Architecture and Evaluation of an Unplanned 802.11b Mesh Network

John Bicket, Daniel Aguayo, Sanjit Biswas, Robert Morris
M.I.T. Computer Science and Artificial Intelligence Laboratory
jbicket, aguayo, biswas, rtm @csail.mit.edu

ABSTRACT

This paper evaluates the ability of a wireless mesh architecture to provide high performance Internet access while demanding little deployment planning or operational management. The architecture considered in this paper has unplanned node placement (rather than planned topology), omni-directional antennas (rather than directional links), and multi-hop routing (rather than single-hop base stations). These design decisions contribute to ease of deployment, an important requirement for community wireless networks. However, this architecture carries the risk that lack of planning might render the network's performance unusably low. For example, it might be necessary to place nodes carefully to ensure connectivity; the omni-directional antennas might provide uselessly short radio ranges; or the inefficiency of multi-hop forwarding might leave some users effectively disconnected.

The paper evaluates this unplanned mesh architecture with a case study of the Roofnet 802.11b mesh network. Roofnet consists of 37 nodes spread over four square kilometers of an urban area. The network provides users with usable performance despite lack of planning: the average inter-node throughput is 627 kbits/second, even though the average route has three hops.

The paper evaluates multiple aspects of the architecture: the effect of node density on connectivity and throughput; the characteristics of the links that the routing protocol elects to use; the usefulness of the highly connected mesh afforded by omni-directional antennas for robustness and throughput; and the potential performance of a single-hop network using the same nodes as Roofnet.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing protocols*

General Terms

Design, Experimentation, Measurement, Performance

Keywords

Mesh networks, Multi-hop wireless networks, Ad hoc networks, Wireless routing, Route metrics

1. INTRODUCTION

Community wireless networks typically share a few wired Internet connections among many users spread over an urban area. Two approaches to constructing community networks are common. The first approach is to carefully construct a multi-hop network with nodes in chosen locations and directional antennas aimed to engineer high-quality radio links [31, 8, 29]; these networks require well-coordinated groups with technical expertise, but result in high throughput and good connectivity. The second approach consists of individuals operating "hot-spot" access points to which clients directly connect [5, 4]. These access points often operate independently and are loosely connected, if at all. Access-point networks do not require much coordination to deploy and operate, but usually do not provide as much coverage per wired connection as multi-hop networks.

A more ambitious vision for community networks would combine the best characteristics of both network types, operating without extensive planning or central management but still providing wide coverage and acceptable performance. This paper provides an evaluation of such an architecture, consisting of the following design decisions:

1. Unconstrained node placement, rather than a topology planned for coverage or performance. The network should work well even if the topology is determined solely by where participants happen to live.

2. Omni-directional antennas, rather than directional antennas used to form particular high-quality links. Users should be able to install an antenna without knowing in advance what nodes the antenna might talk to. Nodes should be able to route data through whatever neighbors they happen to find.

3. Multi-hop routing, rather than single-hop base stations or access points. Multi-hop routing can improve coverage and performance despite lack of planning and lack of specifically engineered links.

4. Optimization of routing for throughput in a slowly-changing network with many links of intermediate quality, rather than for route repair in a mobile network.

These decisions ease deployment, but they also risk reduced network performance. For example, radio ranges might be too short to connect some nodes; many links might be low quality; nodes might interfere with each other and cause persistent packet loss; standard TCP might interact poorly with low-quality radio links; or the outdoor omni-directional antennas might pick up unacceptable levels of interference from other ISM-band users throughout the city.

While none of the individual elements of the unplanned mesh architecture outlined above are new [1, 3, 2, 6], there has been no systematic evaluation of whether the architecture as a whole can achieve its goals. Prior evaluations of similar mesh networks have focused on routing metrics [14] and the radio-level causes of packet loss [7]. MANETs have been extensively studied [10, 19, 12], but their routing protocol design is focused on coping with rapid topology changes due to mobility.

This paper evaluates the unplanned mesh architecture with a case study of Roofnet. Roofnet is a multi-hop 802.11b Internet access network consisting of 37 nodes spread over about four square kilometers of a city. While previous work investigated the physical-layer causes of packet loss in Roofnet [7], this paper describes Roofnet's end-to-end characteristics.

The paper's conclusions are as follows. The unplanned mesh architecture of Roofnet works: with few exceptions, users see throughput and delay comparable to DSL links and the average throughput between nodes is 627 kbits/second. Throughput decreases with number of hops, but even eight-hop routes average 160 kbits/second. Single-flow throughput increases with node density, since the routing protocol makes use of short links and drives them at high bit-rates. The majority of the radio links are between 500 and 1300 meters long, though the links that prove the most useful to the routing protocol are a relatively small number of short high-throughput links. While Roofnet includes a few nodes positioned on the tops of tall buildings, its performance and robustness do not greatly depend on any small set of nodes; it is a true mesh. Finally, while a single-hop base-station architecture would have been possible for Roofnet, for any given number of wired access points, Roofnet's multi-hop forwarding improves coverage and throughput.

Some areas that this paper does *not* investigate include the effect of multiple concurrent flows, network and protocol scalability, the detailed design of routing protocols, change in network performance over time, and the long-term evolution of the network as users join and depart.

The next section outlines Roofnet's design. Section 3 is the core of the paper, presenting measurements and discussing what they imply about the unplanned mesh architecture. Section 4 presents a snapshot of user traffic on Roofnet. Section 5 reviews related work, and Section 6 concludes.

2. ROOFNET DESIGN

Roofnet is deployed over an area of about four square kilometers in Cambridge, Massachusetts. Figure 1 shows a map of the network. This area is urban and densely populated. Most buildings are three- or four-story apartment buildings;

Figure 1: A map of Roofnet. Each dot represents a wireless node. The map covers the south-east portion of Cambridge, Massachusetts. The Charles River curves along the lower boundary of the map, MIT is at the lower right, and Harvard is at the upper left.

most Roofnet nodes are located in such buildings, but eight are in taller buildings. Line-of-sight signal propagation between nodes is often obstructed.

Each Roofnet node is hosted by a volunteer user. Volunteers were solicited by a combination of word of mouth, posters distributed in and near the network's coverage area, and links to a Web sign-up page. Each volunteer installed his or her own node, including the roof-mounted antenna. The resulting node locations are neither truly random nor selected according to any particular plan.

2.1 Hardware

Each Roofnet node consists of a PC, an 802.11b card, and a roof-mounted omni-directional antenna. The PC is small and quiet so that it is suitable for residential use. The PC's Ethernet port provides Internet service to the user. Each PC has a hard drive for collecting traces and a CD reader in case an over-the-network upgrade fails. An entire Roofnet kit (PC, antenna, mounting hardware, and cable) can be carried by one person.

Each 8 dBi omni-directional antenna has a 3-dB vertical beam width of 20 degrees. The wide beam sacrifices gain but means the antenna need not be perfectly vertical. The antenna is connected to its node with coaxial cable which introduces 6 to 10 dB of attenuation. Three nodes, located on the roofs of tall buildings, have 12 dBi Yagi directional antennas with 45-degree horizontal and vertical beam widths.

