Effects of Loss Rate on Ad Hoc Wireless Routing

Douglas S. J. De Couto Daniel Aguayo Benjamin A. Chambers Robert Morris*

8 March 2002

Abstract

This paper uses measurements from two deployed wireless ad hoc networks to illustrate the effects of link loss rates on routing protocol performance. Measurements of these networks 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.

Based on these observations, we present a routing metric intended to allow routing protocols to find good routes in wireless ad hoc networks. The metric is the expected total number of transmissions required to deliver a packet along a route. This metric favors routes with high throughput and low total impact on spectrum. It is expected to perform better than existing techniques that eliminate links based on loss rate thresholds.

1 Introduction

Routing protocols designed for ordinary wired networks usually assume that a link either works or doesn't work. More specifically, it is generally the case that if a link delivers routing protocol packets, it will also deliver enough data packets to be useful. To a great extent this assumption has been carried over into the realm of multi-hop wireless ad hoc networks. For example, DSR [13], AODV [19], and Grid [15] favor shortest paths, with no explicit attention paid to link quality.

Unfortunately, the assumption of bimodal link quality turns out to be far from true in real ad hoc networks. This paper presents measurements taken from two prototype networks which show that many links can be expected to be of intermediate quality: good enough 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 long path may have better links and thus be of higher quality than a shorter path with bad links. Furthermore, preferring short paths 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. This paper uses simulation and measurements on a real network to demonstrate that existing ad hoc routing protocols often choose routes that are substantially worse than the best available.

One approach to fixing this problem is to improve the effective performance of low-quality links. Forward error correction, MAC-level acknowledgment and retransmission, and solutions such as Snoop-TCP [5] and Tulip [18] all take this approach. For example, the 802.11 ACK mechanism resends lost packets, making all but the lowest-quality 802.11 links appear loss-free.

Link-level retransmission may mask the losses on low-quality links, but it does not make them desirable for use in paths. The retransmissions reduce path throughput and reducing overall system performance. In many cases there are longer but higher-quality paths that would afford substantially better end-to-end capacity as well as higher total system capacity. As evidence of this, we will show that shortest-path ad hoc routing with per-link masking leaves a good deal of performance on the table by choosing sub-optimal routes.

One potential solution would be to choose routes based on observed link loss rates. A number of problems must be overcome in order to make this work. First, a specific path metric must be chosen; this paper argues that neither total nor maximum loss rate is appropriate, but instead the total expected number of transmissions. This metric properly penalizes longer paths. Second, routing proto-

*Parallel and Distributed Operating Systems Group, MIT Laboratory for Computer Science. Email: {decouto, aguayo, bac, rtm}@lcs.mit.edu. The authors would like to thank Jinyang Li for her help with data analysis. This research was funded in part by NTT corporation under the NTT-MIT collaboration.

cols tend to have a natural bias in favor of shortest paths, even if they use some other metric; this is because advertisements along the shortest path arrive first. Third, observation of a real ad hoc network reveals that loss rates change rapidly with time, even in non-mobile networks. This means that the long-term measurements needed for precise loss rate measurements are not practical unless augmented with predictions based on short-term information.

In summary, this paper makes four main contributions. First, we present an extensive set of link-quality measurements from an 18-node indoor ad hoc network, and a 7-node rooftop network. Second, we identify the problem of highly variable link quality as a key obstacle to practical use of existing ad hoc networks. Third, we evaluate this problem's impact on existing routing protocols. Finally, we describe the design, implementation, and analysis of a loss-based routing metric that copes well with a wide distribution of link loss rates.

2 Overview of 802.11

This section briefly reviews some relevant details of the IEEE 802.11 standard for wireless networks [6], which describes a set of protocols for the physical and MAC layers. This paper considers only 802.11 in ad hoc mode, which allows nearby nodes to communicate directly with each other, without any intervening access point.

2.1 Physical Layer

The physical layer used in this paper is direct sequence spread spectrum (DSSS). In the United States, DSSS can be used on any of 11 channels centered every 5 MHz from 2412 to 2462 MHz. Since channels must be at least 30 MHz apart to be non-interfering [6, section 15.4.6.2], at most two completely non-interfering channels can be used simultaneously. The standard defines modulation schemes for a variety of bit rates ranging from 1 to 11 megabits per second (Mbps). Adapters can switch rates for each packet they send.

2.2 MAC Layer

The 802.11 medium access control (MAC) layer provides mechanisms for carrier sense, collision avoidance, and collision detection.

A node implements carrier sense by deferring transmission until it can hear no other node. Broadcast packets are controlled by this mechanism alone.

Basic carrier sense is not sufficient in cases where the receiver is already receiving a packet that the transmitter cannot hear. For this reason, 802.11 controls unicast packets with an additional RTS/CTS mechanism. Before sending a data packet, the sender sends a short RTS message; if the receiver gets the RTS and is idle, it returns a CTS packet, giving the sender permission to send the whole data packet. To avoid unnecessary overhead from RTS/CTS exchanges, they are disabled for data packets whose size is less than the *RTS threshold*.

While carrier sense and RTS/CTS decrease the probability of collisions, they do not eliminate them. 802.11 specifies that receivers return an ACK message for each unicast packet successfully received. If the sender hears no ACK before a specified timeout, it resends the packet after a backoff period. The maximum number of retransmissions is a configurable parameter known as the *short retry limit* or *long retry limit*, depending on the size of the packet. The 802.11 ACK mechanism addresses both the problem of collisions between simultaneous transmissions, and the problem of packets corrupted by noise or interference.

802.11 transmitters can fragment unicast packets larger than a specified *fragment threshold*, allowing each fragment to be separately acknowledged or retransmitted.

