Unwire Portland Proof-of-Concept Network Testing

Caleb Phillips Russell Senior

May 23, 2007

Abstract

On October 27, 2006, the City of Portland issued an Informal Request for Proposals (IRFP) for testing the Proof-of-Concept (POC) network being built by MetroFi, a company based in Mountain View, California. The authors of this report participated in a bid on the project but were not selected. They were, however, interested in learning for themselves about how well or poorly the POC network was performing and decided to pursue their own testing in parallel with the City-sponsored testing.

This report describes the testing methodology and results obtained. The authors devised a three-part testing methodology which leverages simple and trusted statistical methods to make inferences about the performance of the city-wide mesh network.

This investigation found that the POC network as deployed by MetroFi complied with many of the metrics required by the Testing IRFP. The POC network principally fails in the coverage criterion, where a value of 58% usable coverage was found, well-short of the 90% goal specified.

It should be noted that both authors are active volunteers for the Personal Telco Project, a public non-profit corporation based in Portland, Oregon. Personal Telco is engaged in free-as-in-freedom community wireless networking. This investigation was performed by the authors as a public service using their own resources. This report is solely the product of the authors and does not necessarily reflect the views of the Personal Telco Project.

About the Authors

Caleb Phillips has been working closely with wireless networks for over two years and has been self-employed as a programmer and IT Consultant for more than 6 years. He is currently employed as a student researcher in the Mobile Computing Lab at Portland State University under his advisor, Dr. Suresh Singh, where his primary research is in characterization and modeling of wireless networks. In the Fall of 2007, he is planning to start his Ph.D. research in the area of wireless networking at the University of Colorado at Boulder. In addition, he is currently serving as the Director of Education for the Personal Telco Project and as a member of the Network Operations Team which is responsible for maintaining the 100+ Personal Telco nodes city-wide.

Russell Senior has nearly 20 years of experience working on diverse data collection and scientific data management problems. He serves as Secretary of the Personal Telco Project and is the technical lead on the Mississippi Grant Project, a neighborhood-wide community wireless network in the area surrounding Mississippi Avenue in North Portland. Russell has also led a wireless network mapping effort for the group, the purpose of which is to determine the current extent of Personal Telco network growth in the community. He has experience successfully building embedded computing devices for supporting the mapping effort among other purposes.

Acknowledgements

We would like to thank Jim Binkley of the Department of Computer Science at Portland State University for his sage advice concerning methodologies for measuring network performance, and for reviewing this paper. We also thank the others that reviewed drafts of this report, including Darrin Eden, Michael Weinberg and Logan Kleier. We are indebted to Jason McArthur for the basic design of the devices used in this investigation and to Gus Swanson who assisted during some of the data collection. We also thank the private property owners that granted us access to locations used in some of our testing. Russell Senior would like to thank his wife Heidi and the rest of his family for allowing him the time to participate in this investigation.

1 Background & Scope

In September of 2005, the City of Portland issued a Request for Proposals to build and operate a "city-wide broadband wireless system"[1]. In April of 2006, the City chose MetroFi (Mountain View, CA) as the winning bidder, and in the following summer the City and MetroFi signed a Non-exclusive License Agreement[2]. Thereafter, MetroFi began to deploy their network in preparation for a December 2006 launch of a Proof-of-Concept (POC) network, as called for in the agreement. The deal was structured such that the POC network would first be built and afterward an independent third party would test it. When the City was satisfied that the POC network met its performance criteria, it would issue a Certificate of Acceptance, whereupon MetroFi would be permitted to begin rolling out its service to the rest of the city.

On October 27, 2006, the City issued an Informal Request for Proposals[3] (IRFP) to perform the independent third-party testing. The City's project manager, Logan Kleier, approached computer science professor Bart Massey at Portland State University (PSU) to solicit a bid on the project. Caleb Phillips joined in the bid. On November 7th, Russell Senior asked to participate as well. A PSU bid was submitted on the November 10th, 2006 deadline. On January 8, 2007 the PSU bidders learned that they had not been selected, but a contract was instead being negotiated with Uptown Services, Inc. (Boulder, Colorado).

Because we had become intellectually engaged in the technical aspects of how to execute the study and because we expected the investigation would be both fun and interesting, we decided on January 11, 2007 to go ahead and do what we considered the interesting and/or more tractable parts of the Testing IRFP on our own. We felt that, in addition to being fun and interesting, our investigation would provide a useful check and/or comparison to the official testing project, and thus be valuable as an informational public service to the community. Portland is an early adopter in the area of municipal wireless networks, so we additionally hoped that our findings and our approach would help to inform testing of future municipal wireless networks[4].

Because of limitations of time and resources, we restricted our investigation to three principal areas: the generation of coverage maps, the characterization of variability of network performance over time, and an assessment of coverage completeness. In the course of performing these tests, we made miscellaneous observations about the way the MetroFi network functions, which we also discuss.

All of the testing described here was performed in a passive manner, without any special access or cooperation by the network operator, MetroFi.

[1] http://www.portlandonline.com/shared/cfm/image.cfm?id=129687

[2] http://www.portlandonline.com/shared/cfm/image.cfm?id=129511

[3] http://www.portlandonline.com/shared/cfm/image.cfm?id=150071

[4] All of our data, software, and configuration will be made available for public inspection and reuse, at http://unwirepdx-watch.org

2 Network Mapping

Among the deliverables called for in the Testing IRFP was "a graphic representation of the coverage areas that exceed, meet, or do not meet the performance criteria."[5]. As we began our own investigation, we needed to identify and establish the locations of functioning access points in the POC network. We generated maps of signal levels available from the POC network in order to satisfy those requirements.

2.1 Methodology

Coverage maps were generated with data collected from automobile-mounted radio equipment and global positioning system (GPS) devices. Over the course of several days in the second half of February 2007, we drove a vehicle on virtually all publicly accessible streets, as well as some accessible parking lots, in the POC area. The equipment consisted of a roof-mounted magnetic-base 7 dBi antenna, attached to a Wistron Neweb CM9 Atheros-based wireless radio, mounted in a single-board computer and powered by a 12 Volt battery. A GlobalSat BU-353 GPS receiver simultaneously collected location data. The data acquisition equipment utilized Open Source software, Kismet[6] and gpsd[7], running the Linux operating system.

In the United States, 802.11b/g (Wifi) operates on a set of 11 channels in a frequency band above 2.4 GHz. The receiving radio used in this investigation can be tuned to only one of these channels at a time. The software compensates for this by frequently "hopping" from one channel to the next, dwelling on each channel long enough to receive beacons typically transmitted by access points approximately 10 times per second. Each channel is visited every 2.2 seconds. For example, a vehicle moving at 20 mph would visit a channel for about 6 feet in every 65 feet travelled.

Each transmission from a wifi radio contains a unique identifier of the radio that sent it, called the Basic Service Set Identifier (BSSID). Each time the mapping radio receives one of these transmissions, various characteristics are recorded: the BSSID of the sender; the location of the mapping receiver at the time of reception; and the relative signal strength indicator (RSSI). Utilizing the relative strength of signals received in various locations from a single transmitting radio, the location of the transmitter may be estimated. Furthermore, the same information provides an estimate of the extent of the coverage, directly along the lanes of streets and, by inference, in locations near the streets.

It is important to note that computer networking, in contrast to a broadcast system like television, requires bi-directional communication to function. A television station transmits its signal at very high power, and individual receivers passively listen to those broadcasts. In contrast, a web server needs to receive a request from a client before it sends the requested content. In a wireless network, it is not enough that the client device, a laptop for example, can "hear" the access point. For the wireless network to function, the access point needs to be able to "hear" the client device as well. A laptop that does not have as much transmit power as the access point might well be unable to get its traffic back to the access point. The network mapping described here involves simply "hearing" a MetroFi access point. Reception at a particular location in this context does not imply that a connection can

[5] Section 4(b)

[6] `http://www.kismetwireless.net/`

[7] `http://gpsd.berlios.de/index.html`

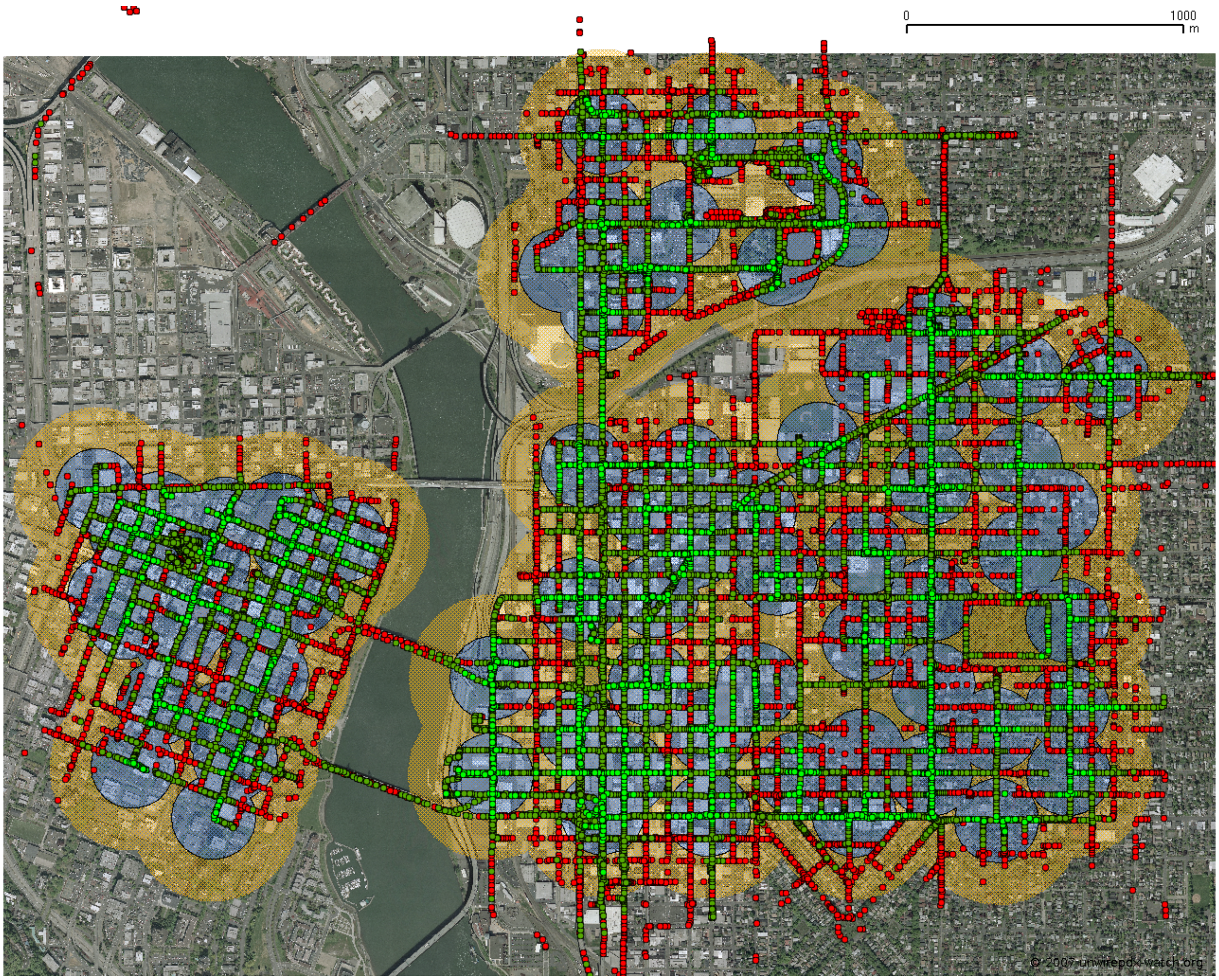

Figure 1: Observed signal strength during network mapping. -95 to -75 dBm (red), -74 to -55 dBm (dark green), >-55 dBm (light green)

necessarily be established there.