Each node is equipped with an 802.11b wireless card based on the Intersil Prism 2.5 chip-set. The radios transmit 200 milliwatts of power, operate with RTS/CTS disabled, and all share the same 802.11b channel. The cards use a non-standard "pseudo-IBSS" mode that is similar to standard 802.11b IBSS (or "ad hoc") mode, in that nodes communicate directly without access points. Pseudo-IBSS omits 802.11's beacons and BSSID (network ID) mechanism, solv-

ing IBSS mode's tendency to form partitions which have different BSSIDs despite having the same network ID. These partitions made it impossible to operate Roofnet reliably with IBSS mode.

2.2 Software and Auto-Configuration

Each Roofnet node runs identical turn-key software consisting of Linux, routing software implemented in Click [22], a DHCP server, and a web server so users can monitor the network status. Most users pick up nodes from us at our lab with software pre-installed. We regularly upgrade all the nodes' software over Roofnet, and occasionally by mailing out installation CDs.

From the user's perspective, the node acts like a cable or DSL modem: the user connects a PC or laptop to the node's Ethernet interface, and the node automatically configures the user's computer via DHCP, listing the node itself as the default IP router. Some users choose to connect the node to their own wireless access point.

In order that Roofnet nodes be completely self-configuring, the software must automatically solve a number of problems: allocating addresses, finding a gateway between Roofnet and the Internet, and choosing a good multi-hop route to that gateway.

2.2.1 Addressing

Roofnet carries IP packets inside its own header format and routing protocol. Each Roofnet node needs a unique address at the Roofnet layer, as well as an IP address so that IP applications work between Roofnet nodes. The Roofnet software running on a node must be able to assign itself addresses automatically, without requiring explicit configuration. It does this by choosing an address whose low 24 bits are the low 24 bits of the node's Ethernet address, and whose high 8 bits are an unused class-A IP address block. The node uses the same address at both the Roofnet and IP layers. These addresses are meaningful only inside Roofnet; they are not globally routable.

A Roofnet node must also allocate IP addresses via DHCP to user hosts attached to the node's Ethernet port. Each node allocates these addresses from the reserved 192.168.1.x IP address block. The node uses NAT between the Ethernet and Roofnet, so that connections from a user's host appear to the rest of Roofnet to have originated from the user's node. This use of NAT allows nodes to allocate IP addresses to users' hosts independently, but prevents these hosts from connecting to each other through Roofnet.

2.2.2 Gateways and Internet Access

Roofnet's design assumes that a small fraction of Roofnet users will voluntarily share their wired Internet access links. Multiple consumer DSL ISPs in the Roofnet area (e.g. Speakeasy and Cyberion) have Acceptable Use Policies that allow wireless sharing of Internet connections, so there is no contractual difficulty.

On start-up, each Roofnet node checks to see if it can reach the Internet through its Ethernet port: it asks for an IP address as a DHCP client, and then tries to connect to some well-known Internet servers. If this succeeds, the node advertises itself to Roofnet as an Internet gateway. Otherwise the node acts as a DHCP server and default router for hosts on its Ethernet, and connects to the Internet via Roofnet.

Each gateway acts as a NAT for connections from Roofnet to the Internet, rewriting the source address of packets coming from Roofnet with the IP address of the gateway's Ethernet interface.

When a node sends traffic through Roofnet to the Internet, the node selects the gateway to which it has the best route metric. The node keeps track of which gateway is being used for each open TCP connection, because the gateway's use of NAT requires each connection to use a single gateway. If the routing protocol later decides that a different gateway has the best metric, the node continues to forward data on existing TCP connections to those connections' original gateways.

Each new TCP connection uses the gateway with the best metric when the connection starts. If a Roofnet gateway fails, existing TCP connections through that gateway will fail (because of the NAT), but new connections will use a different gateway.

Roofnet currently has four Internet gateways. Two are located in ordinary residences, one is on the roof of a six-story university building, and the last is in a ninth-story window of another university building.

2.3 Routing Protocol

Roofnet's routing protocol, Srcr, tries to find the highest-throughput route between any pair of Roofnet nodes. Omni-directional antennas give Srcr many choices of links it could use, but since most links are low-quality, Srcr must evaluate each link's usable throughput. Srcr's design is motivated by recent measurement studies of real-world wireless behavior [25, 18, 32, 13, 7, 14].

Srcr source-routes data packets, like DSR [20], in order to avoid routing loops when link metrics change. Each Srcr node maintains a partial database of link metrics between other pairs of nodes (see Section 2.4), and uses Dijkstra's algorithm on that database to find routes. Nodes learn fresh link metrics in three ways. A node that forwards a packet over a link includes the link's current metric in the packet's source route, so that other nodes on the route can see the metric. If a node needs to originate a packet but cannot find a route with its current database contents, it sends a DSR-style flooded query and adds the link metrics learned from any responses to its database. Finally, nodes that overhear queries and responses add the metrics in those packets to their databases.

This combination of link-state and DSR-style on demand querying was inspired by MCL [14]. It is particularly efficient when most nodes talk only to a gateway. Each Roofnet gateway periodically floods a dummy query that allows all other nodes to learn about links on the way to that gateway. When a node sends data to a gateway, the gateway will learn (from the source routed data packets) about links back to the node. Thus in ordinary operation nodes do not ever need to send flooded queries.

One problem with the above scheme is that flooded queries often do not follow the best route [18]. Srcr addresses this problem as follows. When a node receives a new query it first adds the link metric information in the query's source route to its link database. Then the node computes the best route from the query's source, and replaces the query's source route with that best route. Finally the node re-broadcasts the query. If the node later receives another copy of the query and is able to compute a best route with a lower

metric, it will forward the query again with the new best route. In practice, most nodes forward a query only once or twice. A larger or denser network than Roofnet in which nodes send data to many destinations might require a more scalable query mechanism.

Link conditions may change so that an active route is no longer the best route. If a link on an active route stops forwarding packets altogether, the upstream node will send a notification back to each failed packet's source. If a link on an active route merely degrades, sources using that link will learn of the link's new metric from forwarded data packets. A source re-runs Dijkstra's algorithm on its database whenever it learns of a changed link metric, so the source will switch routes if it knows about a better one. If a link changes to have a better metric, sources that might want to use that link can learn about it either through dummy queries from gateways, or by a mechanism in which nodes forwarding data packets include unsolicited link metric information about nearby links.

2.4 Routing Metric

Srcr chooses routes with an "estimated transmission time" (ETT) metric, derived from estimated transmission count (ETX) [13]. ETT predicts the total amount of time it would take to send a data packet along a route, taking into account each link's highest-throughput transmit bit-rate and its delivery probability at that bit-rate. Srcr chooses the route with the lowest ETT, since that route is likely to be able to deliver the most packets per unit time.

