, despite the fact that it requires on average two transmissions of each data packet. An asymmetric link can exhibit an even more serious problem, by delivering updates in one direction with high probability while providing decreased bandwidth in the other. These failures are seen as pairs where the average route length used by DSDV is lower than optimal.

Multiple paths of equal length. The most common failure of DSDV results from the fact that between any pair of nodes, there are usually multiple paths of the optimal length, most of which have suboptimal quality. From these choices, the protocol will always select the route it hears about first for each sequence number. As the nodes get further apart and the number of paths increases, this first-received path is less likely to be the ideal. This is the reason that for many pairs, performance is well below the ideal while the utilized route length appears to be close to 'correct'.

4.2.2 DSR

Figure 4-2 shows that DSR fares much better than DSDV, performing at 98% of maximum on one-hop routes. On longer routes, however, performance degrades dramatically, averaging just 85% of maximum on two-hop routes, and 29% on three-hop routes.

In cases in which the source and receiver nodes are far apart, there are more potential routes between them, and it becomes less likely that a query will return the ideal route. However, DSR never changes routes except in the case of transmission failure. Retransmission attempts make these failures very unlikely, so it is not surprising that performance is

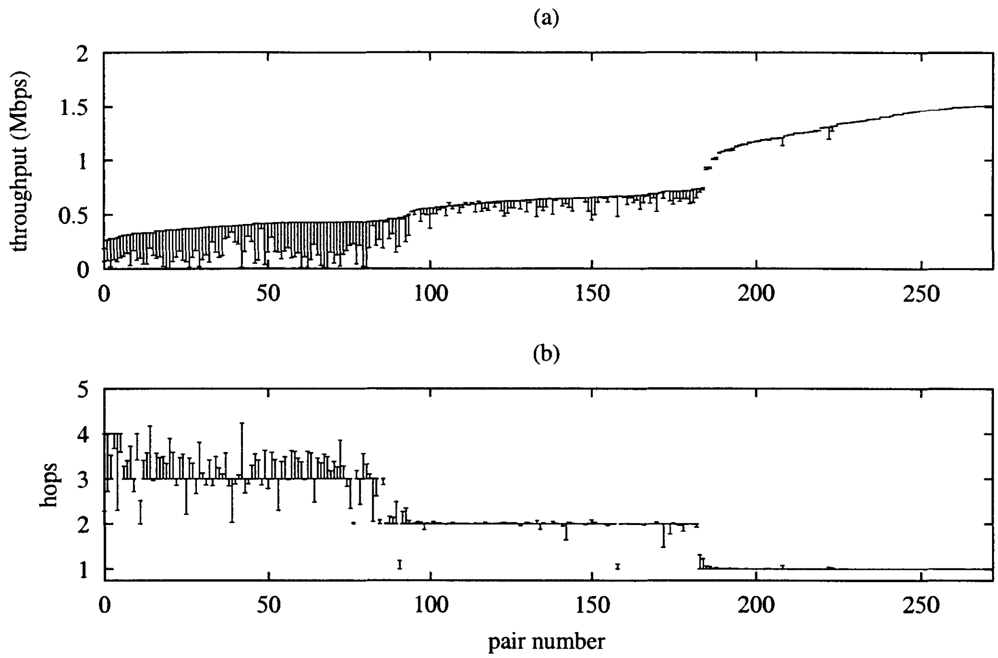


Figure 4-2: A comparison similar to the one shown in Figure 4-1, but for DSR: (a) shows end-to-end throughput made available by DSR for each pair of nodes, compared to "best". (b) shows the corresponding average route length used for successfully delivered packets.

reduced as nodes move further apart. Unfortunately it is clear that the average performance of DSR in a network will fall as the network grows.