2.2 Results

Between February 19 and February 28, 2007, we mapped the POC network. We drove approximately 210 miles of streets with the radio gear collecting data. Figure 1 shows the signal-strength values collected during the mapping. Note that the signal strength values are enhanced by the 7 dBi antenna employed. For a typical client device (where a 2 dBi antenna gain would be more common), the values should be reduced accordingly. Triangulating from this data, we estimated locations for the MetroFi access points, associating the observed BSSID's to the access points located there.

The Testing IRFP listed locations for 67 access points. However, we found the accuracy of those locations inconsistent and sometimes off by as much as a few blocks. Consequently, we revisited each observed access point to check the location with a hand held GPS. The locations were further refined using the orthoimagery provided by Google Maps. We estimate the revised access point locations are

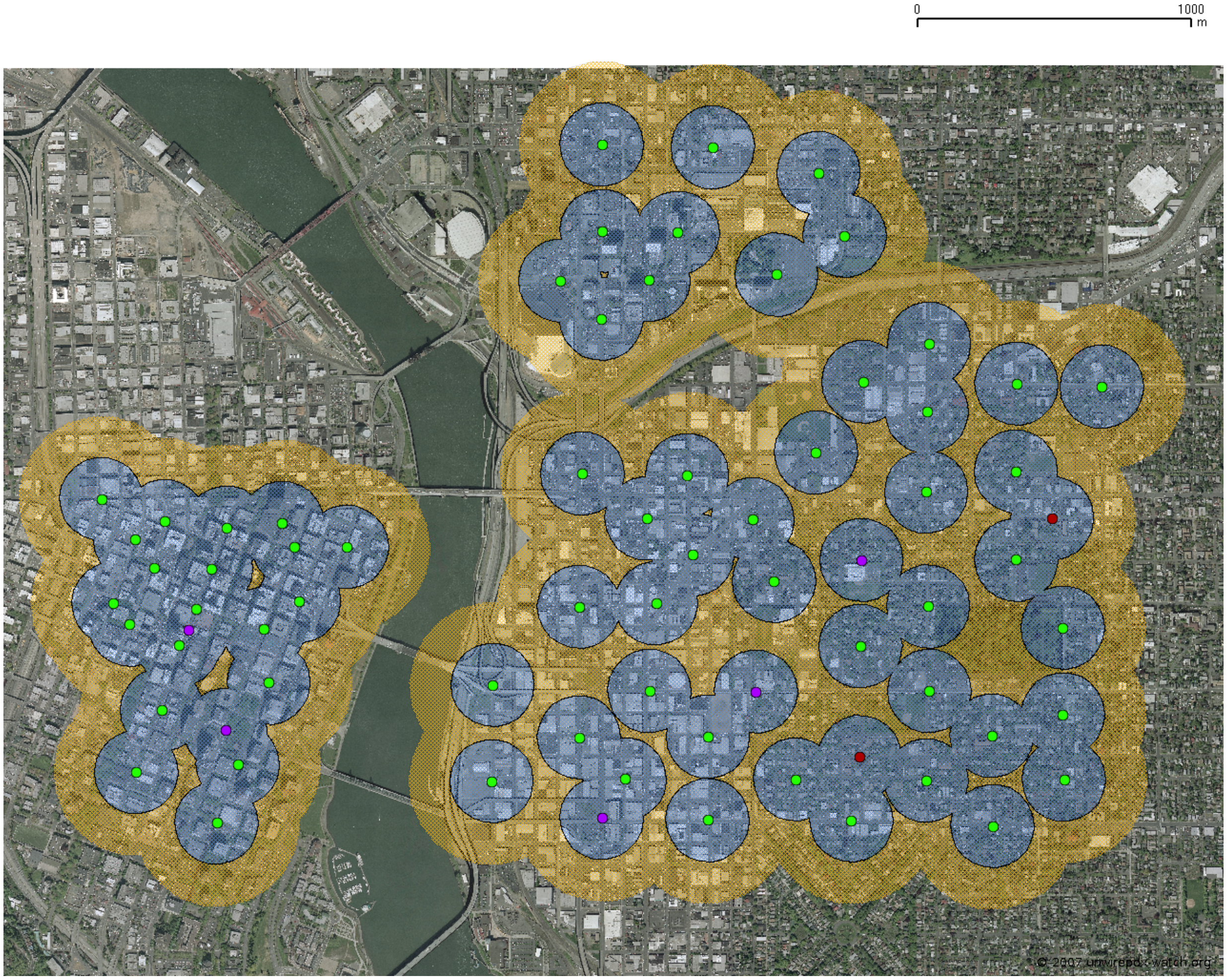

Figure 2: 72 POC access points with advertised coverage (orange) and city-tested area (blue)

accurate to within about 5 feet.

In our mapping effort, we located 5 MetroFi-Free access points that were not listed in the Testing IRFP as part of the POC network. These were located at SE 12th and Belmont, SE 17th and Pine, SE Grand and Hawthorne, SW 4th and Madison, and SW Broadway and Yamhill. We detected one additional MetroFi-Free access point near the Marquam Bridge, but it was observed inconsistently and we lacked adequate data to pinpoint its location. We excluded it in these analyses. A total of 72 MetroFi-Free access points were considered part of the POC area network in these analyses. A map of the 72 access points is shown in Figure 2. Green and red represent the 67 access points listed in the Testing IRFP; purple represents the 5 operating access points not included in the IRFP.

Of the 72 access points considered, two of them (shown as red in Figure 2) did not appear to be operating local coverage radios during this survey. Of the remaining 70 access points, we observed that two of them operated on different channels at different times: 00:0A:DB:01:74:E0 (6 and 11) and 00:0A:DB:01:9E:E0 (1 and 11). The distribution of access points across channels is presented in Table 1. The BSSID, approximate intersection and latitude, longitude of each of these 72 access points are listed in Appendix A.

Channel(s)	Number of Access Points
1	25
6	17
11	26
1 and 11	1
1 and 6	1
N/A	2

Table 1: Channel Assignment for POC Access Points

3 Network Performance and Its Variability Over Time

In this part of our investigation, we address the following questions from the Testing IRFP:

- the availability of a "1 Mbps downstream/256 Kbps upstream connection in a stationary position" (I.A);
- the "network's ability to provide a connection (64 Kbps upstream and downstream) 99% of the time" (II.A);
- the "network's ability to deliver packets between an end user device and the MetroFi Point of Presence in less than or equal to 100 milliseconds" (II.B);

3.1 Methodology

We chose three test sites in different areas of the POC network where we were able to acquire permission to deploy a test device for a week or more at a time. A larger, more random selection of sites would provide a more reliable assessment of overall network performance; however, our limited sample provides some diversity in location and proximity to access points so as to provide a reasonable suggestion as to what one might expect more generally.

The three sites were located as follows:

- A: near NE 20th and Irving (about 70 feet to nearest AP)
- B: near SW 10th and Washington (about 200 feet to nearest AP)
- C: near SW 5th and Alder (about 420 feet to nearest AP)

The test device consisted of a small, single-board computer with an Atheros 5213 802.11b/g radio and an attached 2 dBi omni-directional antenna[8]. In this configuration the transmit power of the test device radio was 20 dBm (100 mW). We programmed the device to record its data onto a USB-storage

[8] The test device consisted of stock Netgear WGT634U hardware, with OpenWrt firmware.

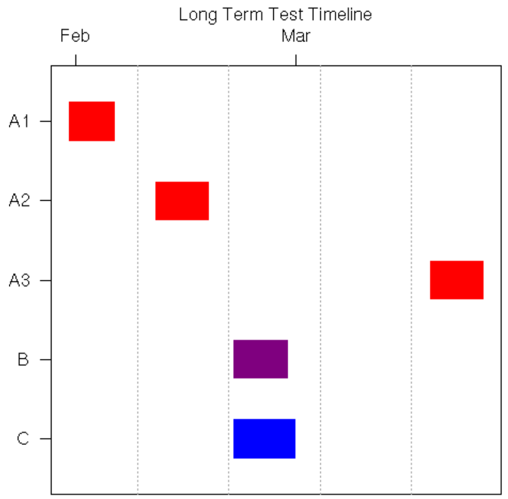

Figure 3: Timeline for Longterm Tests

device, which was retrieved for analysis at the conclusion of the measurement period. We placed the device near an AC outlet that could be used for power.

Each test run was started manually by logging into the test device and initiating a script (see Appendix B.1). The script started a series of subscripts that ran in parallel, sending their output to individual log files. Tests which required a destination were directed at different servers on the same remote network and staggered so as to prevent interference between tests (and test devices). Figure 3 shows when the tests occurred at each site.

One of the challenges for the longterm test was to deal with the ad-supported nature of the MetroFi-Free network. Every so often, MetroFi requires the user to retrieve an advertisement via HTTP in order to keep passing network traffic. The test device did not have a graphical web browser that would allow viewing advertisements, nor was a human present continuously to view them. One of the subscripts (see Appendix B.2) was dedicated to satisfying this requirement[9].

In order to test latency and packet loss, one of the subscripts (see Appendix B.3) was dedicated to running a ping command repeatedly. In a loop this script ran a command that sent 10 ping request packets at 1-second intervals to a well-connected host on the Internet and logged a timestamp and the result to a file. In between each loop, the script slept for 1 second. Each successful ping reply identified the difference between the time the ping request was sent and when the matching ping reply was received in milliseconds, providing a measure of round-trip latency in the network. Each ping request without a matching ping reply indicated a lost packet due to network congestion or other network failures.

To test the available bandwidth to a client device, one of the subscripts (see Appendix B.4) was dedicated to periodically measuring upstream and downstream transfer rates. In a loop, this script ran ttcp[10] to measure upstream transfer rates. A well-connected host on the Internet was configured to

[9] The authors do not endorse systematically bypassing advertisements on MetroFi's ad-supported network, however this technique was necessary to test the variability of performance consistently over extended time periods.