3 Measured Delivery Rates

We conducted a set of experiments to characterize the underlying behavior of radio links in our networks. This section presents the main empirical lessons from our experiments, followed by a description of our wireless testbeds and the experimental methodology. The section concludes with a detailed look at the experimental data.

3.1 Empirical Lessons

The experiments presented below confirm three observations about wireless links which affect how multi-hop routing protocols should be designed to work over these links. These observations are that wireless links vary widely in their delivery rates, that some links are asymmetric, and that link delivery rates can vary quickly.

3.1.1 Link Variation

Most routing protocols use hop count as their link metric: they try to choose routes with the smallest number of links. This works well if all links have similar characteristics, which means using a longer route won't improve end-to-end performance. However, as we show below, wireless links can offer a wide range of delivery rates. In this case, a longer route made up of links with high delivery rates can have better end-to-end performance than a shorter route which is made up of links with low delivery rates.

3.1.2 Link Asymmetry

Our results show that some wireless links have asymmetric delivery rates. This means that low-loss delivery of routing updates in one direction does not mean that sending data back along the route will work well. It turns out to be hard to take advantage of asymmetric links with protocols (such as 802.11) that use link-layer acknowledgments. The best approach to asymmetric links, therefore, is to recognize and avoid them if possible.

3.1.3 Link Variation Over Time

The last result is that link performance varies over several different time scales, from hours to seconds. A routing protocol could measure, for example, the delivery rate of its routing updates, and use these to predict link quality. However, since calculating precise delivery rates requires counting lost and received packets over many transmissions, direct measurements may not react quickly enough to frequent changes in link performance. One alternative is to use measured signal strength. However, as we show in Section 5.2, signal strength and "quality" indicators reported by typical 802.11 hardware does not correlate very closely with delivery rates.

3.2 Testbeds

We used two wireless testbeds in our experiments. One is an indoor network, while the other is an outdoor rooftop network.

3.2.1 Indoor Network

The indoor testbed is a collection of PCs equipped with 802.11 wireless adapters distributed around our building, as shown in Figure 1. We placed eighteen nodes casually around the fifth and sixth floors, such that the resulting network was connected. Radio propagation was not considered when installing 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 middle 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.

	340	**350**
Firmware Version	4 (node 11: 3)	4
Hardware Revision	00:20	00:22
Software Revision	04:23 (node 11: 03:82)	04:25
Software Subrevision	00:00	00:05
Interface Subrevision	00:00	00:00
Bootblock Revision	01:50 or 01:43	01:50

Table 1: Cisco Aironet 340 and 350 details.

The lab runs a wireless network using 802.11 access points. The experiments described in this paper do not use the access points, but the access points may have affected the results. There are 3 access points on each floor, using 802.11 channels 1, 4, 8, and 11. The access points are also shown in Figure 1.

All nodes in the indoor 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 [6]. The first column of Table 1 shows detailed version information for the adapters we used in the indoor network.

3.2.2 Rooftop Network

The rooftop network consists of seven nodes distributed over a region approximately one square kilometer in area. Node 30 is located on the ninth floor of our lab building, while the rest are distributed in the residential neighborhood located to the northwest. Their locations are shown in Figure 2. The node in our building is equipped with a 13.5 dBi Yagi (directional) antenna which is indoors and points out the north window. It is connected to the 802.11 interface with a 20-foot (1.3 dB loss) cable. The remaining nodes are each equipped with a roof-mounted 5.2 dBi omnidirectional antenna, connected to the 802.11 interface with a 50-foot (3.4 dB loss) cable.

All but one of the houses hosting the rooftop nodes are three stories high, and have their antennas mounted on a mast 5-10 feet above the level of the roof. There is one node (node 32) located in a two-story house, and its antenna is mounted 15-20 feet above the roof height. Some pairs of antennas are within line-of-sight of each other, while others have varying amounts of obstructions between them, including slightly taller buildings, trees, or other obstructions.

All the rooftop nodes use the PCI version of the Cisco Aironet Model 350 wireless adapter, which is very similar to the Model 340 used by the indoor network. The most significant difference between the two models is that the 350 has a 100mW output power, compared with the 30mW output power of the 340. The second column of Table 1 shows detailed version information for the Model 350 cards we are using.

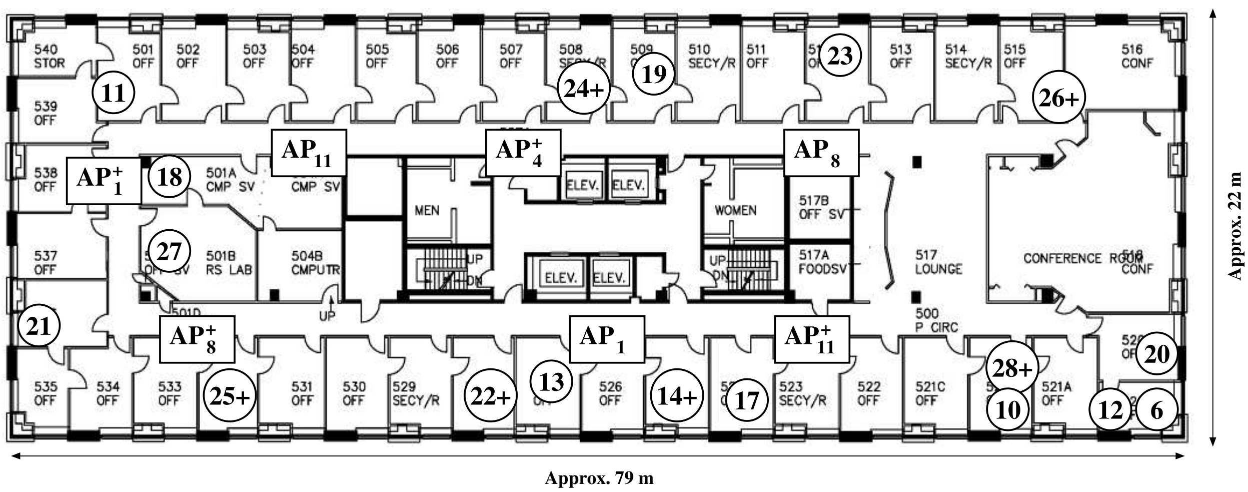

Figure 1: Node and lab access point (AP) locations on the 5th and 6th floors. A map of the indoor testbed network. Nodes are circles labeled with their identifier; APs are squares labeled with 'AP' and the channel number. 6th floor nodes and APs are marked with '+'.