Each Roofnet node sends periodic 1500-byte broadcasts at each available 802.11 bit-rate, and periodic minimum-size broadcasts at one megabit/second. Nodes keep track of the fraction of these broadcasts that they receive from each neighbor, and report the statistics back to each neighbor. Srcr predicts that a link's highest-throughput bit-rate is the bit-rate with the highest product of delivery probability and bit-rate. Delivery probability at a bit-rate is the product of the fraction of 1500-byte broadcasts delivered and the fraction of 60-byte one megabit/second broadcasts delivered in the reverse direction, to account for lost 802.11 ACKs.

The ETT metric for a given link is the expected time to successfully send a 1500-byte packet at that link's highest-throughput bit-rate, including the time for the number of retransmissions predicted by the measured delivery probabilities. The ETT metric for a route is the sum of the ETTs for each of the route's links. That is, Srcr assumes the following relation between the throughputs t_i of a route's hops and the route's end-to-end throughput t:

$$t = \frac{1}{\sum_i \frac{1}{t_i}} \qquad (1)$$

This relation is reasonably accurate for short routes where at most one hop can send at a time [24]. It underestimates throughput for longer routes in which distant hops can send concurrently without interfering, and overestimates throughput when transmissions on different hops collide and are lost. Section 3.7 examines the accuracy of Equation 1.

ETT is similar to WCETT [15]. ETT uses broadcast probes to predict transmission time, while WCETT directly measures ETT using unicast packets between each pair of nodes. ETT is likely to be less accurate than WCETT, but has lower measurement overhead.

2.5 Bit-Rate Selection

Roofnet has its own algorithm to choose among the 802.11b transmit bit-rates of 1, 2, 5.5, and 11 megabits/second. The algorithm included in the Prism firmware tends to avoid using bit-rates with significant loss-rates, a policy which performs badly on Roofnet. The highest-throughput routes often involve links with relatively high link-level loss rates, since a single-hop route with 40% loss can deliver more data than a two-hop route with perfect links. In addition, because the 802.11b bit-rates are at roughly power-of-two intervals, a high bit-rate with up to 50% loss is preferable to the next-lowest bit-rate.

Roofnet's algorithm, called SampleRate [9], adjusts the bit-rate as it sends data packets over a link. Like ETT, SampleRate judges which bit-rate will provide the highest throughput based on delivery probabilities measured at the different bit-rates. Unlike ETT, SampleRate bases its decisions on actual data transmissions rather than on periodic broadcast probes. As a result SampleRate can adjust its choice more quickly and accurately than ETT.

SampleRate sends most data packets at the bit-rate it currently believes will provide the highest throughput. SampleRate periodically sends a data packet at some other bit-rate in order to update a record of each bit-rate's delivery probability. SampleRate switches to a different bit-rate if the throughput estimate based on that bit-rate's recorded delivery probability is higher than the current bit-rate's throughput.

3. EVALUATION

This section characterizes the end-to-end performance of Roofnet and the impact of some of its architectural choices.

Section 3.2 presents basic measurements of throughput and latency over the network. Over all pairs, throughput averages 627 kbits/second, and round-trip latency averages 39 milliseconds. Transfers from the Internet gateways ran at an average of 1395 kbits/second, with an average latency of 22 milliseconds.

Next the paper investigates the effects of three choices in the unplanned mesh architecture. Section 3.3 studies how Srcr makes use of the large number of links afforded by omni-directional antennas. Most links are between 500 and 1500 meters long and have throughputs of less than 500 kbits/second; however, Srcr largely ignores them and uses a relatively small number of shorter links that run at two or more megabits/second. The median link-level loss rate of the links Srcr chooses is 20%. Section 3.4 explores the effect of node density on connectivity and throughput, by evaluating different-size subsets of the Roofnet nodes. These data provide a feel for how well other mesh deployments with different node densities might work. Section 3.5 assesses how Roofnet takes advantage of a highly connected mesh. Most Roofnet nodes route through many neighbors. The mesh is robust in the sense that no small set of links contributes disproportionately to overall throughput.

Section 3.6 compares multi-hop routing with a single-hop architecture that resembles an access-point network. With only single-hop forwarding, five well-placed gateways would be required to cover all Roofnet nodes, and even more would be required to match the average throughput that multi-hop Roofnet provides.

Finally, Section 3.7 presents measurements suggesting that inter-hop collisions are a major limiting factor in multi-hop throughput.

3.1 Method

The results presented here are derived from four sets of measurements on Roofnet:

1. The "multi-hop TCP" data-set is the result of a 15-second one-way bulk TCP transfer between each pair of Roofnet nodes. Throughput is measured by the number of data bytes delivered to the receiving application. Each transfer is preceded by a 30-second quiet interval and then ten seconds in which the sender sends 84-byte pings once per second to establish the route and measure latency. The throughput result and the median round-trip ping time are recorded along with the route in use at the end of the transfer. The measurements are taken with RTS/CTS disabled; measurements with RTS/CTS enabled are presented in Section 3.7.

   About 10% of pairs failed to find a working route in the multi-hop TCP measurements. The reason for this is that flooded routing queries sometimes do not reach distant nodes due to link losses: the query packets are broadcast without link-layer retransmission. Srcr re-floods every five seconds if needed, but in many cases even this was not enough. The failed pairs are included in the throughput results with throughputs of zero.

2. The "single-hop TCP" data-set is the result of measuring the TCP throughput on the direct radio link between each pair of nodes.

3. The "loss matrix" data-set is the result of measuring the loss rate between each pair of nodes using 1500-byte broadcasts at each 802.11b transmit bit-rate. These loss rates reflect underlying link losses, since 802.11 does not retransmit broadcast packets.

4. The "multi-hop density" data-set measures multi-hop TCP throughput between a fixed set of four nodes, while varying the number of Roofnet nodes that are participating in routing.

Some of the analyses involve simulated route throughput calculated from the single-hop TCP data-set and Equation 1. This technique allows us to estimate the throughput of paths that were not used by the all-pairs TCP measurements.

Each graph's caption ends with the data-set from which the graph was derived, in parentheses, and says "Simulated" when appropriate.

Roofnet ordinarily provides Internet access to its users, and this service was not disabled during these experiments. Thus the experiments may be affected by Roofnet traffic as well as other uses of the 2.4 gigahertz ISM band.