Figure 4-2 also shows that, for nodes that are far apart, DSR often uses routes longer than the optimal. This is a consequence of the "route repair" mechanism in the protocol, which operates when a transmission failure occurs. The node attempting to forward the packet consults its list of cached routes (which it obtains from its own route queries and from overheard traffic), to see if it has an alternate route to the packet's intended recipient. Thus, when any link along that route fails to deliver a packet, the network begins to use an alternate, most likely longer, route. This becomes more likely the more hops there are in the route the network tries to use.

The end effect of DSR switching to a different route only when the current one fails is that it keeps switching routes until it finds one that doesn't produce any link delivery failures. As long as that link continues to deliver traffic, DSR will continue to use it. For this reason, DSR tends avoid long-term use of low quality routes, and thus outperforms DSDV.

Chapter 5

Related work

Previous work that considers quality-based wireless link selection for multihop ad hoc networks has been carried out using both simulations and prototype networks.

In [9], the mean time for which a link will be "available" is predicted based on the positions and motions of the nodes at each end of the link, and a parameter to adapt to environmental changes. The reliability of a link is defined as the mean time it is available, and the reliability of a route, or path, is defined as the minimum reliability (T_{min}) of the route's links. The best route is that with the maximal T_{min} and the minimal number of links. Simulator results in [9] show that this metric provides better results that just using shortest paths.

In *preemptive routing* [7], low received signal strength is used to predict when a link, and thus a route, will break. When signal strength becomes low, the routing protocol can preemptively select a new good route to the same destination, on the assumption that low signal strength indicates that the other end of the link will soon be out of range. New routes are selected so that all links have a signal strength greater than some threshold. Low signal strengths are monitored for a period of time to ensure that random signal fades do not prematurely trigger a route change. Simulations with DSR [7] show that this technique decreases the number of broken routes, and generally decreases latency.

Signal stability-based adaptive (SSA) routing [5] also uses signal strength to choose routes. SSA classifies a link as weak or strong by comparing the link's signal strength to a threshold. Since SSA assumes that links are symmetrical, the signal strengths of packets received *from* a destination are used to classify the link *to* that destination. SSA tries to pick routes that only have strong links. SSA also adds a stability criterion to only consider links that have been classified strong for more than a specified time; however, [5] reports that this is not effective in reducing broken routes.

The CMU Monarch Project constructed a testbed network [16, 15] running DSR [10]. Their implementation included a quality metric [19] based on predictions of link signal strengths; these predictions were calculated using a radio propagation model that considered the locations and movement of the nodes at the end of each link. The quality of a link is defined as the probability that the link's signal strength will be above some reception threshold; a route's quality is the product of its links' qualities. Higher quality routes are preferred when selecting a new route.

The above techniques try to pick links in mobile networks so that the links won't break soon, or to detect when links are about to break. They all assume that link quality is a simple threshold function of signal strength (except for [9], which assumes quality is a function of distance). Figure 3-6 shows that, in the network testbed, there is no clear, single relationship between measured signal strength and loss rate.

The DARPA packet radio network (PRNET) [11] directly measures bidirectional link quality, by counting the fraction of packets received on each link between routing advertisements. This is possible since, unlike 802.11, the MAC and routing protocol are integrated, and the routing protocol can examine every packet seen by the interface. These measurements are smoothed, and links are classified as good or bad based on a threshold, with hysteresis. Bad links are not considered when choosing routes, using a shortest paths algorithm. Although the PRNET radios provided measurements about received signal power and noise, these were not used to pick routes.

The combat net radio system [4] also directly measures link quality using received packet counts, and links are classified as good, bad, or non-existent by comparing the measured qualities to thresholds. Routes are chosen to minimize the hopcount, except that routes with bad links are avoided. Route quality is defined as the number of bad links in the route.

Chapter 6

Conclusions

The contribution of this thesis is an analysis of how lossy radio links can severely impact ad hoc routing protocol performance. Qualitative observations of performance of an 18-node indoor testbed network suggested that packet loss interfered with routing protocol operation. An extensive set of broadcast packet loss experiments on the testbed confirmed that lossy links were common in the network. Moreover the results illustrated three important points about link behavior:

1. The distribution of link delivery rates is not strictly bimodal. Links frequently have intermediate delivery rates.
2. Asymmetric links are not uncommon.
3. Link delivery rates can change quickly.