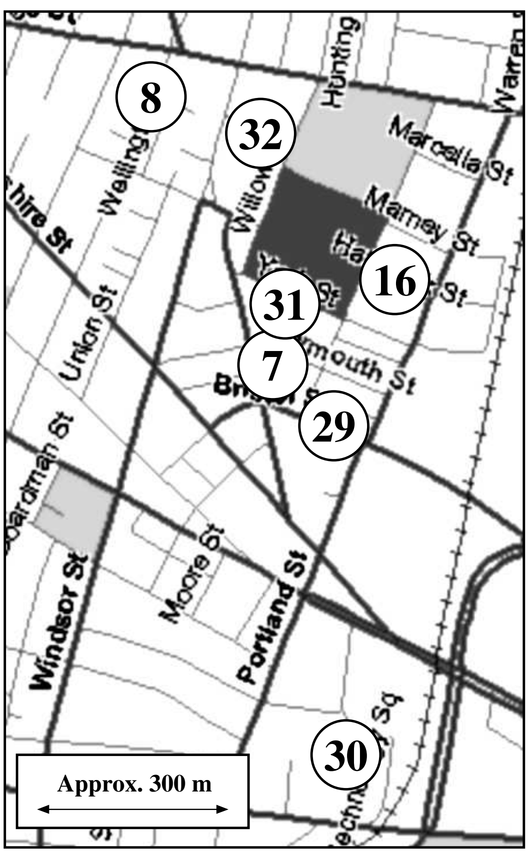

Figure 2: A map of the outdoor rooftop network testbed. Nodes are labeled with their network identifier. Node 30 is located on the ninth floor of our building and is equipped with a Yagi (directional) antenna, while the other six nodes are equipped with omnidirectional antennas.

Transmit Rate	Auto (1, 2, 5.5, or 11 Mbps)
Channel	2 (2417 MHz)
Transmit Power	30 mW (indoor), 100mW (rooftop)
Mode	Ad hoc
Antenna	2.14 dBi rubber duck (indoor), 5.2 dBi omnidirectional or 13.5 dBi Yagi (rooftop)

Table 2: 802.11 settings.

3.3 Experimental Procedure

We performed a series of experiments to determine the loss characteristics between each pair of nodes in the testbeds. During an experiment, one node tries to broadcast a series of equally-sized packets at a constant rate, and the other nodes record which packets they receive. In a complete set of experiments, every node takes a turn at broadcasting its share of packets. Since the broadcast periods do not overlap, nodes do not interfere with each other.

Each packet contains the sender's identifier and a sequence number. The transmitting node logs the transmission time and sequence number of every packet sent. Each receiving node logs the sender's identifier, sequence number, and reception time for every successfully received packet. Signal information is also reported, as provided by the 802.11 interface on a per-packet basis.

No routing protocol is running during these experiments: only experiment packets are sent or received on each node's wireless interface. The interfaces are configured to use a unique 802.11 SSID (network name); other 802.11 parameters for both testbeds are shown in Table 2.

We attempted to set the cards' maximum transmit rate to the lowest available setting, 1 Mbps, to prevent the cards 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.

Finally, using broadcast packets instead of a unicast packets avoids the 802.11 ACK and RTS/CTS mechanisms.

We performed experiments with big and small packets. Small packets were 50 bytes (8 bytes data plus UDP, IP, and Ethernet headers), approximating the size of 802.11 RTS/CTS and ACK packets. These were sent at 1024 packets per second. Big packets were 1024 bytes, more representative of large data transfers. These were sent at 50 packets per second, at an even rate. The result is a send rate of somewhat more than 400,000 bits per second, due to 802.11 headers. This should be well below the minimum 802.11 capacity of 1 megabit per second. However, on some occasions node were not able to broadcast at the desired rate, perhaps because of 802.11 traffic outside our control, or to interference appearing to the card as carrier.

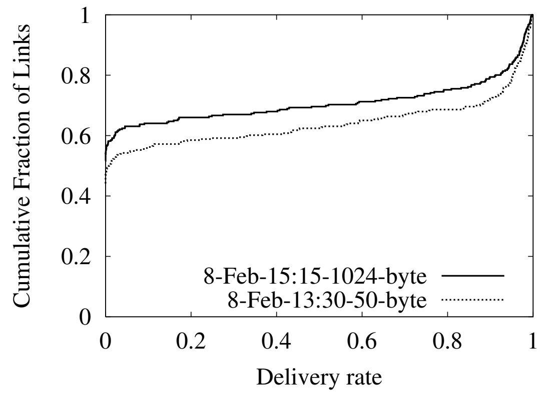

Figure 3: Cumulative distribution of per-link delivery rates on the indoor network. Note that many links are of intermediate quality.