[10] `http://ftp.arl.mil/pub/ttcp/sgi-ttcp.c`

run a corresponding ttcp program for the client device to talk to. The ttcp program carefully times data transfers and reports detailed results summarizing performance to a log file. For this test, 16 megabytes of data were tranferred from the client device to the server (2048 8KB TCP/IP packets). After the ttcp test completed, two files of 1 and 5 megabytes, respectively, were downloaded from the same server using the program wget, a command-line HTTP fetching utility. The time required for each download was recorded to a log file. The script slept for 5 minutes between each test.

In order to assess parameters relating to the association of the client device to the MetroFi network, one of the subscripts (see Appendix B.5) was dedicated to periodically recording the state of the connection from the client device's point of view. A system command[11] was executed in a loop to report the state of the wireless association. A timestamp and the results of this command were stored in a log file. Between each loop the script slept for 1 minute.

3.2 Results

Our analysis here is divided into three sections: physical layer; throughput; and latency and loss. Physical layer analysis involves the condition of the wireless medium (the interaction between the client and access-point radios), throughput is the rate at which a network is able to transfer data between two locations, latency is the time required to move a small piece of data across the network, and loss is a measure of how reliably data are delivered across the network.

3.2.1 Physical Layer

In this section we characterize the temporal stability and performance of the physical layer. We focused our analysis on several metrics: association, received signal strength (in dBm), channel (frequency) usage, and data rate.

Table 2 presents a breakdown of the access points used at each site. Two of the test sites connected to multiple nearby access points, though both had a strong preference. Table 3 presents the duration of each test, as well as the probability of being associated. We can see that at each site, the probability of losing an association is less than 0.01, which indicates that the network met the 99% uptime requirement at the sites tested.

Figure 4 presents observed signal strength by time of day for each site. Table 4 reports the relative frequencies that various modulation rates supported by 802.11b/g (1-54 Mbps) were used. The sites with weaker signal strength utilized lower data rates more often.

3.2.2 Throughput

The Testing IRFP asks for an assessment of 1Mbps downstream and 256Kbps upstream throughput at a stationary location. Figures 5 and 6 present downstream and upstream throughput by time of day.

[11] The system command was the standard "iwconfig" from the Linux Wireless Extensions which receives information from the wireless device driver, Madwifi (`http://madwifi.org/`) in our case.

Site	BSSID	Channel	Probability	Mean Signal (dBm)	Signal Std. Dev. (dBm)
A	00:0A:DB:03:81:A0	6	1	-46.6	3.4
B	00:0A:DB:01:A0:60	1	0.999806	-67.1	2.3
B	00:0A:DB:01:86:00	11	0.000193	-80.0	1.4
C	00:0A:DB:01:86:00	11	0.999829	-74.6	1.6
C	00:0A:DB:03:49:40	6	0.000170	-82	—

Table 2: APs used at each Longterm Test Site

Site	Duration of Test (hours)	Probability of Disassociation
A	456.44	0.00149
B	173.83	0.00106
C	197.53	0.00449

Table 3: Site Test Durations and Disassociation Probabilities

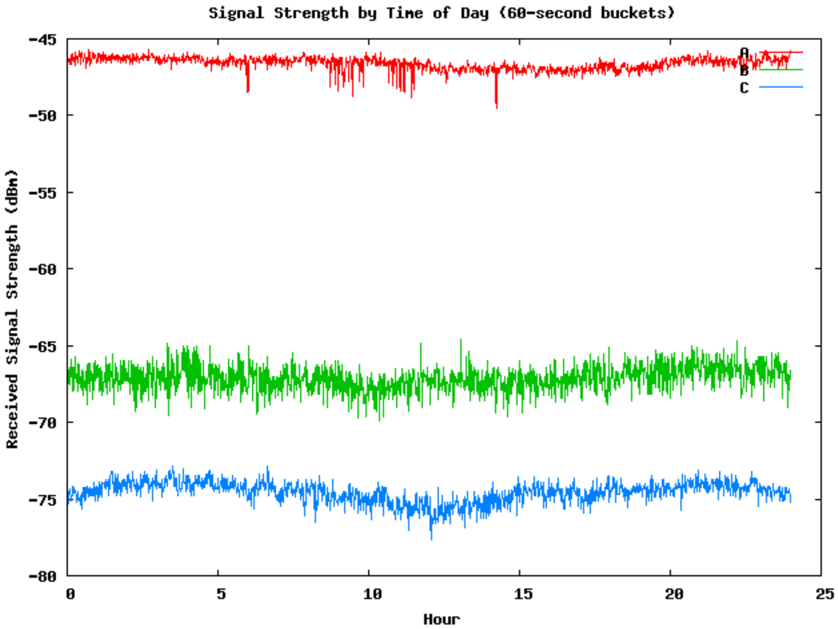

Figure 4: Signal Strength by Time of Day at Longterm Test Sites

Rate (Mbps)	Site A	Site B	Site C
Total Count	31456	10356	11761
1	1 (<0.01%)	490 (4.73%)	10902 (92.70%)
2	54 (0.17%)	960 (9.27%)	254 (2.16%)
5	8319 (26.45%)	4145 (40.03%)	337 (2.87%)
11	23071 (73.34%)	4762 (45.98%)	268 (2.28%)
24	3 (0.01%)		
36	4 (0.01%)	1 (0.01%)	
48	2 (0.01%)		
54	2 (0.01%)		

Table 4: Rate Frequencies over Duration of Longterm Test

Site	Up Mean (Kbps)	Up Std. Dev (Kbps)	Down Mean at 1MB (Mbps)	Down Std. Dev. at 1MB (Mbps)	Down Mean at 5MB (Mbps)	Down Std. Dev at 5MB (Mbps)
A	388.89	23.023	0.66179	0.53252	0.63106	0.47927
B	393.45	27.184	2.4624	0.59123	2.0248	0.90238
C	139.67	65.488	0.50017	0.75346	0.70811	1.1961

Table 5: Throughput Summary

Table 5 summarizes those findings. At the two sites with good signal strength, the throughput is quite stable. However, at the site with a weaker signal, the throughput in both directions is barely half of the standard and experiences greater variability. The odd-looking graph for upstream throughput at Site C is due to the way ttcp tests throughput—if there are dropped packets, the connection will stall and it will exit without providing a measurement (as it would be affected by packet loss). Since this test shows high packet loss during the second half of the day (as we will see in Section 3.2.3), it makes sense that we have almost no data for that part of the day.

The average throughput measurement for Site A is muddied by changes we observed in bandwidth during testing. Our initial tests (January 31st to February 19th) for Site A put it well above the 1Mbps requirement, but during later tests (March 18th to March 25th) the throughput is apparently throttled at or below 500 Kbps. Based on the earlier data, we know that the network is capable of sustaining 1Mbps, but based on the later data, it appeared as if a limit had been imposed by the network operators substantially under 1Mbps. This issue is discussed below in Section 3.2.4. When the physical layer is stable, it appears the network is *capable* of supporting the required throughput.

Note that we did not have access to any information regarding the overall load on the MetroFi network during our tests. It is not clear from these limited tests how the network throughput will be affected if usage grows significantly.

3.2.3 Latency and Loss

In this section, we characterize the temporal stability of the network by looking at the round-trip latency (the time it takes for a packet to traverse the network and return) and packet loss (likelihood

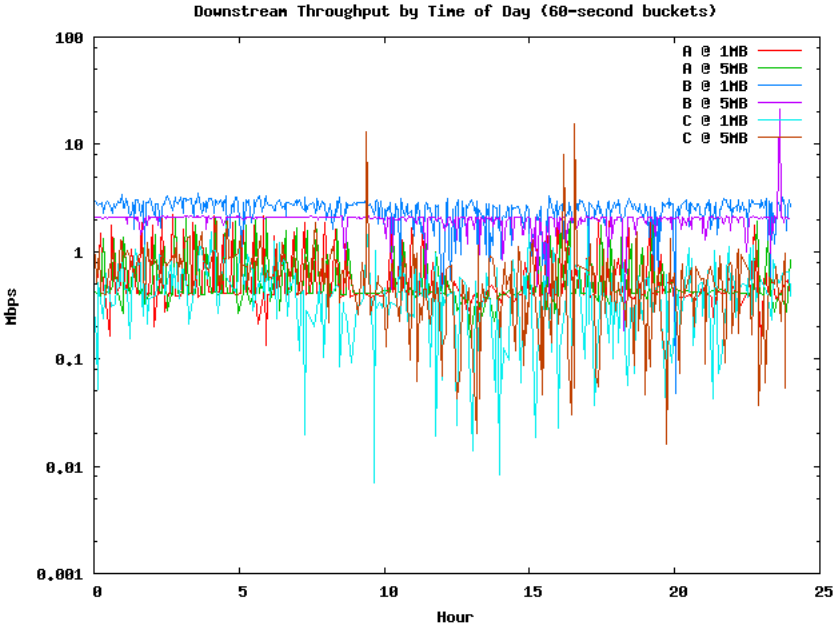

Figure 5: Downstream Throughput by Time of Day at Longterm Test Sites

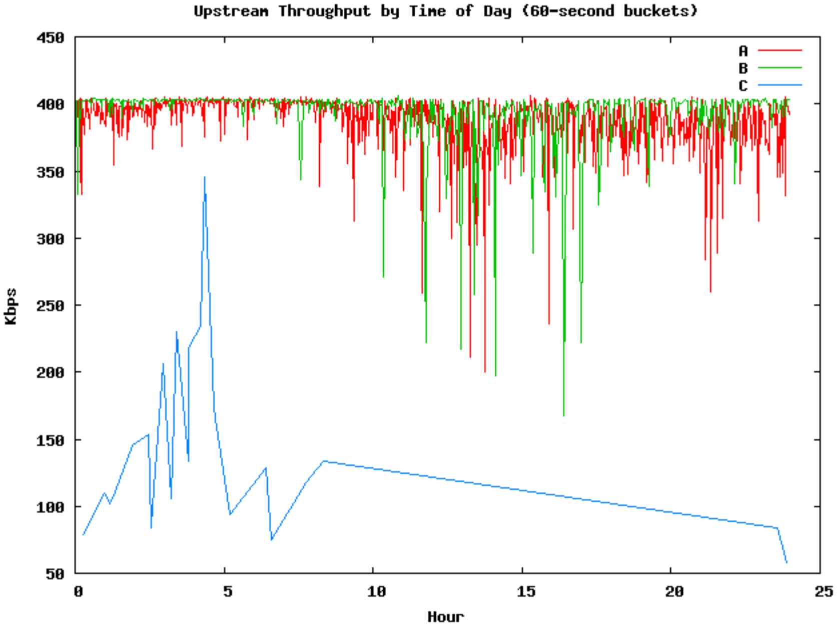

Figure 6: Upstream Throughput by Time of Day at Longterm Test Sites

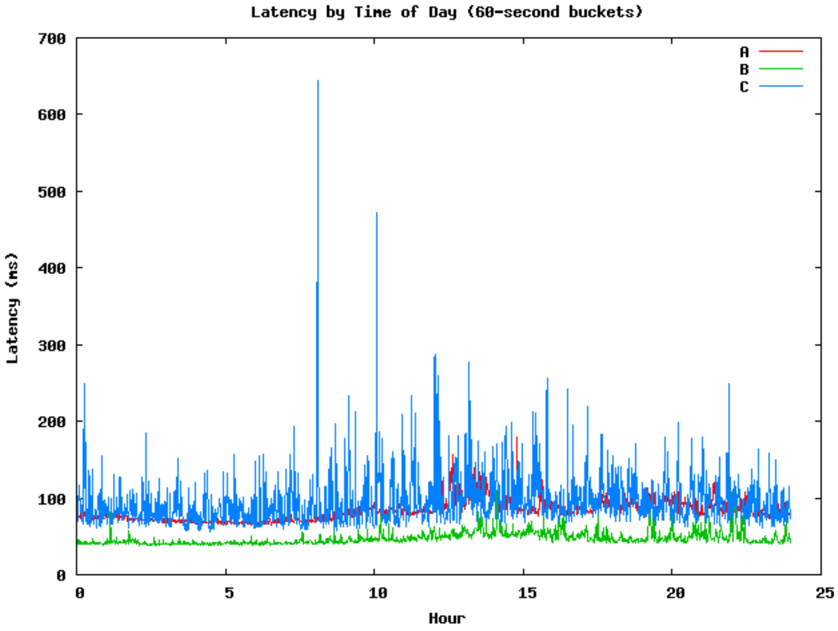

Figure 7: Round-trip Latency by Time of Day at Longterm Test Sites

for such a packet to be dropped somewhere along the way).