Most of the results presented below include all pairs of Roofnet nodes, rather than just communication with the gateways. This is because the Internet performance Roofnet users see is highly dependent on the specifics of Roofnet's Internet gateways, which in a community network would probably be randomly placed.
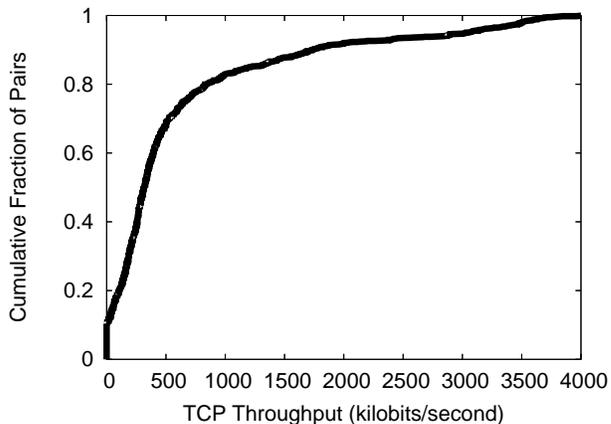

Figure 2: CDF of the TCP throughput between each pair of nodes in the network. (multi-hop TCP)

Hops	Number of Pairs	Throughput (kbits/sec)	Latency (ms)
1	158	2451	14
2	303	771	26
3	301	362	45
4	223	266	50
5	120	210	60
6	43	272	100
7	33	181	83
8	14	159	119
9	4	175	182
10	1	182	218
no route	132	0	–
Avg: 2.9	Total: 1332	Avg: 627	Avg: 39

Table 1: Average TCP throughput and round-trip ping latency between each pair in the network, arranged by the number of hops in a typical path between the pair. (multi-hop TCP)

3.2 Basic Performance

Figure 2 shows the distribution of TCP throughputs among all pairs of Roofnet nodes. The median is 400 kbits/second, and the average is 627.

The distribution of throughputs is explained largely by hop-count, as successive nodes in a multi-hop route usually must defer to each others' transmissions. Table 1 arranges the throughput data by hop-count in order to illustrate this point. The routes with low hop-count have much higher throughput than those with many hops.

For comparison, Table 2 shows the theoretical maximum throughput for various bit-rates over multiple lossless hops. This table is computed with Equation 1. Per-link throughputs include transmission time, inter-frame gaps, and link-level acknowledgments.

Tables 1 and 2 suggest that Roofnet's one-hop routes operate at an average speed consistent with the 5.5 megabit transmission rate, but that longer routes are disproportionately slower. Section 3.7 suggests that multi-hop routes suffer from inter-hop collisions and have lower performance than predicted by Equation 1.

	Max Throughput (kbits/sec)		
Rate	1 Hop	2 Hops	3 Hops
1	890	445	297
2	1634	817	545
5.5	3435	1718	1145
11	5013	2506	1671

Table 2: Theoretical loss-free maximum throughput over one, two, and three hops for each 802.11b transmit bit-rate, with 1500-byte packets.

Hops	Number of nodes	Throughput (kbits/sec)	Latency (ms)
1	12	2752	9
2	8	940	19
3	5	552	27
4	7	379	43
5	1	89	37
Avg: 2.3	Total: 33	Avg: 1395	Avg: 22

Table 3: Average TCP throughput and round-trip ping latency to the 33 non-gateway nodes from each node's chosen gateway, arranged by hop-count. Even at four hops, the average throughput is comparable to many DSL links. (multi-hop TCP)

Most Roofnet users talk only to the Internet gateway with the best metric, and thus use routes with fewer hops than the average of the all-pairs routes. Table 3 shows the TCP throughput to each node from its chosen gateway, again arranged by hop-count. The maximum hop-count is only five because no node is very far from the nearest gateway. The average throughput for each hop-count is typically higher because the Roofnet gateways happen to be more centrally located than the average Roofnet node. Even at four hops, the average of 379 kbits/second is comparable to many DSL links.

The tables also show round-trip latencies for 84-byte ping packets to estimate interactive delay on a relatively idle network. Latency is affected by per-hop processing time as well as by 802.11 retransmissions and back-offs when packets are lost. Tables 1 and 3 suggest that interactive latency is acceptable over a few hops but would be bothersome over nine hops. Roofnet users see on average 22 ms of latency to the gateways, which is hardly noticeable in an interactive session.

3.3 Link Quality and Distance

While high quality 802.11 links can be constructed using directional antennas, it is not clear what useful ranges and speeds to expect with omni-directional antennas, or what kinds of links will be most useful to the routing protocol.

The upper graph in Figure 3 shows the throughput and distance of each available link. Most of the available links are between 500 and 1300 meters long, and can transfer about 500 kbits/second at their best bit-rate. There are also a small number of links a few hundred meters long with
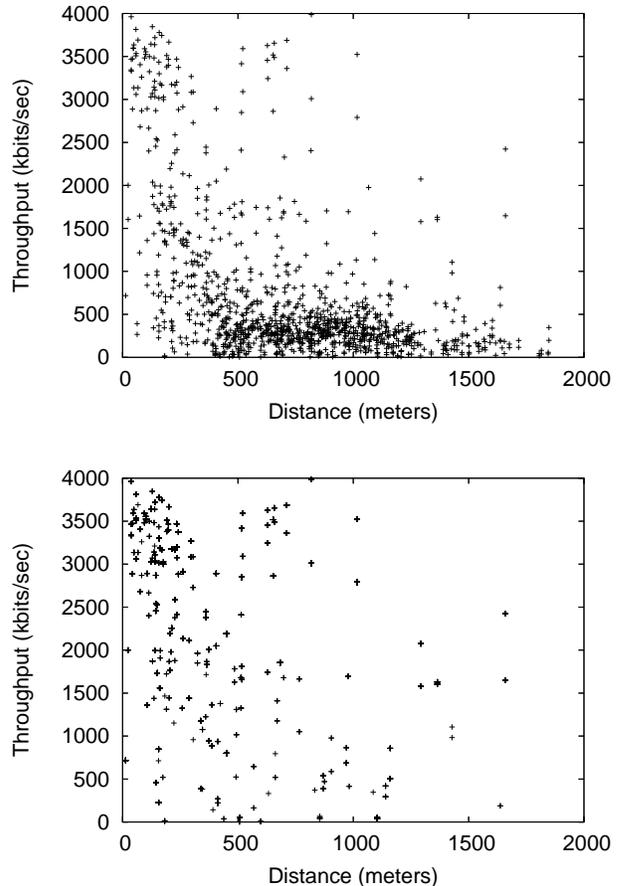

Figure 3: Link throughput versus distance for all links (top) and for the links used by Srcr (bottom). Srcr makes the most use of short high-throughput links. (single-hop TCP, multi-hop TCP)

throughputs of two megabits/second or more, and a few longer high-throughput links.

The lower graph shows just the links that Srcr uses in some route. Srcr uses almost all of the links faster than two megabits/second, but largely ignores the majority of the links, which are slower than that. This means that Srcr effectively favors short links of a few hundred meters, ignoring many links that would carry packets a kilometer or more in one hop. Fast short hops are the best policy: for example, four 250-meter hops that individually run at three megabits/second yield a route with a throughput of 750 kbits/second, which is faster than most of the single 1000-meter links.