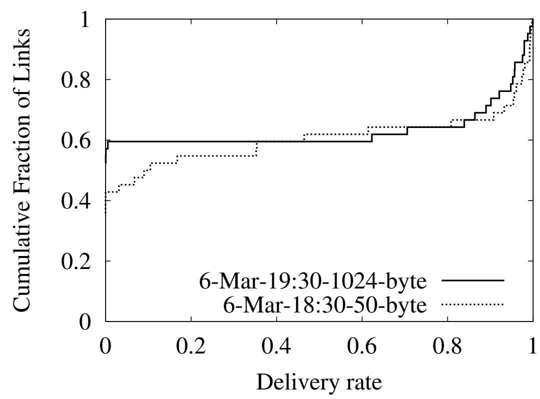

Figure 4: Cumulative distribution of per-link delivery rates on the rooftop network. Again, many links are of intermediate quality.

3.4 Results

3.4.1 Link Variation

We conducted two sets of experiments with the indoor 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 shows the cumulative distribution of delivery rates across all links for each packet size. 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 links are evenly distributed. Other experiments on different days, at different times, and with different parameters confirm that in general the links in the network exhibit a wide range of delivery rates.

We conducted identical sets of experiments on our rooftop network. Figure 4 shows the cumulative distribution of delivery rates from two of these sets of experiments, which were carried out on the evening of Wednesday 6 March 2002. Like the indoor testbed, the rooftop testbed has widely varying delivery rates for both packet sizes. Other experiments over several days exhibited the same distribution of delivery rates.

As discussed in section 3.1.1, the wide variation in delivery rates for both testbeds suggests that shortest-path routing will not work well on these networks.

3.4.2 Link Asymmetry

Figure 5 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, defined as a difference of more than 20% between the rates in each direction. The links between nodes 27 and 25 stand out: node 27 is located in the fifth floor machine room, while node 25 is to the side on the sixth floor. Similar results were obtained from experiments conducted at different times, and with large packets.

While Figure 5 suggests that there is limited value to using the good direction of asymmetric links, but noticeable value in avoiding the bad direction.

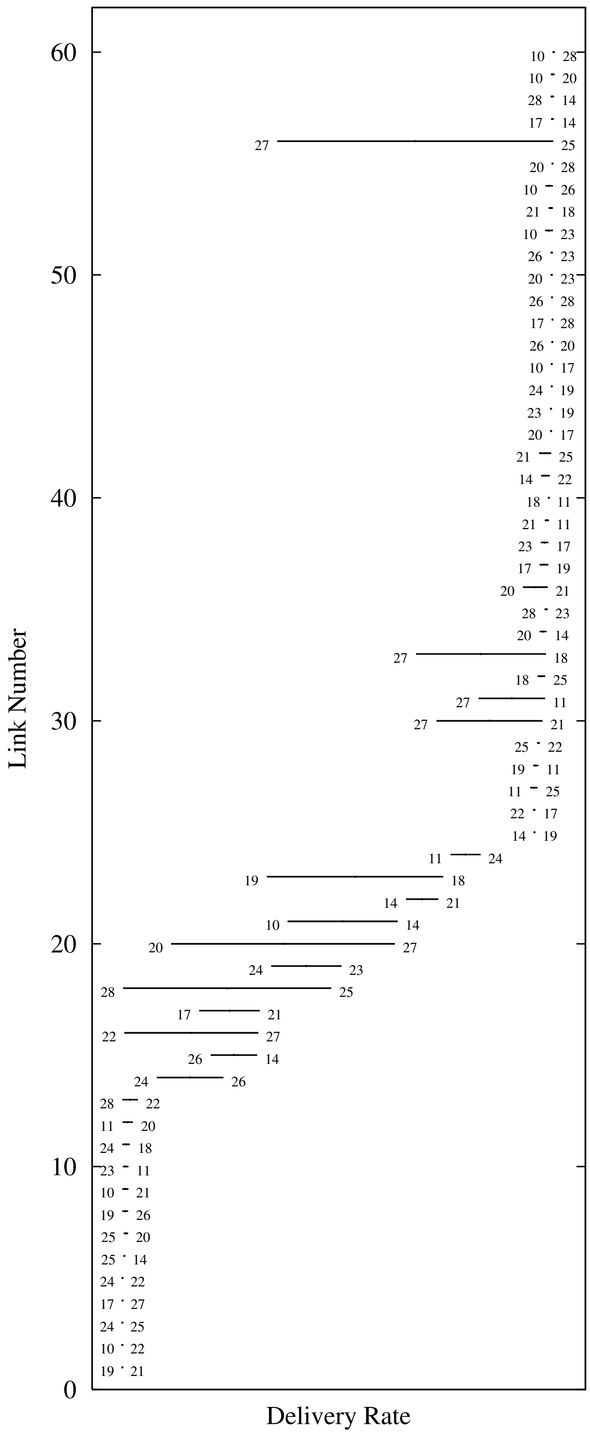

Figure 5: Delivery rates for 8-Feb-13:30-50-byte on each link pair, sorted by the larger delivery rate of each pair. The x values of the two ends of each line indicate the deliver rate in each direction; the numeric labels indicate the node IDs. Links with zero delivery rate in both directions are omitted. While most links are symmetric, a few are high quality in one direction and low quality in the other.