In Figure 7, we show the measured latency by time of day at the three sites, and in Table 6, the overall averages for latency.

The language of the Testing IRFP seems odd on the issue of latency, asking to assess "[t]he network's ability to deliver packets between an end user device and the MetroFi Point of Presence (POP) in less than or equal to 100 milliseconds." Since latency is measured in terms of round trips (because it allows one clock to be used to measure both the time the request is sent and the time the reply is received), specifying a delivery time in only one direction, as this language does, may have been unintentional. However, as the plain language specifies a one-way time, we have assumed that delivery times are symmetrical, and have simply extrapolated the standard to a presumably equivalent round-trip time of 200 milliseconds or less. At all three sites, the network outperforms the 200 ms round-trip latency standard most of the time.

Packet loss is presented in Table 6 and Figure 8. There is an observed correlation between packet loss, latency, and signal strength, as we can see that the sites with stronger signal were associated with less latency and less loss, while the site with the weakest power signal strength had higher average losses and higher latency, particularly during typical waking hours. The Testing IRFP does not define a goal for packet loss.

3.2.4 Network Changes

In mid-March we ran another longterm test at Site A about four weeks after the initial test. It appears that in the interim the dowstream throughput for the network had been throttled to a lower value

Site Site	Latency Mean (ms)	Latency Std. Dev. (ms)	Mean Percent Packet Loss	Percent Packet Loss Std. Dev
A	73.908	48.346	1.562%	4.789%
B	28.601	47.754	2.549%	7.418%
C	95.320	139.87	33.031%	28.983%

Table 6: Summary of Round-trip Latency and Packet Loss

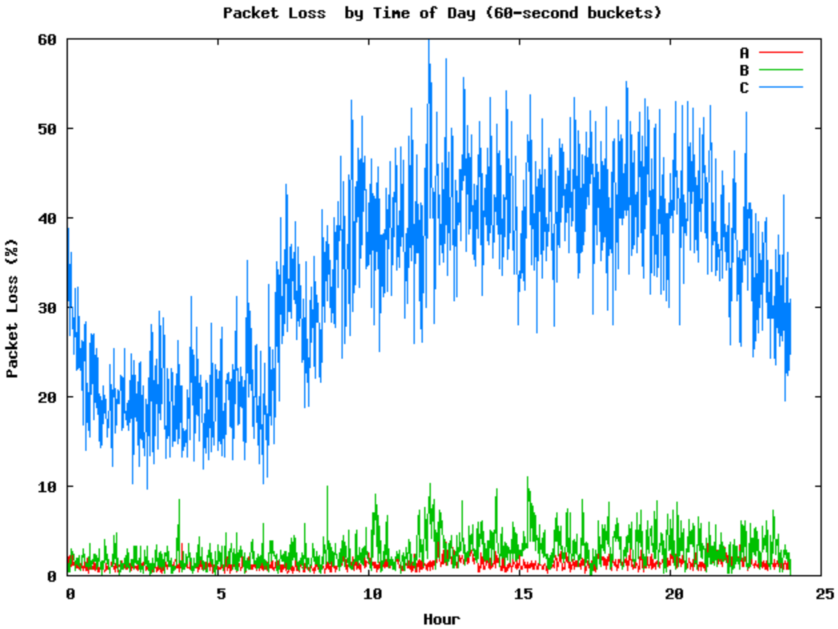

Figure 8: Packet Loss by Time of Day at Longterm Test Sites

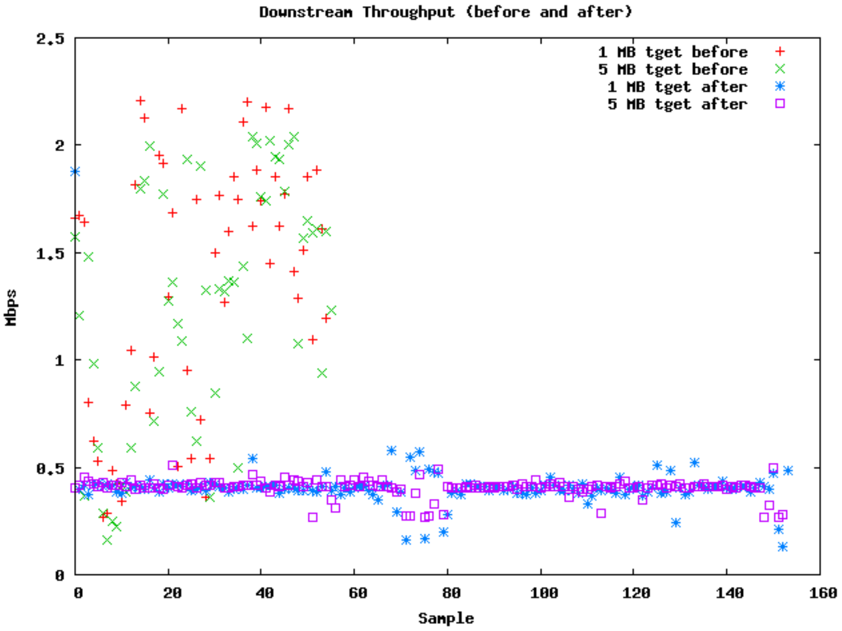

Figure 9: Observed Downstream Throughput at Site A during first runs, and then during later run.

than before, around 500 Kbps (although, more analysis is needed to verify this is true for the entire network). Figure 9 shows a comparison of the first data set from this site with the later data set. This new data reduces the average throughput from around 1.5Mbps to around 0.6Mbps. We include all data in our above analysis, but we think it is important to note the obvious change in performance.

4 Usable Coverage

This test evaluates "[t]he network's ability to provide a connection to at least 90% of the outdoor POC area" as specified in the Testing IRFP's section II(D).

4.1 Methodology

We address this evaluation by simple random sampling. The general approach was to identify the POC areas, select a random sample of locations in outdoor areas within the POC areas (excluding locations within buildings), and then measure the ability to make a connection at those locations.

4.1.1 Identification of POC Areas

The various documents published by the City did not indicate precisely what locations were considered to be within the POC network areas to be tested. The Testing IRFP provided a list of MetroFi access points, which we augmented and refined in our mapping effort. In the absence of specific documentary guidance from the City, we initially relied on the MetroFi coverage map[12], which shows green areas suggesting a coverage claim extending 1000 feet from each eastside access point location. The MetroFi coverage map is less expansive in its claims on the westside, where the green coverage areas tend to extend along streets. In preparation for our random sample, we utilized the 1000-foot claims of the MetroFi coverage map (in Figure 2 the orange areas indicate 1000-foot radii). It was not until an email correspondance with Logan Kleier, and his specific reply on March 20, 2007, that we learned MetroFi's POC areas to be considered extended only 500 feet from each access point (in Figure 2 the blue areas indicate 500-foot radii). The areas enclosed by the 500-foot radii added up to approximately 1.6 square miles.

4.1.2 Random Sample

From the list of 72 MetroFi access points that we considered to be in the POC network, we constructed a bounding box in latitude and longitude[13] extending 1000 feet beyond the extremities of the access point locations. The bounding box ranged from 45.512039 to 45.534452 degrees North and from 122.684071 to 122.637330 degrees West. Because we expected that many locations in the bounding box would fall outside of the POC areas, and because we were not certain how many locations we would be able to measure, we computed an excessive sample of 1001 locations using a random number generator such that each location in the bounding box had an equal probability of being chosen (see Appendix C).

The locations were numbered from 1 to 1001, and the distance to the nearest access point was computed for each location[14]. Locations not within 1000 feet of an access point were immediately excluded. Each remaining location was plotted against orthoimagery using Google Maps. If the location fell in the Willamette River, was inside a building, or was not practically reachable, it was also excluded.

Ultimately, the first 250 locations in our sample of 1001 were either excluded on the basis of the criteria above or were visited and measured (see Figure 11)[15].

4.1.3 Test Gear

To determine whether a connection was available at a selected location, we required a wireless client device. Again, for its utility as a general purpose embedded platform, we chose a Netgear WGT634U[16] running OpenWrt[17] software and custom measurement scripts. As we did for the longterm tests, we used the stock Atheros 5213 radio and antenna. The WGT634U was mounted on a six-foot length

[12] `http://www.metrofi.com/content/city/maps/portland_downtown_mapHRv2.jpg`

[13] All latitude/longitude coordinates in this report are with respect to the WGS84 ellipsoid, unless otherwise noted.

[14] The distances were computed by assuming an average height above the WGS84 ellipsoid of 40 meters, converting the latitude, longitude and heights to x, y, and z coordinates and then computing the cartesian distance between the two points.

[15] On March 28th, 2007, we presented preliminary results to the public. Those results were based on analyses that included some locations numbered above 250, which had been sampled on a convenience basis early in our measurements. Also, at that time some of the locations numbered less than 250 had not been measured. Now that a completed set of locations numbered less than 250 is available, the higher numbered locations have been excluded.