A link's throughput is determined by its best transmit bit-rate and the delivery probability at that bit-rate. Figure 4 shows the CDF of delivery probabilities for the links used by Srcr at the bit-rate chosen by SampleRate. The median delivery probability is 0.8, and nearly a quarter of the links have loss rates of 50% or more. Section 2.5 justifies the use of links and bit-rates with significant loss rates, as opposed to favoring low-loss links. 802.11 detects the losses with its ACK mechanism and re-sends the packets; this decreases throughput but has little perceptible effect on latency, since the retransmissions occur within a few milliseconds.
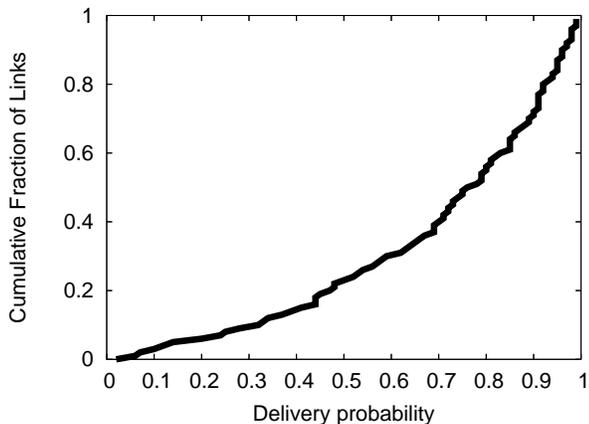
Figure 4: The distribution of delivery probabilities of links used in Srcr routes, at the bit-rates chosen by SampleRate. The median is 0.8, meaning that Srcr often uses links with loss rates of 20% or more. (multi-hop TCP, loss matrix)

3.4 Effect of Density

Mesh networks are only effective if the node density is sufficiently high. This section examines the effects of density by simulating different size subsets of Roofnet; subset size and density are roughly proportional, since the network area is about the same for all subsets.

The simulations function as follows. For each subset size n, a random set of n Roofnet nodes are selected. An estimate of the multi-hop throughput between every pair in the subset is computed, using only members of the subset as potential forwarders. The throughputs are estimated along the route that gives the highest throughput with Equation 1 and the single-hop TCP data-set.

Figure 5 shows the simulation results. The x axes show the subset size. The top graph shows the fraction of node pairs in the subset that are connected by a route that provides throughput of more than one kilobyte per second. The middle graph shows the average throughput over all pairs. The bottom graph shows the average hop-count of the routes. The ticks in each vertical line show 25th, 50th, and 75th percentiles over 100 random subsets.

The network only starts to approach all-pairs connectivity when there are more than 20 nodes, corresponding to a density of about five nodes per square kilometer. As the number of nodes increases, the average throughput also increases. The increase in hop-count in the third graph suggests the reason: a denser network offers a wider choice of short high-quality links, though using them causes routes to have more hops.

3.5 Mesh Robustness

This section investigates the benefits of the routing choices afforded by a mesh architecture and omni-directional antennas.

The most immediate measure of a mesh network's robustness is the number of potentially useful neighbors each node has. Figure 6 shows a histogram of the number of neighbors for each node, where a neighbor is defined as a node to which the delivery probability is 40% or more. Most nodes have
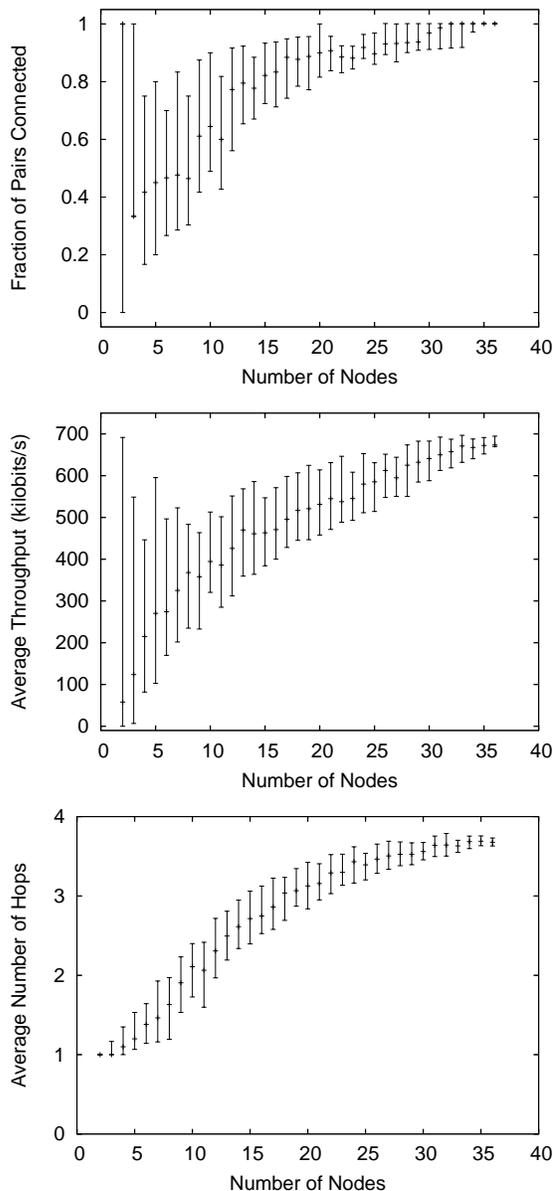

Figure 5: The simulated effects of node density on connectivity and throughput. The x axes show the number of nodes in the network. From top to bottom, the y axes show the fraction of node pairs that achieve throughput of more than one kilobyte per second; the average throughput over all pairs; and the average hop-count. Each bar shows the 25th, 50th, and 75th percentile over 100 random subsets. Increasing density causes the network to be more highly connected, and increases the average throughput; higher density allows routes to be constructed from shorter, higher-quality hops. (Simulated from single-hop TCP)
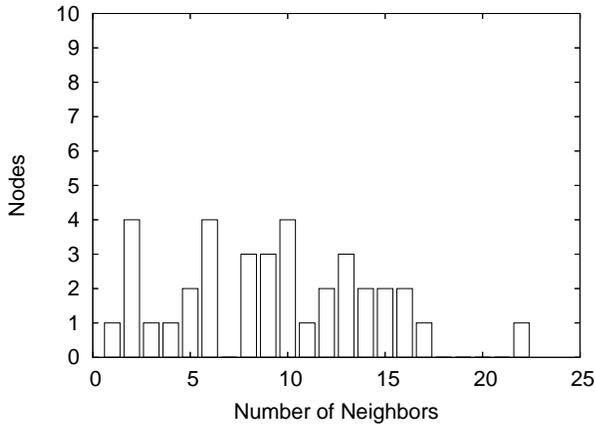
37

Figure 6: Histogram of the number of neighbors per node. A node counts as a "neighbor" if it has greater than 40% delivery probability for 1 megabit per second packets. (loss matrix)
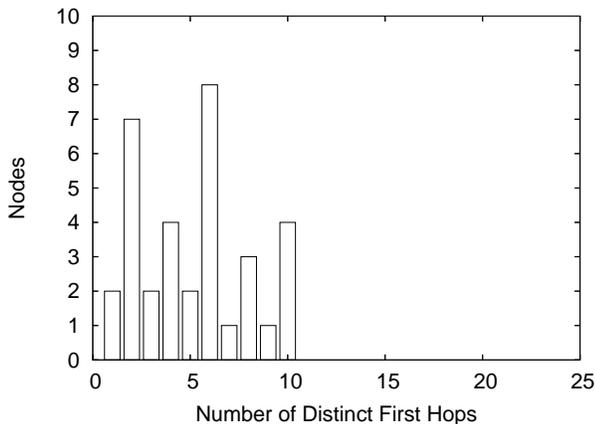