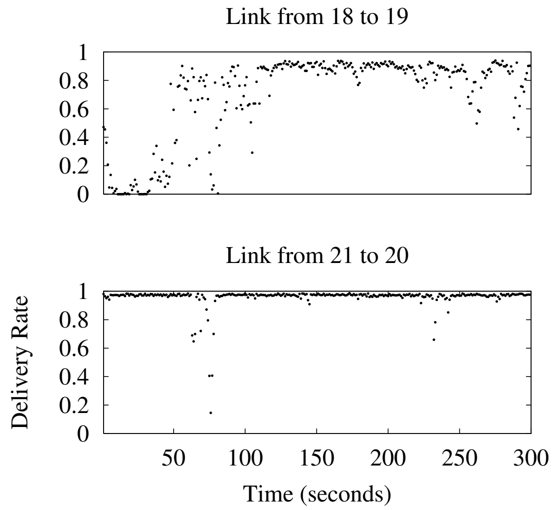

Figure 6: Example per-second variation in link delivery rates. Each point is the delivery rate over one second during 8-Feb-13:30-50-byte. The delivery rate of the 18→19 link fluctuates on a time-scale of seconds, while the 21→20 link is comparatively stable.

3.4.3 Link Variation Over Time

Figure 6 shows the second-by-second delivery rates for two links from the 8-Feb-13:30-50-byte experiment. The graphs show that while delivery rates are generally stable, they can sometimes change very quickly.

Figure 7 summarizes variation in loss rate over time for all links. For each link, we calculated the mean and standard deviation of the 1- and 10-second loss rates over the whole experiment. The graph shows the cumulative distribution of these standard deviations, normalized by the respective means. We use loss rates rather than delivery rates for this analysis because we want the graph to reflect more strongly the changes in the delivery rate on links with low loss, since very lossy links are useless for data traffic regardless of their variation.

Results for 1 and 10-second windows 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 1-second loss rate. This suggests that wireless routing protocols should use agile predictors of link loss rates.

Figure 8 shows the variation in short-term loss rates from the same experiment as in Figure 7, but carried out on the rooftop network (6-Mar-18:30-50-byte). This figure shows that short-term loss rates in the rooftop network vary nearly as much as they do in the indoor network.

A third set of experiments was performed over a 24-hour period (10-Jan-24h-1024-byte), spanning two days,

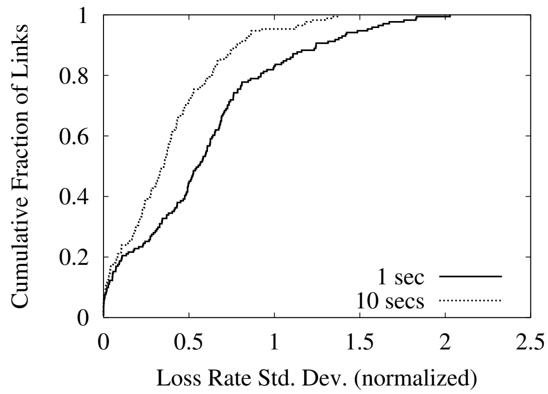

Figure 7: The cumulative distribution of the normalized standard deviation of short-term link *loss* rates calculated over 1 and 10 second intervals on the indoor network (8-Feb-13:30-50-byte). Many links show significant variation in short-term loss rates over time.

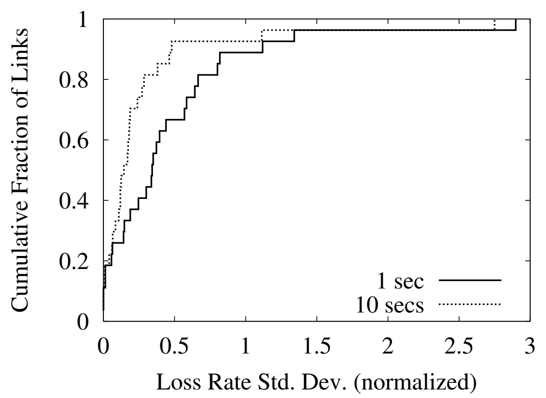

Figure 8: Cumulative distribution of the normalized standard deviation of short-term link loss rates calculated over 1 and 10 second intervals on the rooftop network. (6-Mar-18:30-50-byte).

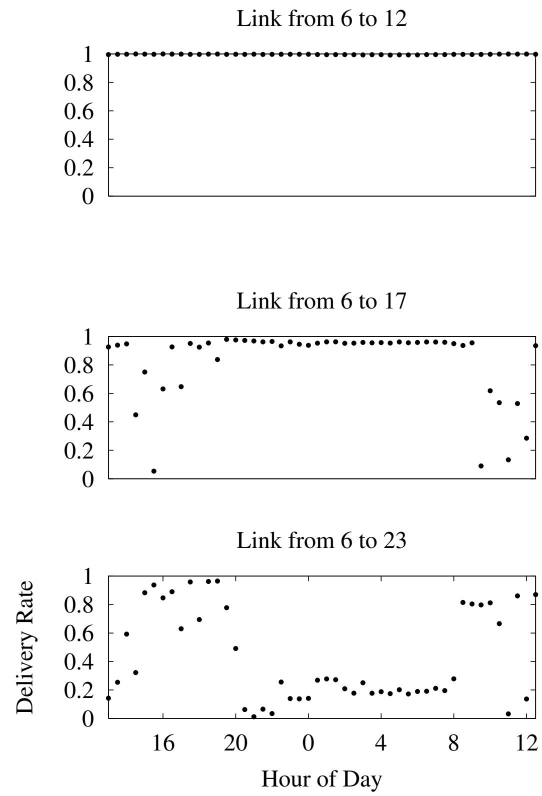

Figure 9: Example variations in link delivery rates during the day, from 10-Jan-24h-1024-byte. Each point is a different experiment. The 6→17 and 6→23 links show strong day/night and hour-to-hour variation, while the 6→12 link does not.

to examine the variation in link performance throughout the day. Each experiment was 30 minutes long, during which each node attempted to broadcast 100 1024-byte packets per second for 30 seconds. As expected, most links showed daily variations; some example link delivery rates are shown in Figure 9. Some links became worse, perhaps because more people were around: more interfering radios and appliances were in use, and human bodies attenuate signals in the 802.11 spectrum. Surprisingly, some links became better during the day. We conjecture that this is because office doors are open during the day, improving radio propagation for some links.