[16] `http://kbserver.netgear.com/products/WGT634U.asp`

[17] `http://www.openwrt.org`

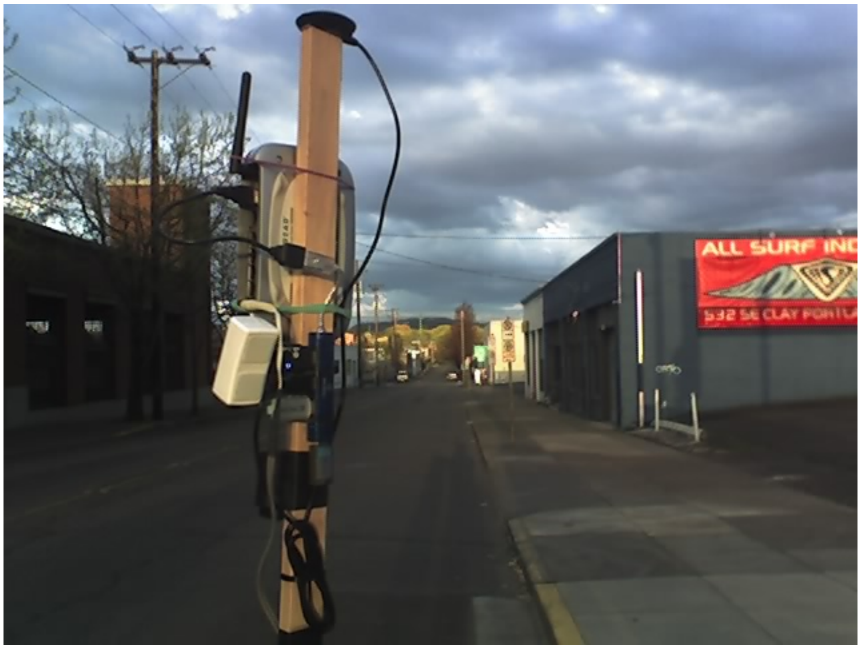

Figure 10: Random-point Connectivity test-rig

of 1x2-inch dimensioned hemlock lumber, which in turn was bound to a 10x10-inch square cast-iron tamper, providing a stable base to hold it vertical without assistance (see Figure 10). Also mounted were a USB hub, USB GPS, storage and audio devices, and a lithium-ion battery. The USB audio device provided feedback on the progress of each location test.

As mentioned previously, networking requires communication both from access point to client device and from client device to access point. If the access point has a powerful transmitter then it can be heard from longer distances, but if the client device does not have a similar transmit power then information cannot make the necessary round trip at those longer distances. Client devices with weaker radios will fail to achieve a connection where a more powerful radio might succeed. Therefore, a central question for testing usable coverage is the transmit power and antenna gain of the client device. We found guidance on this question in the initial Unwire Portland RFP, in sections 2.2.1.2 and 2.2.1.5, which state:

> “Coverage: Any individual using a device containing an integrated /built-in IEEE 802.11b/g (MiniPCI) radio or non-integrated/external card (PCMCIA, USB, etc.) shall be able to receive acceptable signal and mostly uninterrupted service in all outdoor Coverage Areas, [...]”

and

> “Device Support: Any individual using a device containing an integrated/built-in IEEE 802.11b/g (Mini PCI) radio or non-integrated/external card (PCMCIA, USB, etc.) shall be able to receive acceptable signal and mostly uninterrupted service in all outdoor areas of the Wi-Fi Tier.”

From those clauses of the RFP, we reasonably inferred that the stakeholders intended that outdoor coverage was expected for common denominator client devices. While some laptops have higher powered radios, in our experience typical client devices have a transmit power of 30mW. While laptops with integrated wireless may have higher gain antennas, the integrated antennas on a typical PCMCIA card or USB radio typically do not. As such, we concluded that the WGT634U device with a transmit power of 30mW and a stock antenna gain of 2 dBi would provide a reasonable, if potentially optimistic, surrogate for a typical client device.

4.1.4 Field Protocol for Placing Random Location

The random locations not excluded and numbered less than 250 in our list were plotted on a computerized map[18] (tied to a GPS device) and/or loaded into a handheld GPS device. Typically, we used the computerized map to navigate a car nearby, then we used the handheld GPS device to locate the target point on foot as close as we could. Sometimes the GPS put us right on our spot, with approximately a meter or two of uncertainty. In some cases our uncertainty was as much as 4 meters. In the downtown area, we found that often GPS did not function well due to multi-path reflections off tall buildings. In those cases we relied on our memory of the orthoimagery to find the closest spot.

In some cases we were unable to reach the exact location, either because it was in the middle of a busy street, or because it was on inaccessible private property. In these cases, we selected a nearby location with approximately the same distance to the access point and approximately the same degree of line of sight. In a few cases we excluded locations that were inaccessible and had no reasonable nearby surrogate location, particularly those along the Sullivan's Gulch railroad right-of-way.

4.1.5 Random Location Test Sequence

Once located at the spot, we activated our test device by inserting a USB dongle[19], acting as an enable key, into the USB hub. Our script detected the presence of the enable key and began its tests. The random location tests are very similar those used for longterm tests: we use ttcp to test upstream throughput; ICMP ping to test latency and loss; and wget to test downstream throughput. We used the logic of our "keep connected" subscript to bypass advertisement traps. We found it necessary to use several watchdog scripts: to check for a lost association, to check for GPS issues, and to check for stalled tests (for example, ttcp has a habit of taking a very long time on unstable connections). Depending on the results, a random location test might take anywhere from about 60 seconds (the length of time we would wait for an association) to around 7 minutes. The outline of the testing algorithm follows:

- Disassociate;
- Try to associate to "MetroFi-Free" for 60 seconds;

[18] `http://www.ostertag.name/gpsdrive.de/`

[19] The USB dongle used was an inactive bluetooth device. To assess the possibility of interference (Bluetooth also uses the 2.4 GHz band) we tested the device and found that it did not emit detectably when inserted in our test gear and therefore was not a significant source of interference.

- Record information about the physical layer (BSSID, Signal, etc.);
- Try to obtain a DHCP lease by sending up to 10 DHCP requests;
- Make sure we can pass traffic, otherwise bypass captive portal;
- Test latency and loss using ping;
- Test downstream throughput with a 1MB file, and a 5MB file;
- Test upstream throughput using ttcp;
- Grab the contents of the ARP table;
- Grab some statistics about our test device (memory and CPU utilization, etc); and
- Perform a traceroute to an internet host to record routing topology.

In addition to these steps, we also record GPS position and timestamp throughout the test.

The results of each test were stored on the USB storage device. At the conclusion of the tests we retrieved and analyzed the results. In our analysis we categorized each visited location according the following list:

1. Couldn't associate
2. Lost association mid-test
3. Couldn't get a DHCP lease
4. Couldn't pass traffic
5. Performance below specified (the IRFP defines a connection as a 64Kbps symmetric bandwidth)
6. Success

4.2 Results

Of the first 250 random locations in our list, shown in Figure 11, 22 were located in the Willamette River (white), 78 more were more than 1000 feet (red) from the nearest access points listed in Appendix A, 36 were located in buildings (pink), and we judged 6 inaccessible with no nearby surrogate locations (orange). Between March 25 and April 4, 2007, the remaining 108 accessible random locations (green) were measured using the protocol described above. Of the 108 measured locations, 53 were within 500 feet of an access point, the definition used by the City as the POC area.

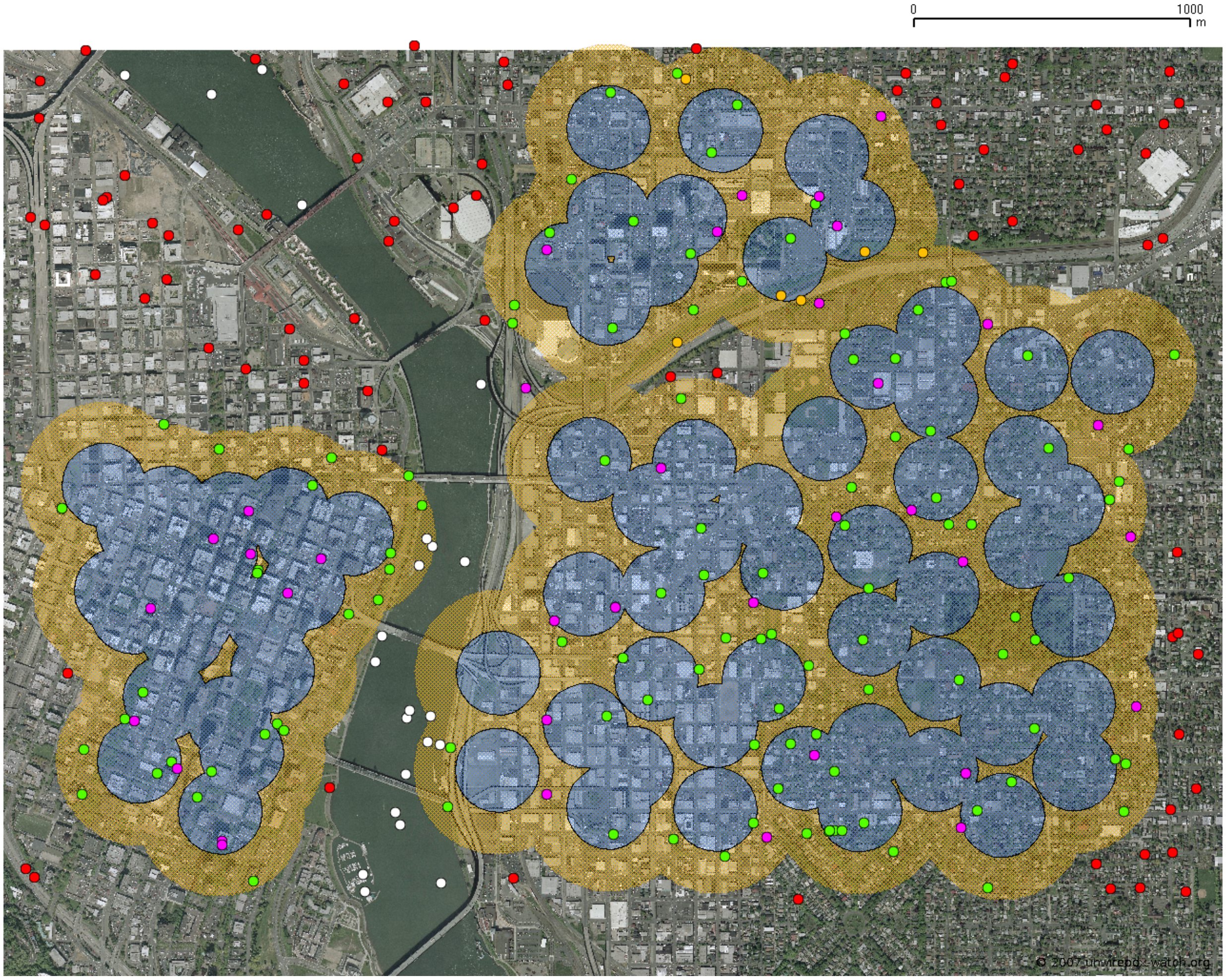

Figure 11: Two Hundred and Fifty Random Locations

Area	Number of Samples (n)	Successes (r)	Success Rate
within 250'	9	8	89%
within 500'	53	31	58%
within 1000'	108	50	46%
between 500 and 1000'	55	19	35%