Figure 7: Histogram of the number of different first hops that Roofnet nodes use in all-pairs routes. (multi-hop TCP)
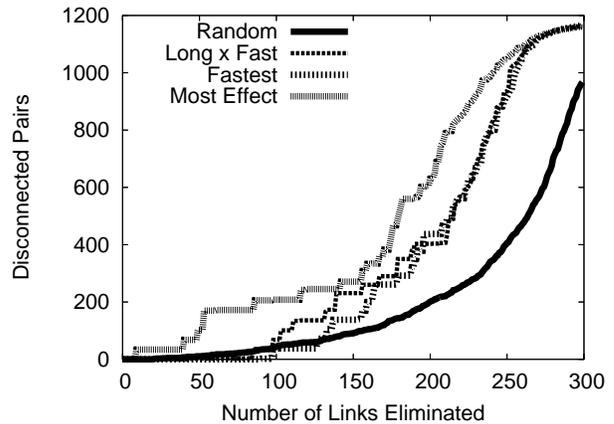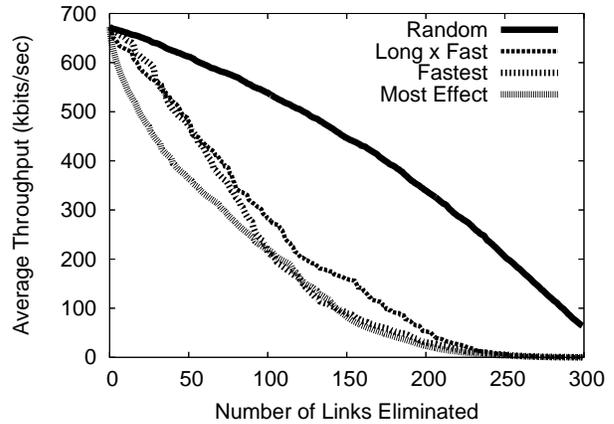

Figure 8: Simulated average throughput and connectivity among all pairs versus the number of links eliminated. Each curve shows the result of eliminating links in a particular order. (Simulated from single-hop TCP)

many neighbors, though there are a few poorly-connected nodes.

If a node never routes through one of its neighbors, the neighbor's value is questionable. For example, if most nodes routed through only one or two neighbors, it might be worth using directional antennas pointing just at those neighbors. Figure 7 shows a histogram of the number of unique first hop neighbors used by each node when routing to all the other nodes in the network. While some nodes do indeed route through only one or two neighbors, the majority of nodes use many more neighbors. In this sense Roofnet makes good use of the mesh architecture in ordinary routing.

Another aspect of robustness is the extent to which the network is vulnerable to the loss of its most valuable links. The graphs in Figure 8 shows how the average throughput and connectivity among all pairs decreases as individual links are deleted. The results are simulated with Equation 1 and the single-hop TCP data-set. Each line shows the cumulative effect of a different deletion order. Most Effect deletes the link whose absence decreases average throughput the

most; Long x Fast deletes the link with the highest product of distance and throughput; Fastest deletes the link with the highest link throughput; and the Random line shows the average of 40 simulations in which links were deleted in random orders.

Figure 8 shows that the best few links contribute noticeably to average throughput, but that dozens of the best links must be eliminated before throughput is reduced by half. The fastest links are generally more important than the long/fast links for throughput, though the first few long/fast links are more important than the fastest links, and both kinds are disproportionately important (deleting them lowers throughput faster than deleting random links). On the other hand, the long/fast links are more important for connectivity than the fastest links.

Figure 9 shows the effect on throughput of cumulatively eliminating the best-connected nodes. The throughputs are from the multi-hop density data set. The best-connected nodes are identified by looking for the nodes that appear in the most all-pairs routes.

Figure 9 shows that the best two nodes are important for performance, since losing both decreases the average throughput by 43%. The penalties for losing more nodes are more gradual.
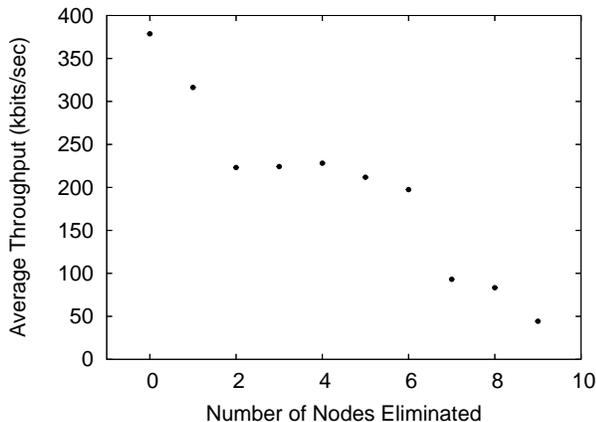
38

Figure 9: **The effect on throughput of eliminating the best-connected Roofnet nodes. The x axis shows the cumulative number of the best nodes eliminated. The y axis shows the average throughput among four particular nodes. (multi-hop density)**

| GWs | Multi-Hop | | Single-Hop | |
	Conn	Throughput (kbits/sec)	Conn	Throughput (kbits/sec)
1	37	781	23	174
2	37	1450	32	824
3	37	1871	34	1102
4	37	2131	36	1140
5	37	2355	37	1364
6	37	2450	37	2123
7	37	2529	37	2312
8	37	2614	37	2475
9	37	2702	37	2564
10	37	2795	37	2659
⋮	⋮	⋮	⋮	⋮
15	37	3197	37	3180
20	37	3508	37	3476
25	37	3721	37	3658

Table 4: **Comparison of multi-hop and single-hop architectures, with "optimal" choice of gateways. The GWs column shows the number of gateways. The Conn columns show the number of nodes with non-zero throughput to at least one gateway, and the Throughput columns show the throughput to the best gateway averaged over all connected nodes. For small numbers of gateways, multi-hop routing improves both connectivity and throughput. (multi-hop TCP and single-hop TCP)**

3.6 Architectural Alternatives

One way to evaluate Roofnet's architecture is to compare it to a network in which each node must communicate over a direct radio link to a gateway, without multi-hop routing. This corresponds to an access-point network. In such a single-hop network the number and placement of the gateways is important, since there must be a gateway near every node. This section evaluates the extent to which multi-hop routing improves performance and allows greater freedom in gateway placement.

3.6.1 Optimal Choice

Table 4 compares multi-hop and single-hop routing for increasing numbers of "optimally" chosen gateways. For single-hop, each successive gateway is the node that maximizes the number of additional nodes with non-zero throughput to some gateway, with ties broken by average throughput. The multi-hop gateways are chosen similarly, but with multi-hop connectivity and multi-hop throughput. The GWs column indicates the number of gateways. The Conn columns show the number of nodes which have non-zero throughput to at least one gateway. The Throughput columns show the average over these connected nodes of the throughput to each node's best gateway.