A similar set of 24-hour experiments was also carried out on the rooftop network. As in the case of the indoor network, the performance of some links varied over the course of the day, although to a somewhat lesser extent than in the indoor network. This is most likely because the physical obstructions at rooftop heights are unlikely to change significantly over the course of a 24 hour period. The variation that we did observe in the rooftop network over the course of a day were likely due to changing patterns of RF interference in the area.

4 Simulations

To explore the effects of the observed link characteristics on the performance of existing network protocols, we evaluated DSDV and DSR on a simulated network with loss characteristics similar to our indoor 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 [9] with the CMU wireless extensions [11]. 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 the indoor testbed. To model loss rates, we replaced the *ns* radio propagation model's power calculations with the 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 randomly assigned a received power level of 0 at each node with a probability corresponding to the observed loss rate from the sender to that node. Otherwise the packet is assigned a received power level well above the power level required for successful packet reception.

Because we used the loss rates to assign power levels, they also determine interference and carrier sense between nodes. The original propagation 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 2 Mbps CBR traffic with 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 delivery rate of the 2 Mbps CBR 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.

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. 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 10 and 11. 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 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 better than 50% of the available one-hop throughput, and a three-hop path can do no better than 33%.

4.2.1 DSDV

The DSDV results in Figure 10 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 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

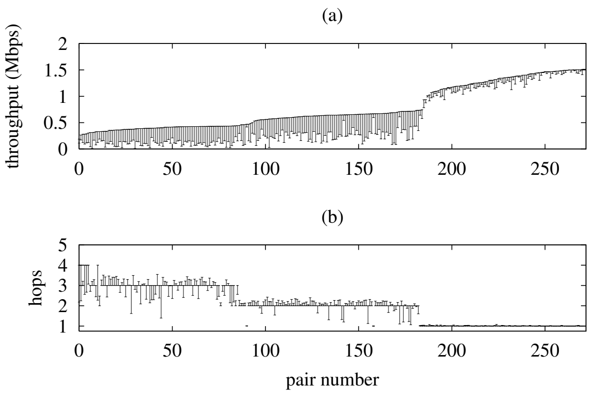

Figure 10: (a) A comparison between the end-to-end throughput made available using DSDV (marked by the bottom of each vertical line) and the estimated "best" route. One line is plotted for each pair of nodes. (b) The corresponding average route length used by successfully delivered packets for each of the pairs in (a).

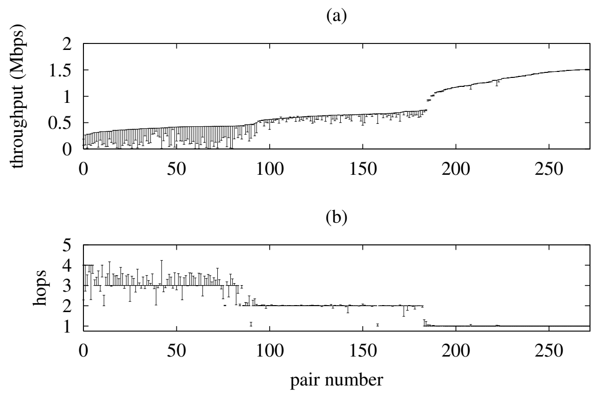

Figure 11: A comparison similar to the one shown in Figure 10, but for DSR: (a) shows end-to-end throughput made available by DSR for each pair of nodes, compared to "best". (b) shows the corresponding average route length used for successfully delivered packets.

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 11 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 contrast to DSDV, DSR generally uses routes longer than the optimal. This is a consequence of the "route repair" mechanism in the protocol, which operates when 802.11 signals that the next hop has repeatedly failed to ACK a packet. 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.

In simulations in which the source and receiver nodes are far apart, there are more potential routes between them, and it becomes less likely that the sender's initial route query will result in an acceptable route. When any link along that route fails to deliver a packet, an alternate, most likely longer, route will be used. DSR never changes routes except in the case of failure, so it is not surprising that most traffic flows along these longer paths.

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 work well, DSR will continue to use it. For this reason tends avoid long-term use of low quality routes, and thus performs better than DSDV.

5 Route Metrics

This section proposes routing techniques intended to handle the observed loss characteristics of wireless ad hoc networks. These techniques have been partially implemented but not yet evaluated.

The main proposal is that routing protocols use the expected *transmission count* of a packet as the route metric. The transmission count includes all transmissions of a packet, including any retransmissions. Transmission count quantifies the total impact of sending a packet on system resources. That is, it measures the square-meter-seconds of spectrum consumed by sending a packet along a particular route. If a multihop wireless network uses routes with lower transmission counts, it can support more users. Transmission count is also desirable because it penalizes routes with more hops, and routes with poor links.

The expected transmission count of a packet along a route is the sum of the expected transmission counts of the packet on each of the route's links. Transmission count is an additive metric, and routing protocols should prefer paths with lower transmission counts.

Transmission count is calculated from the estimated broadcast delivery rate of each link. Assuming that unicast transmissions use link-layer ACKs, then a link with a broadcast delivery rate of r_f in the forward direction, and a rate of r_r in the reverse direction has an expected transmission count $1/(r_f \times r_r)$.[1] The rates r_f and r_r may be functions of the packet size.

Before we can use transmission count as a metric, we must determine the forward and reverse rates r_f and r_r of each link. We can do this in two main ways: we can observe the actual delivery rates on the link, or we can derive delivery rates from signal measurements provided by the 802.11 hardware.