Table 7: Success Rate by Area Measured

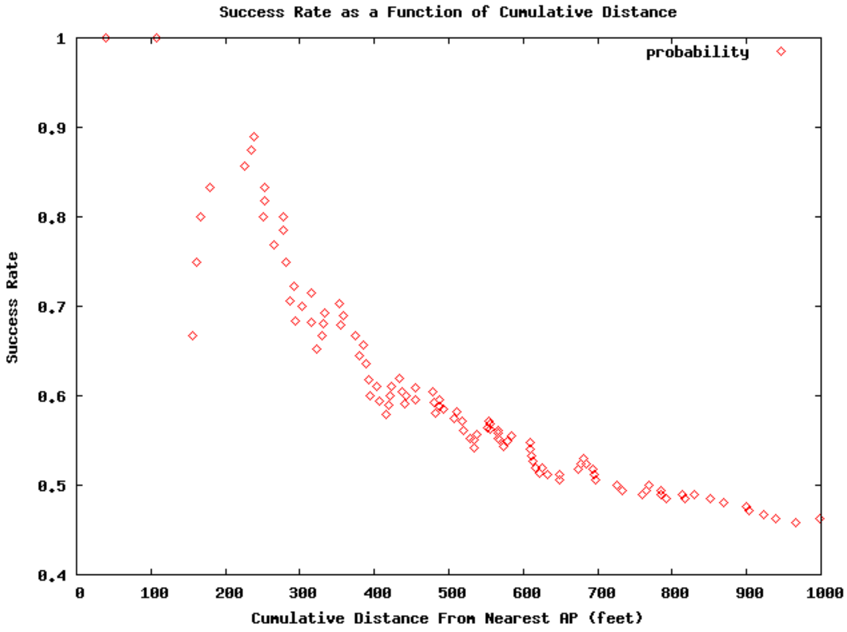

Figure 12: Success Rate as a Function of Cumulative Distance from the Nearest AP

4.2.1 Success Rate

Of the 53 locations within 500 feet, 31 succeeded (category 6 from the list above), about 58%, and 22 failed (categories 1 through 5). Table 7 presents the success rates within various ranges from the nearest access point[20].

In Figure 12 we plot an estimate of the rate of success within areas with increasing distances from the nearest access point. The estimated success rate is simply the number of successes divided by the number of locations within that distance of an access point.

[20] In the areas between 500 and 1000 feet "from the nearest acccess point", we discovered that some access points had been added that we had not accounted for and that the locations may have actually been closer to an active access point than we had believed. Since this area was not our primary focus, we have not attempted to correct for this. The success rate in this distance range is therefore an overestimation. This is discussed in more detail in Section 5.7.

4.2.2 Likelihood that Outdoor Coverage Meets 90% Threshold

To assess whether the POC network provided a connection in 90% of outdoor areas, we applied a statistical test. The random sample of outdoor locations can be thought of as a Bernoulli process, a series of independent dichotomous trials, where each trial is labelled "success" or "failure". We assume as the null hypothesis that the POC network is providing the ability to make a connection in 90% of outdoor areas. We know from basic statistics that in a sampling from a Bernoulli process with a probability of success equal to p and probability of failure equal to $q = 1 - p$, the probability of observing exactly r successes in n independent trials is a binomial distribution of the form:

$$P = \binom{n}{r} p^r q^{n-r} \tag{1}$$

If we consider the likelihood that a given number of successes or fewer occur in n trials by chance, we sum these probabilities:

$$\sum_{i=0}^{n-r} \binom{n}{i} p^i q^{n-i} \tag{2}$$

Applying this summation to 31 successes in the 53 random location trials ($p = 9/10, r = 31, n = 53$), we find that the probability under the hypothesis that the POC network can provide connections to our client device in 90% of outdoor areas within 500 feet is 2.071×10^{-9}, or about 2 chances in a billion.

With that probability, it is reasonable to conclude that the null hypothesis is false. That is, the POC network cannot provide connections to our client device in 90% of outdoor areas within 500 feet of an access point.

As shown in Figure 13 we repeated these calculations with increasing distance from the nearest access point. By 1000 feet, the probability that the observed number of successes or fewer would occur by chance is in the neighborhood of 10^{-29}.

4.2.3 Performance at Succeeding Random Locations

For those tests that were successful, we collected information about the performance of the network. Averaging across the random sample allows us to provide an "average view" of network performance for those locations with a usable connection; these statistics are summarized in Table 8.

Because MetroFi continued to turn on access points between the time of our network mapping in late February and our measurements in late March and early April, several of the locations with successful connections were actually closer than to active access points than we thought. This issue is discussed in more detail in Section 5.7. As a consequence, the values for locations between 500 and 1000 feet should be treated with caution.

Section II(C) of the testing IRFP requires that the network can deliver 1Mbps downstream and 256Kbps upstream throughput "at a distance of up to 250 feet from an access point". Table 8 shows that the average rates for those points below 250', meets those requirements. And of those 7 points, 5 (71%) pass this test.

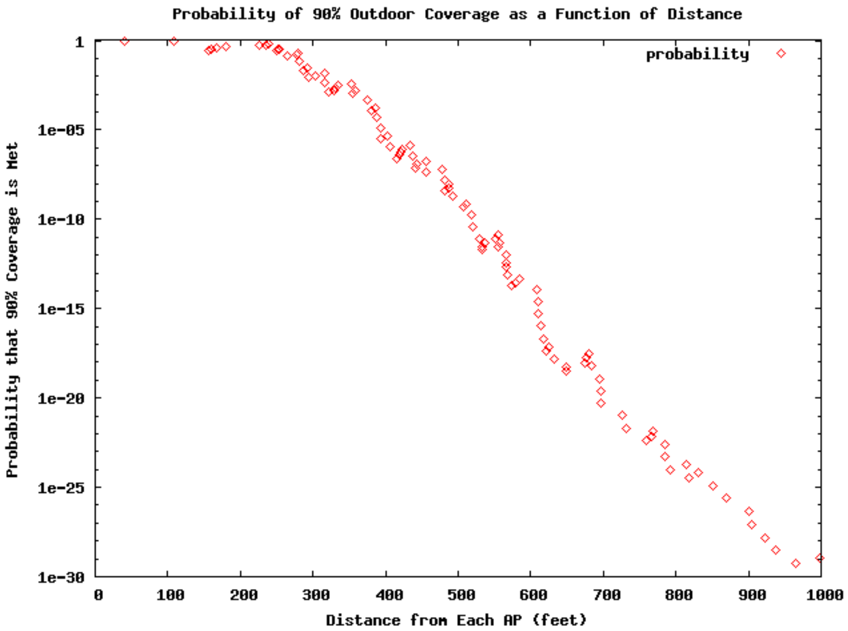

Figure 13: Probability of Finding the Number of Successes or Fewer Assuming the Network Provides Connections in 90% of Outdoor Areas by Cumulative Distance from the Nearest AP

Area	N	Down/Up Throughput (Kbps)		Latency (ms)		Loss (%)		Signal (dBm)	
		Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
< 250'	7	1672.4/373.28	1281.3/83.235	95.700	72.319	5.7143	15.119	-50.857	3.7607
< 500'	16	1508.7/373.42	1002.8/79.181	105.15	69.808	3.125	10.145	-57.938	8.4417
< 1000'	27	1437.2/370.52	875.72/74.682	97.459	59.344	3.33	8.77	-59.333	8.6425
> 500'	11	1333.1/366.30	657.23/71.159	86.273	40.182	3.6364	6.7420	-61.364	8.9249

Table 8: Random Sample Performance Summary

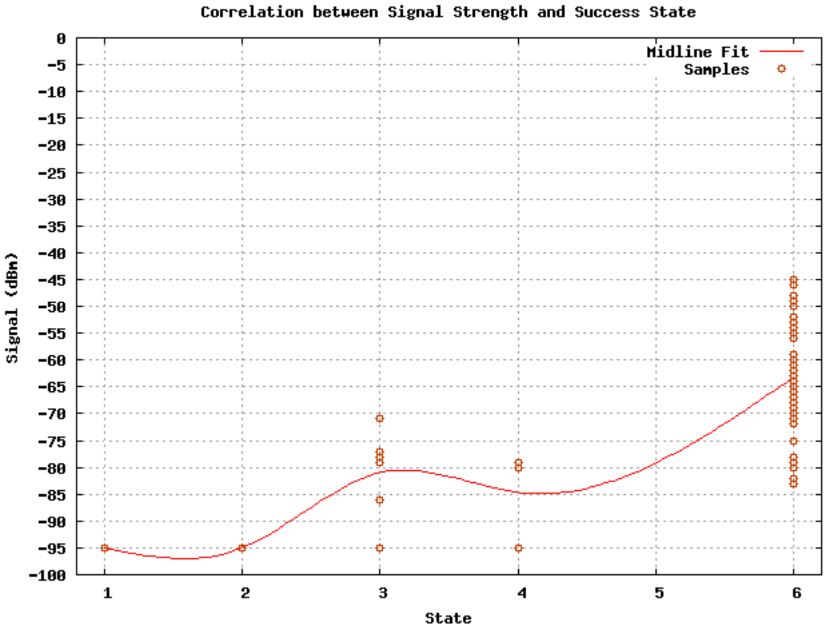

Figure 14: Correlation Between Signal Strength and Success Category

In order to understand the possible cause of failures, we looked at the correlation between several variables and success. The metric which most closely explains success rate is signal strength, which has a correlation coefficient of 0.903. Distance to AP, on the other hand, does not have a clear linear correlation, with a correlation coefficient of -0.306. Figure 14 plots signal strength against success category.

For "success" (category 6), the mean signal strength is -63.2 dBm, the maximum passing signal strength is -45 dBm, and the minimum passing signal strength is -83 dBm. In cases where we did not associate, we have stored no indication of signal strength, so the data here are truncated. However, of those we did record, the mean "failing" (category 1–5) signal is -78.6 dBm and the maximum failing signal is -71 dBm. We can estimate from this, that a signal of -71 dBm to -45 dBm are probably a pass, a signal of -83 dBm to -71 dBm may be a pass or fail, and anything less than -83 dBm is almost certainly a failure.

5 Miscellaneous Observations

5.1 Network Topology

Network addresses are assigned to clients using DHCP in the 10.100.24.0/21 private address range. The default route is set to 10.100.24.1, which our ARP tables universally observed to have a MAC address of 00:30:F7:74:F6:E2. When gathering traceroutes, the initial three hop replies were recorded from 10.100.13.2, 10.100.13.1 and 10.100.14.2, respectively. Please see appendix D for a full listing of traceroute output to two well-connected Internet hosts.

5.2 Private IP Addresses

Since users of the network are not given a public IP, their address is translated into a common public IP address at a gateway to the Internet. The consequence of the network address translation (NAT) is that users of the network cannot provide services to the Internet. While NAT is common at hotspots, it should be recognized that it means that users of the MetroFi networks are limited to being consumers and prevents them from being providers on the Internet.

5.3 Client-to-Client Filtering

It was also discovered that two MetroFi users cannot connect to each other over the network. This policy is apparently intended[21] to limit the spread of viruses and other malware; however, it also limits network users to being consumers and prevents them from being providers on the MetroFi network. Opportunities for innovative uses of the municipal area network are foreclosed by this measure.