The data show that, in a single-hop architecture, five gateways are needed to cover all Roofnet nodes. For any given set of gateways, multi-hop forwarding provides higher average throughput. Multi-hop has a throughput advantage because it can often use a sequence of short high-quality links rather than a single long low-quality link.

The five optimal gateways turn out to be nodes located on three-story residences, not the tallest buildings in the network. This may be due to tallest buildings' locations around the network's perimeter.

3.6.2 Random Choice

It is likely that a community network would not be able to choose the best nodes to act as wired gateways, but would choose them more or less randomly. Table 5 shows performance for different numbers of randomly-selected gateways. For each number of gateways, the set of gateways was chosen by considering 250 distinct sets of that size and choosing the median set where the sets were sorted by the number of connected nodes, and ties were broken by the median average throughput. The reason that not all nodes are connected with small numbers of multi-hop gateways is the query failure problem mentioned in Section 3.1.

If Roofnet were a single-hop network, 25 gateways would be required to cover all the nodes. About 90% of the nodes are covered with 10 gateways, but there are a few nodes which are difficult to reach: the histogram in Figure 6 shows these last ten percent of nodes are within the range of three or fewer neighboring nodes. As with optimal gateway choice, multi-hop routing improves connectivity and throughput.

Comparison of the two tables shows that careful gateway choice increases throughput for both multi-hop and single-hop routing. For five or fewer gateways, even randomly chosen multi-hop gateways provide better performance than carefully chosen single-hop gateways. For larger numbers of gateways, however, carefully chosen single-hop gateways are better than randomly-chosen multi-hop gateways.

3.7 Inter-hop Interference

Table 1 shows that throughput with each additional hop falls off faster than one might expect. The average single-hop route provides 2451 kbits/second. One would expect two-hop throughput to be half that, or 1225 kbits/second. Instead, the average two-hop route delivers 771 kbits/second.

Figure 10 shows this phenomenon more clearly. Each point on the graph represents one node pair. The y-value of the point shows the measured throughput between that

39

| GWs | Multi-Hop | | Single-Hop | |
	Conn	Throughput (kbits/sec)	Conn	Throughput (kbits/sec)
1	34	760	10	535
2	35	1051	17	585
3	35	1485	22	900
4	35	2021	25	1260
5	36	1565	28	1221
6	36	1954	30	1192
7	36	1931	31	1662
8	37	1447	32	1579
9	37	1700	33	1627
10	37	1945	34	1689
⋮	⋮	⋮	⋮	⋮
15	37	2305	36	1714
20	37	2509	36	2695
25	37	2703	37	2317

Table 5: Comparison of multi-hop and single-hop architectures with random gateway choice. (multi-hop TCP and single-hop TCP)

| Hops | Pairs | Average Throughput | |
		without	with
1	3	2094	1735
2	5	836	725
3	6	314	312

Table 6: TCP throughputs without and with RTS/CTS for a few randomly-chosen node pairs, grouped by number of hops. The decrease for one-hop routes reflects the increased overhead of RTS/CTS, and the failure to increase multi-hop throughput suggests that RTS/CTS is not effective at avoiding collisions. (Measured)
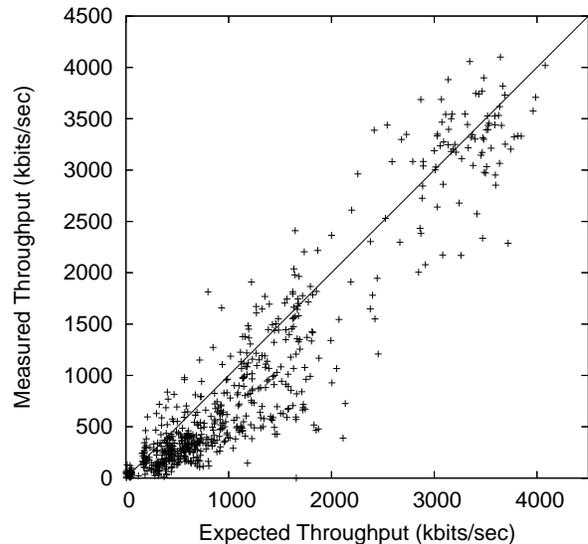

Figure 10: Comparison of multi-hop throughput and throughput predicted from individual hop throughputs. Each point represents one node pair. The expected throughputs for single-hop routes (most of the points above 2000 kbits/second) differ from the measurements, but the errors are not biased, since the predictions are themselves a separate set of measurements. The expected multi-hop throughputs are mostly higher than the measured throughputs. (multi-hop TCP and simulations from single-hop TCP)

pair of nodes. The x-value shows the throughput predicted along that route by Equation 1 and the single-hop TCP data-set. The expected throughputs for single-hop routes (most of the points above 2000 kbits/second) differ from the measurements, but the errors are not biased, since the predictions are themselves a separate set of measurements. In contrast, the measured multi-hop throughputs are mostly lower than expected.

A likely explanation of why links are slower together than separately is that concurrent transmissions on different hops of a route collide and cause packet loss. Informal experiments in which the sender delays each packet to give the previous packet time to reach the destination lead to increased throughput, which supports this hypothesis. Similar observations have been made by others [17].

The 802.11 RTS/CTS mechanism is intended to prevent these collisions. Table 6 shows the effect of RTS/CTS on throughput, measured between a random subset of node pairs. RTS/CTS does not improve performance. This is consistent with existing observations [33].

4. NETWORK USE

This section presents measurements of user activity on Roofnet. To collect this data, one of the four Roofnet gate-ways monitored the packets forwarded between Roofnet and the Internet.

In one 24-hour period, the gateway forwarded an average of 160 kbits/second between Roofnet and the Internet; this is the sum of the traffic in both directions. This data accounted for about 94% of the wireless traffic that the gateway sent or received; the other 5% were protocol control packets.

About 48% of the data traffic was to or from nodes one hop from the gateway, 36% two hops, and the rest, about 16%, was forwarded over three hops or more.

The gateway's radio was busy for a total of about 59,374 seconds during the 24-hour period. This total includes time for packets and retransmissions received, packets and retransmissions sent, and transmissions overheard by the gateway that were not addressed to it. The total does not include inter-frame spacing and back-off time. This means the gateway's radio was busy for about 70% of the monitoring period.

Almost all of the packets forwarded were TCP; less than 1% were UDP. Web requests to the Internet comprised only about 7% of the bytes transferred. The biggest consumer of bandwidth, accounting for about 30% of the total data transferred, was the BitTorrent peer-to-peer file-sharing program. Most of the rest of the traffic consisted of short connections to other unprivileged ports.

While web requests accounted for a minority of the data transferred, they did account for more connections than any

other application: 68% of the connections through the gateway were web connections. Just 3% were BitTorrent.