5.1 Measured Delivery Rates

We can measure the actual broadcast delivery rate of a link by observing the number of broadcasts we hear from each sender during a specified time interval. The measured delivery rate is then used as an estimate of the future delivery rate. This is not a new idea, and was described in [14, 7], for example. However, the details are important. Each node's broadcasts (e.g. routing updates in DSDV) contain a count of the number of broadcasts by that node. Each node also counts the number n of routing broadcasts received from each neighbor over a rolling time window. When a node receives a broadcast, it can determine how many broadcasts it expects to have received from that neighbor by looking at the difference δ between the smallest and largest broadcast counts received during the window. We can then calculate the broadcast delivery rate from the neighbor conservatively as

$$r = \frac{n - 0.5}{\delta + 1}$$

One limitation of this technique is that it requires multiple packets during a window to estimate the delivery rate. If only one packet is received during a window, the number of broadcasts expected cannot be computed. This can be an issue, for example, when a new neighbor comes into range using a protocol such as DSDV. A node cannot calculate the broadcast delivery rate from the new neighbor until the neighbor has sent at least two route updates. Another drawback of this direct observation technique is that it is sensitive to the window size. Longer windows provide more precision but are less responsive to changes in the delivery rate. This is a problem when links change delivery rates quickly, such as in our testbeds.

5.2 Predicted Delivery Rates

Given that directly measured delivery rates may not be responsive enough to accurately measure a link, we would like a more predictive measurement. Most 802.11 adapters provide some information about the signal on each link. The Aironet 340 and 350 adapters can return two signal measurements to the routing software with every received packet: *signal strength* and *signal quality*. Signal strength is recorded in dBm, and measures the total amount of radio energy experienced by the wireless interface, including noise.[2] Unfortunately the quality number is not defined by the Aironet programming manual [1]. However, the Aironets use the Intersil Prism II chipset [4], whose documentation [3] defines quality to be a measure of the signal-to-noise ratio.

To determine if these signal measurements are useful for characterizing links, we examined the relationship between short-term delivery rates and the signal strength and quality reported by the Aironet 340s on our indoor testbed. Figure 12 shows these relationships for the 8-Feb-13:30-50-byte experiment, over 1 second intervals. Each point plots the delivery rate over one second of a particular link, along with average signal strength or quality of the link's sender, as measured by the link's receiver.

Unfortunately, there is no precise correlation between delivery rates and either signal strength or signal quality. Figure 13 shows data broken out by two example links over time. For some links, such as $18 \rightarrow 19$, signal strength does not vary appreciably, even as the delivery rate changes markedly. We calculated a linear regression of delivery rate again both signal strength and signal quality, for each link individually. We discovered that every link had different parameters for these regressions, and that no parameters would suit all links. Indeed, the delivery rates of some links hardly depended on the signal strength or quality at all.

[1] Each attempted transmission is one trial in a Bernoulli process. A successful trial is one in which the packet is successfully received by the recipient, *and* the ACK is successfully received by the sender.

[2] Signal strength is derived from a quantity known as RSSI (received signal strength indication), using a table in the firmware of each adapter. This table approximates the function $\mathrm{dBm} = 100 - \mathrm{RSSI}$. RSSI is described by the 802.11 specification [6] as "a measure of the RF energy received..." (Section 15.4.5.10.2).

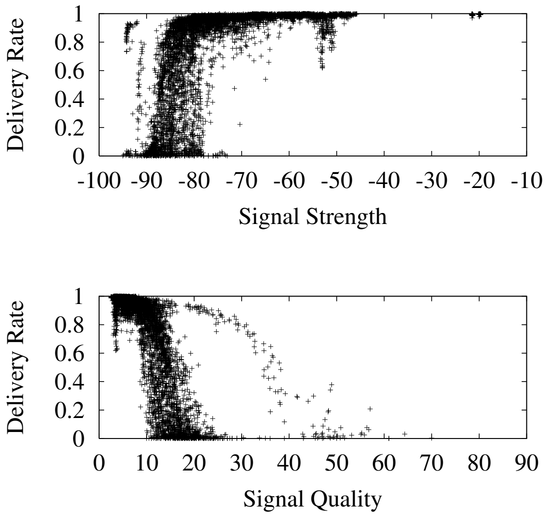

Figure 12: Delivery rate versus quality and signal strength as reported by the Aironet card. Each point is the delivery rate over 1 second for some link, plotted against the link's average signal strength or quality during that interval, from 8-Feb-13:30-50-byte. There is no simple relationship between signal strength and delivery rate; quality is a slightly better predictor.

Looking at the aggregate data though, high signal strength and low quality values do correlate with high delivery rates. One approach would be to simply threshold links based on their signal strength or quality. For example, a link whose signal strength is above -75 dBm could be assigned a delivery rate of 0.75 or 0.8; a link whose signal quality is less than 10 could be assigned a delivery rate of 0.7.

These are very coarse-grained predictors. A more sophisticated approach would be to combine the directly measured broadcast rates with a separate predictor for each link. The predictor would track the delivery rate history and signal strength and quality numbers to find the best relationship between the signal measurements and the link's delivery rate.