Link-level retransmissions minimize actual packet loss between nodes with lossy links, but the use of such links results in reduced bandwidth and overall system capacity.

Simulations based on the measured average loss rates show how links of intermediate quality reduce the performance of the DSR and DSDV protocols, which consistently choose suboptimal routes.

These results indicate that any successful ad hoc routing protocol will need to take loss rate into account during route selection. This will require some estimation of loss rate in *both* directions of each network link. The estimate must rapidly adapt to changing conditions or node mobility. Measurements of signal strength and quality reported by the 802.11 firmware show that neither of these metrics are sufficient alone.

Bibliography

[1] *Proceedings of the IEEE*, 75(1), January 1987.

[2] *Using the Cisco Aironet 340 Series PC Card Client Adapters.* Cisco Systems Inc., March 2000. Part number: OL00379-01.

[3] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* New York, New York, 1997. IEEE Std. 802.11–1997.

[4] Brian H. Davies and T. R. Davies. The application of packet switching techniques to combat net radio. In *Proceedings of the IEEE* [1].

[5] Rohit Dube, Cynthia D. Rais, Kuang-Yeh Wang, and Satish K. Tripathi. Signal stability-based adaptive routing (SSA) for ad hoc mobile networks. *IEEE Personal Communications*, February 1997.

[6] Kevin Fall and Kannan Varadhan. *ns* notes and documentation. Technical report, UC Berkeley, LBL, USC/ISI, and Xerox PARC, November 1997.

[7] Tom Goff, Nael B. Abu-Ghazaleh, Dhananjay S. Phatak, and Ridvan Kahvecioglu. Preemptive routing in ad hoc networks. In *Proc. ACM/IEEE MobiCom*, July 2001.

[8] CMU Monarch Group. CMU Monarch extensions to *ns*.

[9] Shengming Jiang, Dajiang He, and Jianqiang Rao. A prediction-based link avilability estimation for mobile ad hoc networks. In *Proc. IEEE Infocom*, April 2001.

[10] David 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.

[11] John Jubin and Janet D. Tornow. The DARPA packet radio network protocols. In *Proceedings of the IEEE* [1].

[12] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. The Click modular router. *ACM Transactions on Computer Systems*, 18(4), November 2000.

[13] Jinyang Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee, and Robert Morris. Capacity of ad hoc wireless networks. In *Proceedings of the 7th ACM International Conference on Mobile C omputing and Networking*, pages 61–69, Rome, Italy, July 2001.

[14] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris. A scalable location service for geographic ad hoc routing. In *Proc. ACM/IEEE MobiCom*, August 2000.

[15] David A. Maltz, Josh Broch, and David B. Johnson. Experiences designing and building a multi-hop wireless ad hoc network testbed. CMU-CS-99-116, Carnegie Mellon University, School of Computer Science, March 1999.

[16] David A. Maltz, Josh Broch, and David B. Johnson. Quantitative lessons from a full-scale multi-hop wireless ad hoc network testbed. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, September 2000.

[17] Charles Perkins, Elizabeth Royer, and Samir R. Das. Ad hoc On demand Distance Vector (AODV) routing. Internet draft (work in progress), Internet Engineering Task Force, October 1999.

[18] Charles E. Perkins and Pravin Bhagwat. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. In *Proc. ACM SIGCOMM Conference (SIGCOMM '94)*, pages 234–244, August 1993.

[19] Ratish J. Punnoose, Pavel V. Nitkin, Josh Broch, and Daniel D. Stancil. Optimizing wireless network protocols using real-time predictive propagation modeling. In *Radio and Wireless Conference (RAWCON)*, August 1999.