5.4 Changes in Throughput Noted

As noted above in Section 3.2.4, during the latter part of March we noted a decrease in the bandwidth at longterm test Site A (70 feet from an access point). Whereas early in the testing period we measured download speeds sometimes in excess of 2 Mbps, in the period from March 18th to March 25th we found download speeds consistently and substantially below the 1 Mbps standard called for in the contract. Given these changes, it would perhaps be prudent for the City to regularly test the network in order to verify that the contractual requirements are met.

5.5 Packet Loss and Interference

While packet loss is not included among the official testing metrics, some protocols are impacted significantly by excessive packet loss and we suggest that it be considered in similar future testing criteria.

One interesting observation is that packet loss at our noisiest longterm test site (Site C) seems to be a function of the time of day. One possible explanation is that during hours of peak usage on private networks, interference with the MetroFi network is at a maximum and packet loss results. As private and public network usage grows within the shared 2.4 GHz band, interference will likely remain an issue to watch.

[21] `http://www.wifinetnews.com/archives/007494.html`

5.6 Size of POC Area Smaller than Specified in the RFP

The original RFP for the project called for a POC area at least 2 square miles[22]. It should be noted that the POC area as defined was substantially less than that, on the order of 1.6 square miles. This seems to violate the requirements of the RFP.

5.7 Addition of Access Points During the Testing Phase

The Testing IRFP listed 67 access points. Our inquiries in mid-March to the City Project Manager about the detailed description of the POC area were consistently met with reference to this list. However, as we noted in our discussions above, we found additional access points in late February. Furthermore, when we performed our random location tests in late March and early April, we discovered that numerous additional access points (approximately a dozen we had not previously seen) had been activated. When we inquired about these, particularly whether they were allowed to be added during the testing phase, we were told that MetroFi was allowed to add access points within the "Proof-of-Concept Territory", a term of art we had not previously seen and which so far as we are aware has never been publicly defined. It is also notable that at least three of these new access points were at the margins and therefore extended the network rather than filled in holes. The new access points appear to only affect random locations between 500 and 1000 feet of the 72 previously known access points, and therefore they do not affect our conclusions for locations within 500 feet. Because we have not excluded the affected locations beyond 500 feet, the results reported in that range are biased in MetroFi's favor.

5.8 MetroFi Sets Its own Standard

On April 11, 2007, the City published a one-page executive summary of Uptown Services testing report and simultaneously released the Certificate of Acceptance, freeing MetroFi from any further obligation to improve their network. The City Project Manager justified this in a three-page memo[23] on the basis of a standard that had until then not been disclosed, a -79 dBm signal level. Further inquiries have revealed that MetroFi supplied this standard, claiming that it represented the ability to establish a connection at a 1 Mbps modulation rate. There is no indication that this standard was independently verified by anyone outside MetroFi. On the basis of the -79 dBm standard of received signal strength, passed along from MetroFi by the City, Uptown Services concluded from its drive testing that connections were possible in 95% of the outdoor areas.

It should be emphasized that our data, in contrast to those of Uptown Services, were based on actual attempts to establish a connection and that we found that connections were successfully established in less than 60% of the randomly selected locations we tested, contradicting the Uptown Services conclusions. Unless the standard used by the City and Uptown Services was verified as an accurate model for the client devices described in the original RFP, serious grounds for concern exist about the validity of their conclusion.

[22] Section 2.2.5.1, first bullet states: "Covers at least two (2) square miles with Wi-Fi coverage providing the bandwidth described in Sections 2.2.1.2 and 2.2.1.9."

[23] http://www.portlandonline.com/shared/cfm/image.cfm?id=154499

6 Conclusions

In summary, our principal findings in terms of the contract requirements as presented in the IRFP are:

Throughput

- Our longterm tests show that the network is capable of providing 1 Mbps downstream and 256 Kbps upstream from a stationary position. However, recent data indicates that the downstream bandwidth may be capped lower than 1 Mbps.

Availability

- The network is able to provide an association and a 64 Kbps/64 Kbps connection at least 99% of the time.
- The average round trip latency is under the required 200 ms.
- We found that 88% of outdoor locations within 250' are able to provide a connection (the contract requires at least one such connection).
- We found that the network **does not** provide a usable connection (64 Kbps/64 Kbps) to 90% of the POC network area. We found that the network provides a connection at about 58% of the outdoor locations within 500 feet.

We did not test the requirements for security or use with a signal booster.

A Proof-of-Concept Access Points

BSSID	Intersection	Latitude	Longitude
00:0A:DB:01:74:E0	NE Glisan NE 17th	45.526630	-122.648470
00:0A:DB:01:86:00	SW Alder SW 10th	45.520635	-122.681620
00:0A:DB:01:8F:60	SE Stark SE Sandy	45.519400	-122.658190
00:0A:DB:01:90:00	NE Holladay NE 7th	45.530000	-122.658470
00:0A:DB:01:91:20	SE 26th SE Washington	45.518540	-122.639210
00:0A:DB:01:91:40	SW 11th SW Yamhill	45.519459	-122.683516
00:0A:DB:01:92:E0	SE Taylor SE MLK BLVD	45.515010	-122.661812
00:0A:DB:01:93:00	SE 12th SE belmont	45.516480	-122.653550
00:0A:DB:01:98:00	SE 17th SE pine	45.520780	-122.648590
00:0A:DB:01:98:60	SE Water SE Belmont	45.516741	-122.665840
00:0A:DB:01:9A:E0	SE Belmont SE 7th	45.516523	-122.658518
00:0A:DB:01:9C:80	se grand se hawthorne	45.512358	-122.660763
00:0A:DB:01:9D:80	SW Taylor SW Broadway	45.518074	-122.680485
00:0A:DB:01:9D:A0	SW 3rd SW Washington	45.519513	-122.674858
00:0A:DB:01:9E:40	SE MLK Blvd SE Stark	45.519288	-122.661788
00:0A:DB:01:9E:C0	SW 4th SW Oak	45.521282	-122.675070
00:0A:DB:01:9E:E0	SE 13th SE Oak	45.520109	-122.652684
00:0A:DB:01:9F:40	SW Taylor SW 10th	45.518767	-122.682765
00:0A:DB:01:A0:60	SW Stark SW 10th	45.522150	-122.681129
00:0A:DB:01:A5:80	NE Glisan NE 24th	45.526553	-122.641317
00:0A:DB:01:A7:20	SE 12th SE Ankeny	45.522142	-122.653666
00:0A:DB:01:AA:60	SW 3rd SW Taylor	45.516839	-122.676293
00:0A:DB:01:AE:20	SW Broadway SW Oak	45.521937	-122.678219
00:0A:DB:01:B4:20	SW 11th SW Washington	45.521563	-122.682510
00:0A:DB:01:B5:C0	SW 4th SW Morrison	45.518617	-122.676510
00:0A:DB:01:BB:60	SE Belmont SE 20th	45.516474	-122.645445
00:0A:DB:01:C5:00	NE Couch NE 9th	45.523595	-122.656718
00:0A:DB:01:C5:80	NE Holladay NE 3rd	45.529991	-122.662590
00:0A:DB:01:C6:00	NE Davis NE Sandy	45.524310	-122.650690
00:0A:DB:01:C6:40	SE Hawthorne SE 23rd	45.512039	-122.642529
00:0A:DB:01:C7:20	SE Taylor SE 10th	45.515010	-122.655790
00:0A:DB:01:C7:A0	SE Ankeny SE 7th	45.522180	-122.658610
00:0A:DB:01:C9:E0	SW Market SW 3rd	45.512253	-122.678747
00:0A:DB:01:CC:00	NE Grand NE Oregon	45.528731	-122.660685
00:0A:DB:01:D8:40	NE Glisan NE 28th	45.526430	-122.637330
00:0A:DB:01:E7:C0	NE 13th NE Holladay	45.530160	-122.652500
00:0A:DB:01:E8:20	SW Broadway SW Clay	45.513922	-122.682483
00:0A:DB:01:E8:40	NE Weidler NE 10th	45.534343	-122.655447
00:0A:DB:01:E8:80	W Burnside SW 13th	45.522888	-122.684071
00:0A:DB:01:EA:80	NE Couch NE 24th	45.523670	-122.641370
00:0A:DB:01:ED:A0	SW Pine SW 5th	45.522080	-122.675625
00:0A:DB:01:EE:80	NE 15th NE Halsey	45.533480	-122.650520
00:0A:DB:01:F1:80	NE Grand NE Weidler	45.534452	-122.660634
00:0A:DB:03:28:40	SW 3rd SW Jefferson	45.514177	-122.677749

BSSID	Intersection	Latitude	Longitude
00:0A:DB:03:32:60	SE Main SE 6th	45.513630	-122.659690
00:0A:DB:03:34:20	NE Multnomah NE 9th	45.531547	-122.657107
00:0A:DB:03:41:00	SE 20th SE Stark	45.519252	-122.645490
00:0A:DB:03:46:40	SE Water SE Main	45.513585	-122.665905
00:0A:DB:03:47:40	SE Hawthorne SE 10th	45.512277	-122.655828
00:0A:DB:03:47:C0	NE Multnomah NE 16th	45.531425	-122.649321
00:0A:DB:03:48:80	SW Madison SW Broadway	45.515971	-122.681268
00:0A:DB:03:49:20	SE 20th E Burnside	45.523010	-122.645550
00:0A:DB:03:49:40	SW Broadway SW Morrison	45.519269	-122.679662
00:0A:DB:03:4C:E0	SE Yamhill SE 26th	45.515690	-122.639220
00:0A:DB:03:59:A0	SE 23rd SE Taylor	45.515003	-122.642548
00:0A:DB:03:5E:20	SE 24th SE Pine	45.520769	-122.641368
00:0A:DB:03:7B:40	NE 20th NE Flanders St	45.525651	-122.645517
00:0A:DB:03:7B:80	SW Broadway SW Yamhill	45.518598	-122.680024
00:0A:DB:03:7B:C0	SW Broadway SW Washington	45.520598	-122.678946
00:0A:DB:03:7C:C0	SE 20th SE Main	45.513538	-122.645598
00:0A:DB:03:7C:E0	SE 14th SE Main	45.513595	-122.651724
00:0A:DB:03:7F:A0	NE Grand NE Multnomah	45.531620	-122.660648
00:0A:DB:03:80:60	SE 17th SE Alder	45.517960	-122.648650
00:0A:DB:03:81:A0	NE Irving NE 20th	45.527861	-122.645391
00:0A:DB:03:81:E0	NE Couch NE MLK Blvd.	45.523660	-122.661600
00:0A:DB:03:82:40	SW 4th SW Madison	45.515309	-122.678321
00:0A:DB:03:C0:40	SE Sandy SE 9th	45.521000	-122.656470
00:0A:DB:03:C3:A0	SW Pine SW 3rd (2nd)	45.521283	-122.672613
00:0A:DB:03:C4:C0	SE 26th SE Main	45.513546	-122.639182
00:0A:DB:03:C6:60	SE Hawthorne SE 16th	45.512243	-122.649117
—NA—	SE 17th SE Salmon	45.514320	-122.648710
—NA—	SE Ankeny SE 26th	45.522129	-122.639681