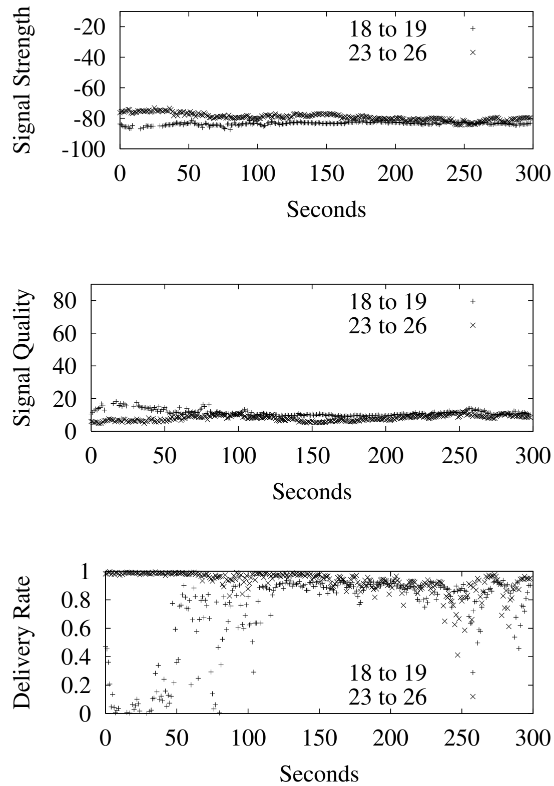

Figure 13: Average signal strength, average signal quality, and delivery rate for example links, over 1 second intervals, from 8-Feb-13:30-50-byte.

5.3 Managing Measurements

The previous subsections describe how to estimate the broadcast delivery rate of a link on the receiver's side. This provides each node with r_r, the reverse delivery rate of each link. For nodes to learn r_f, the forward delivery rate on the link, link receivers must send link measurements back to the link senders.

Link measurements values can be "ping-ponged" back to the link sender in the header of every unicast packet sent back to that neighbor. Most links will experience some two-way communication; for example, TCP sends ACKs back to the TCP sender. However, for links that aren't used, the reverse measurements for every link can be included in routing advertisements. For on-demand protocols that don't have periodic broadcasts, a dedicated packet exchange must occur for nodes to properly estimate link delivery rates.

6 Related Work

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

In [12], 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 [12] show that this metric provides better results that just using shortest paths.

In *preemptive routing* [10], 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 [10] show that this technique decreases the number of broken routes, and generally decreases latency.

Signal stability-based adaptive (SSA) routing [8] 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, [8] reports that this is not effective in reducing broken routes.

The CMU Monarch Project constructed a testbed network [17, 16] running DSR [13]. Their implementation included a quality metric [20] 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 detect when links are about to break. They all assume that link quality is a simple threshold function of signal strength (except for [12], which assumes quality is a function of distance). However, our experience with a static indoor network shows that signal strength remains relatively constant for a given link over a given distance, while delivery rates can vary considerably. This is consistent with the actual behavior of radios: the successful reception of a packet actually depends on the signal to noise ratio at the receiver, along with receiver's sensitivity. Moreover, our measurements of signal strength and cooresponding loss rates indicate that in practice there is no single relationship between the two values. This would make implementation of any of the above techniques difficult on our testbeds.

The DARPA packet radio network (PRNET) [14] 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 [7] 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.

7 Conclusions

This contribution of this paper is to show how lossy radio links can severely impact ad hoc routing protocol performance. We provided an analysis of link delivery rates using measured results from two wireless testbed networks, which illustrated three important points about real link behavior:

1. Link performance is not bimodal; links can have many delivery rates. This means that routing protocols *must* account for individual link performance when choosing routes.

2. Asymmetric links are not uncommon. This violates the assumptions of many proposed ad hoc protocols.

3. Link delivery rates can change quickly, so routing protocols should have good predictors of link performance.

Simulations using the measured delivery rates showed how lossy links reduce the performance of the DSR and DSDV protocols.

We presented a new route metric, the expected *transmission count* of a route, which favors routes with higher throughput, fewer links, and less spectrum consumption. We also discussed how to obtain this metric for each link, either using directly measured broadcast delivery rates, or by using signal indicators from the radio adapter to predict loss rates.

References

[1] *PC4500/PC4800 Developer's Reference Manual*. Aironet Wireless Communications, 1997. Document number: 710-004247, Revision B1.

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

[3] *HFA3863 Direct Sequence Spread Spectrum Baseband Processor with Rake Receiver and Equalizer Data Sheet*. Intersil Americas Inc., December 2001.

[4] *Quick Reference Guide, Cisco Aironet 340 Series Products*. Cisco Systems Inc., February 2002. `http://www.cisco.com/warp/public/cc/pd/witc/ao340ap/prodlit/aiqrg_rg.htm`.

[5] Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan, and Randy H. Katz. A comparison of mechanisms for improving TCP performance of wireless links. *IEEE/ACM Transactions on Networking*, 6(5), December 1997.

[6] 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.

[7] Brian H. Davies and T. R. Davies. The application of packet switching techniques to combat net radio. *Proceedings of the IEEE*, 75(1), January 1987.

[8] 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.

[9] Kevin Fall and Kannan Varadhan. *ns* notes and documentation. Technical report, UC Berkeley, LBL, USC/ISI, and Xerox PARC, November 1997. `http://www-mash.berkeley.edu/ns`.

[10] 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.

[11] CMU Monarch Group. CMU Monarch extensions to *ns*. `http://www.monarch.cs.cmu.edu/`.

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

[13] 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.

[14] John Jubin and Janet D. Tornow. The DARPA packet radio network protocols. *Proceedings of the IEEE*, 75(1), January 1987.

[15] 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.

[16] 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.

[17] 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.

[18] Christina Parsa and J. J. Garcia-Luna-Aceves. TULIP: A link-level protocol for improving TCP over wireless links. In *Proc. IEEE Wireless Communications and Networking Conference 1999 (WCNC 99)*, September 1999.

[19] 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. `http://www.`

ietf.org/internet-drafts/draft-ietf-manet-aodv-04.txt.

[20] 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.