During the 24-hour period, there were 16 Roofnet hosts that accessed the Internet; eight opened more than 100 TCP connections to the Internet during that time.

5. RELATED WORK

There have been a number of evaluations of deployed or test-bed multi-hop wireless networks. Some have focused on evaluating route metrics intended to increase throughput in static mesh networks [14, 13]. Others have primarily considered route repair in the face of mobility [27, 19]. Some have investigated link-level 802.11 behavior in order to guide the design of higher-layer protocols [16, 25, 23, 7]. Our work is unique in that it evaluates a deployed mesh network with active users, and considers the effects of architectural decisions rather than protocol design.

Many of the basic ideas in wireless mesh networking were first developed for the DARPA Packet Radio Network [21]. Roofnet uses derivatives of routing protocols from prior research in mobile ad hoc networking; Srcr is loosely based on DSR [20] and MCL [14]. A number of research groups maintain wireless testbeds with which to evaluate real-world performance of MANET protocols [27, 26, 28, 25, 11].

A number of community wireless mesh network efforts exist, such as Seattle Wireless, the San Francisco BAWUG, the Southampton Open Wireless Network, Wireless Leiden [31], and the Digital Gangetic Plains project [29]. Many of these mesh nets use directional antennas and the OSPF routing protocol. This approach works well for relatively sparse networks, whereas Roofnet targets future networks with dense nodes. Drunen et al. [31] discuss awkward aspects of directional antennas, including the effort required to plan, install and periodically adjust the antennas.

Commercial mesh Internet access services and technologies exist, such as MeshNetworks Inc., Ricochet [30], and Tropos Networks.

Sensor networks use multi-hop wireless networks to collect data, and thus face problems similar to those of Roofnet. Yarvis et al. [34] and Woo and Culler [32] observe that hop-count performs poorly as a routing metric for a sensor network, and present improved loss-aware metrics. Ganesan et al. [18] describe the behavior of a wireless sensor net, including some unexpected phenomena similar to those seen in Roofnet.

6. CONCLUSIONS

This paper outlines the design and evaluates the performance of an urban rooftop 802.11b mesh network. The network's architecture favors ease of deployment in its use of omni-directional antennas, self-configuring software, and link-quality-aware multi-hop routing. Through the participation of volunteers, the network has grown to 37 nodes in the course of a year, with little administrative or installation effort on the part of the researchers.

An evaluation of network performance shows that the unplanned mesh architecture of Roofnet works well: average throughput between nodes is 627 kbits/second, and the entire network is well served by just a few Internet gateways whose position is determined by convenience. Compared to a hypothetical single-hop network, Roofnet's multi-hop mesh increases both connectivity and throughput.

8. REFERENCES

[1] Champaign-Urbana Community Wireless Network CUWiN. http://www.curwireless.net/.

[2] Locust World.org. http://www.locustworld.com/.

[3] meshcube.org. http://www.meshcube.org/.

[4] NYCWireless. http://www.nycwireless.com.

[5] Seattle Wireless. http://www.seattlewireless.net/.

[6] Tropos networks technology whitepaper, May 2003. http://www.troposnetworks.com/.

[7] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. A measurement study of a rooftop 802.11b mesh network. In *Proc. ACM SIGCOMM Conference (SIGCOMM 2004)*, September 2004.

[8] Pravin Bhagwat, Bhaskaran Raman, and Dheeraj Sanghi. Turning 802.11 inside-out. In *Second Workshop on Hot Topics in Networks (HotNets-II)*, November 2003.

[9] John Bicket. Bit-rate selection in wireless networks. Master's thesis, Massachusetts Institute of Technology, February 2005.

[10] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. ACM/IEEE MobiCom*, pages 85–97, October 1998.

[11] Kwan-Wu Chin, John Judge, Aidan Williams, and Roger Kermode. Implementation experience with MANET routing protocols. *ACM SIGCOMM Computer Communications Review*, 32(5), November 2002.

[12] Samir Das, Charles Perkins, and Elizabeth Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proc. IEEE Infocom*, pages 3–12, March 2000.

[13] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom '03)*, San Diego, California, September 2003.

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

[15] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 114–128. ACM Press, 2004.

[16] D. Eckhardt and P. Steenkiste. Measurement and analysis of the error characteristics of an in-building wireless network. In *Computer Communication Review 26:4, pp. 243-254, SIGCOMM '96*, October 1996.

[17] Zhenghua Fu, Petros Zerfos, Haiyun Luo, Songwu Lu, Lixia Zhang, and Mario Gerla. The impact of multihop wireless channel on TCP throughput and loss. In *IEEE INFOCOM'03*, March 2003.

[18] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks. Technical report UCLA/CSD-TR 02-0013, UCLA CS Department, 2002.

[19] Robert S. Gray, David Kotz, Calvin Newport, Nikita Dubrovsky, Aaron Fiske, Jason Liu, Christopher Masone, Susan McGrath, and Yougu Yuan. Outdoor experimental comparison of four ad hoc routing algorithms. In *ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2004.

[20] 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.

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

[22] 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.

[23] David Kotz, Calvin Newport, Robert S. Gray, Jason Liu, Yougu Yuan, and Chip Elliott. Experimental evaluation of wireless simulation assumptions. In *ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2004.

[24] 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 Computing and Networking*, pages 61–69, Rome, Italy, July 2001.

[25] Henrik Lundgren, Erik Nordstrom, and Christian Tschudin. Coping with communication gray zones in IEEE 802.11b based ad hoc networks. In *ACM WoWMoM Workshop*, September 2002.

[26] 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.

[27] 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.

[28] Eric Nordstrom. APE - a large scale ad hoc network testbed for reproducible performance tests. Master's thesis, Uppsala University, June 2002.

[29] Bhaskaran Raman and Kameswari Chebrolu. Revisiting MAC design for an 802.11-based mesh network. In *Third Workshop on Hot Topics in Networks (HotNets-III)*, November 2004.

[30] M. Ritter, R. Friday, R. Garces, W. San Filippo, and C-T. Nguyen. Mobile connectivity protocols and throughput measurements in the Ricochet microcellular data network (MCDN) system. In *Proc. ACM/IEEE MobiCom*, July 2001.

[31] Rudi van Drunen, Jasper Koolhaas, Huub Schuurmans, and Marten Vijn. Building a wireless community network in the Netherlands. In *USENIX/Freenix Conference*, June 2003.

[32] A. Woo and D. Culler. Taming the challenges of reliable multihop routing in sensor networks. In *ACM SenSys*, November 2003.

[33] Kaixin Xu, Mario Gerla, and Sang Bae. Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks. *Ad Hoc Network Journal*, 1(1), July 2003.

[34] Mark Yarvis, W. Steven Conner, Lakshman Krishnamurthy, Jasmeet Chhabra, Brent Elliott, and Alan Mainwaring. Real-world experiences with an interactive ad hoc sensor network. In *Proceedings of the International Workshop on Ad Hoc Networking*, August 2002.