B Longterm Test Scripts

B.1 run.sh

```
#!/bin/ash
#
# Starts everything running...

# Things worth configuring...
TTCP_PORT=443
TTCP_HOST=nishita.cs.pdx.edu
TTCP_SLEEP=5m
PING_HOST=nishita.cs.pdx.edu
PING_SLEEP=1s
ATH0_SLEEP=1m
SYS_SLEEP=1m
KEEP_CONNECTED_SLEEP=1m

if [ $((‘date +%Y‘)) -lt 2007 ];then
        # TODO: check the return code of ntpclient
        # and exit if we can’t set the clock
        ntpclient -h us.pool.ntp.org -s
fi
dir=‘date +%Y_%m_%d_%H_%M_%S‘
mkdir ../$dir
./keep_connected.sh ../$dir $KEEP_CONNECTED_SLEEP > ../$dir/keep_connected_script.log &
./ping_test.sh ../$dir $PING_HOST $PING_SLEEP > ../$dir/ping_test.log &
./ttcp_test.sh ../$dir $TTCP_HOST $TTCP_PORT $TTCP_SLEEP > ../$dir/ttcp_test.log &
./ath0_test.sh ../$dir $ATH0_SLEEP > ../$dir/ath0_test.log &
./sys_test.sh ../$dir $SYS_SLEEP > ../$dir/sys_test.log &
```

B.2 keep_connected.sh

```
#!/bin/ash
#
# Try to keep connected by going to the meta redirect
# that appears after we haven’t passed port-80 traffic
# for a bit.

dir=$1
sleep=$2
if [ ! ${sleep} ];then
        sleep=1m
fi

PINGDEST=google.com
```

```
WGETDEST=http://www.google.com

while true;do
  ping -c 1 -q ${PINGDEST} > /dev/null 2>&1
  # if we can’t pass traffic
  if [ $? -eq 1 ];then
     wget -O - ${WGETDEST} > ${dir}/1.html
     if [ -f ${dir}/1.html ];then
       # extract URL from redirect
       url=$(head -n 1 ${dir}/1.html | sed ’s/^.*URL=\(.*\)">$/\1/g’)
       if echo ${url} | grep -q ^http ; then
           wget -O - "$url" > ${dir}/2.html
           echo "$(date +%s) ${url}" >> ${dir}/keep_connected.log
       else
           echo "$(date +%s) ${WGETDEST}" >> ${dir}/keep_connected.log
       fi
       sync
     else
       echo "Failed to grab redir page into 1.html"
     fi
  fi
  echo "Sleeping"
  sleep ${sleep}
done
```

B.3 ping_test.sh

```
#!/bin/ash

dir=$1
host=$2
sleep=$3
if [ ! $sleep ];then
        sleep=1s
fi
while true;do
        date +%s >> $dir/ping.log
        ping -c 10 $host >> $dir/ping.log 2>&1
        sync
        sleep 1s
done
```

B.4 ttcp_test.sh

```
#!/bin/ash

```

```
dir=$1
host=$2
port=$3
sleep=$4
five=http://nishita.cs.pdx.edu/~calebp/five_meg
one=http://nishita.cs.pdx.edu/~calebp/one_meg
if [ ! $sleep ];then
        slee="5m"
fi
echo "Starting TTCP test to $host on $port sleeping $sleep" > $dir/ttcp.log
while true;do
        # tput test
        date +%s >> $dir/ttcp.log
        ttcp -t -s -v -p $port $host >> $dir/ttcp.log 2>&1
        sync
        sleep $sleep

        # 1MB tget test
        ts=‘date +%s‘
        one_dl_time=‘time wget -O $dir/one -q $one 2>&1 | grep real‘
        echo "tget 1 $one_dl_time at $ts" >> $dir/ttcp.log
        rm -f $dir/one
        sync
        sleep $sleep

        # 5MB tget test
        ts=‘date +%s‘
        five_dl_time=‘time wget -O $dir/five -q $five 2>&1 | grep real‘
        echo "tget 5 $five_dl_time at $ts" >> $dir/ttcp.log
        rm -f $dir/five
        sync
        sleep $sleep
done
```

B.5 atho_test.sh

```
#!/bin/ash

dir=$1
sleep=$2
if [ ! $sleep ];then
        sleep=1m
fi
while true;do
        date +%s >> $dir/ath0.log
        iwconfig ath0 >> $dir/ath0.log 2>&1
        sync
        sleep $sleep
```

```
done
```

B.6 sys_test.sh

```
#!/bin/ash
#
# Fields in logfile are:
#  timestamp
#  used memory bytes
#  free memory bytes
#  runqueue 1 min avg
#  runqueue 5 min avg
#  runqueue 15 min avg
#  processors/schedulers
#  last pid

dir=$1
sleep=$2
if [ ! $sleep ];then
        sleep=1m
fi
while true;do
  ts=‘date +%s‘
  mem=‘free | grep Mem | awk ’{print $3," ",$4}’‘
  cpu=‘cat /proc/loadavg‘
  echo "$ts $mem $cpu" >> $dir/sys.log
  sync
  sleep $sleep
done
```

C Computation of Random Locations

We assumed a spherical approximation for the earth, which given the scale of the area considered in this evaluation, seemed a reasonable one:

```
(defun random-locations (n lat-min lat-max long-min long-max)
  (let* ((la1 (deg-to-rad lat-min))
         (la2 (deg-to-rad lat-max))
         (lo1 (deg-to-rad long-min))
         (lo2 (deg-to-rad long-max))
         (dlo (- lo2 lo1))
         (z1 (sin la1))
         (z2 (sin la2))
         (dz (- z2 z1)))
    (loop for i from 0 to n
          collect (let* ((ra1 (random dz))
                         (ra2 (random dlo))
                         (z (+ ra1 z1))
                         (lat (asin z))
                         (long (+ ra2 lo1)))
                    (list (rad-to-deg lat)
                          (rad-to-deg long))))))

;; (random-locations 1000 45.512039 45.534452 -122.684071 -122.637330)
```

D Traceroutes

From the Unix traceroute manual page:

> The Internet is a large and complex aggregation of network hardware, connected together by gateways. Tracking the route one's packets follow low (or finding the miscreant gateway that is discarding your packets) can be difficult. Traceroute utilizes the IP protocol "time to live" field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to some host.

D.1 To pdx.edu

```
traceroute to pdx.edu (131.252.120.50), 30 hops max, 38 byte packets
 1  10.100.13.2 (10.100.13.2)  22.390 ms  37.484 ms  21.399 ms
 2  10.100.13.1 (10.100.13.1)  31.170 ms  25.225 ms  29.813 ms
 3  10.100.14.2 (10.100.14.2)  17.178 ms  33.567 ms  28.732 ms
 4  fa0-1.na01.b006468-1.pdx01.atlas.cogentco.com (38.102.192.101)  30.412 ms  16.383 ms  27.781 ms
 5  g3-1.core01.pdx01.atlas.cogentco.com (38.112.37.217)  29.640 ms  24.783 ms  17.555 ms
 6  p2-0.core01.smf01.atlas.cogentco.com (154.54.3.125)  32.108 ms  40.766 ms  35.786 ms
 7  p4-0.core02.sfo01.atlas.cogentco.com (154.54.1.253)  39.995 ms  28.998 ms  35.498 ms
 8  t3-3.mpd01.sfo01.atlas.cogentco.com (154.54.3.118)  55.022 ms  31.923 ms  28.774 ms
 9  t9-2.mpd01.sjc01.atlas.cogentco.com (154.54.2.126)  36.329 ms  35.419 ms  49.027 ms
10  v3499.mpd01.sjc03.atlas.cogentco.com (154.54.6.238)  35.153 ms  35.564 ms  43.154 ms
11  te-3-3.car3.SanJose1.Level3.net (4.68.110.137)  58.042 ms  39.056 ms  48.902 ms
12  ae-2-54.bbr2.SanJose1.Level3.net (4.68.123.97)  49.055 ms
    ae-2-56.bbr2.SanJose1.Level3.net (4.68.123.161)  63.132 ms
    ae-2-52.bbr2.SanJose1.Level3.net (4.68.123.33)  40.130 ms
13  ae-0-0.mp2.Seattle1.Level3.net (209.247.9.122)  53.862 ms  49.590 ms  51.721 ms
14  ge-11-1.hsa2.Seattle1.Level3.net (4.68.105.103)  53.145 ms
    ge-10-2.hsa2.Seattle1.Level3.net (4.68.105.135)  55.310 ms
    ge-10-0.hsa2.Seattle1.Level3.net (4.68.105.7)  57.070 ms
15  nero-gw.Level3.net (63.211.200.246)  54.710 ms  66.588 ms  64.678 ms
16  ptck-car1-gw.nero.net (207.98.64.139)  58.978 ms  62.038 ms  67.377 ms
17  * * *
18  131.252.1.10 (131.252.1.10)  40.762 ms  63.282 ms  46.710 ms
19  www.pdx.edu (131.252.120.50)  49.007 ms  48.333 ms  58.983 ms
20  www.pdx.edu (131.252.120.50)  41.027 ms  53.908 ms  45.838 ms
21  www.pdx.edu (131.252.120.50)  49.340 ms  52.616 ms  47.155 ms
```

D.2 To google.com

```
traceroute to google.com (64.233.187.99), 30 hops max, 38 byte packets
 1  10.100.13.2 (10.100.13.2)  20.101 ms  22.008 ms  25.686 ms
 2  10.100.13.1 (10.100.13.1)  24.816 ms  31.597 ms  26.597 ms
 3  10.100.14.2 (10.100.14.2)  24.059 ms  22.833 ms  24.688 ms
 4  fa0-1.na01.b006468-1.pdx01.atlas.cogentco.com (38.102.192.101)  58.074 ms  20.293 ms  30.011 ms
 5  g3-1.core01.pdx01.atlas.cogentco.com (38.112.37.217)  18.063 ms  29.094 ms  24.142 ms
 6  p2-0.core01.smf01.atlas.cogentco.com (154.54.3.125)  31.817 ms  28.421 ms  29.864 ms
 7  p4-0.core02.sfo01.atlas.cogentco.com (154.54.1.253)  39.306 ms  43.009 ms  49.686 ms
 8  p6-0.core01.sjc04.atlas.cogentco.com (66.28.4.234)  51.581 ms  49.668 ms  35.773 ms
 9  p11-0.core01.sjc03.atlas.cogentco.com (154.54.3.42)  34.923 ms  42.092 ms  30.467 ms
10  google.sjc03.atlas.cogentco.com (154.54.10.254)  47.021 ms  33.546 ms  34.797 ms
11  66.249.94.2 (66.249.94.2)  46.174 ms  31.295 ms  46.673 ms
12  66.249.95.212 (66.249.95.212)  134.834 ms  100.648 ms  121.054 ms
13  jc-in-f99.google.com (64.233.187.99)  99.510 ms
    216.239.49.44 (216.239.49.44)  116.026 ms  116.132 ms
```