Self-Healing Home Networks Test Bed

Yuen Hsi Chang, Karen Halls, and Jonathan Knudson

Background

Ever since the use of the networking space has gotten widespread to home users, many problems regarding network connectivity and stability have surfaced. Due to the complex structure of the Internet, problems within the home network tend to be very difficult to troubleshoot and solve, and this is especially true for home users who do not have a proficient understanding of networks. The addition of a "self-healing" function would significantly ease network maintenance, as developing networks to independently recover from connection problems means that expert users would no longer be needed to aid the troubleshooting process. Ultimately, this would ease many difficulties in maintaining home networks, as human intervention would no longer be needed to preserve network stability.

Abstract

To further the research in self-healing home networks, we created a home network to collect data. This paper contains an explanation of the setup of our home network test bed, the measurement tools we put on it, the data we collected, an analysis and discussion of the data, and our conclusions with suggested future work.

Tool design

In order to be able to collect measurements, we designed a few tools. For all of the devices on the network, we were able to collect the ping time. This tool simply consisted of a script that would ping the devices and would then write the ping times to a text file. From these ping times, we were able to determine which devices were on the network at any given time. Our algorithm for deciding if the device was on the network was to see if the last ping (to that device) got a response, and if it did, then that device was determined to be on the network.

Another tool we created allowed us to retrieve a list of the programs that were running on the laptops. If a laptop was determined to be on the network, the command center remotely accessed the laptop (with SSH) to find out what programs were running at the time.

From the laptops, we were also able to collect bandwidth data. If a laptop was determined to be on the network, then the command center would remotely run a program on the laptop. The program determined the bandwidth by downloading a webpage and measuring how long that took.

We also developed a tool to measure bandwidth (as well as link speed and some other measurements) on the Android tablet. It was an application we put onto the tablet where it took measurements and then wrote those measurements to a file. The command center would then remotely access that file. However, this tool needs to be updated a bit. It turns out that it only runs once when the application is open, so in order to get updated measurements, one must constantly be opening and closing the app. This is not very practical in a home network situation, or in a situation where one is collecting data. Because of this flaw in the tool, we were not able to use the data that this tool collected.

The last tool that we used measured the quality of some specific flash videos and recorded the measurements in a repository. This tool was developed a couple of summers ago. We modified this code to instead write the measurements onto a local file on the command center. The web server for this tool was set up on the command center. When anyone went to watch a video on that website, the measurements, such as frames per second and audio bandwidth, were collected and stored on the command center. The difference with this tool is that the command center did not have control over when this data was collected – it was completely dependent on the user.

Network design and setup
Our network consisted of a modem and router independent of the college's main network that served as the internet source for our home network, an iMac desktop computer that served as our "command center," three laptops (two MacBooks and one PC), and an Android tablet.

The network router was an Asus product, and it supported a maximum bandwidth of 54 Mbps. We made the connection to the network private by using a WPA2 Personal password protection.

The purpose of the command center was to be the all-seeing hub that sent the network measurement requests to the other devices, stored the records of all the measurements in local folders, and performed analysis on the data.

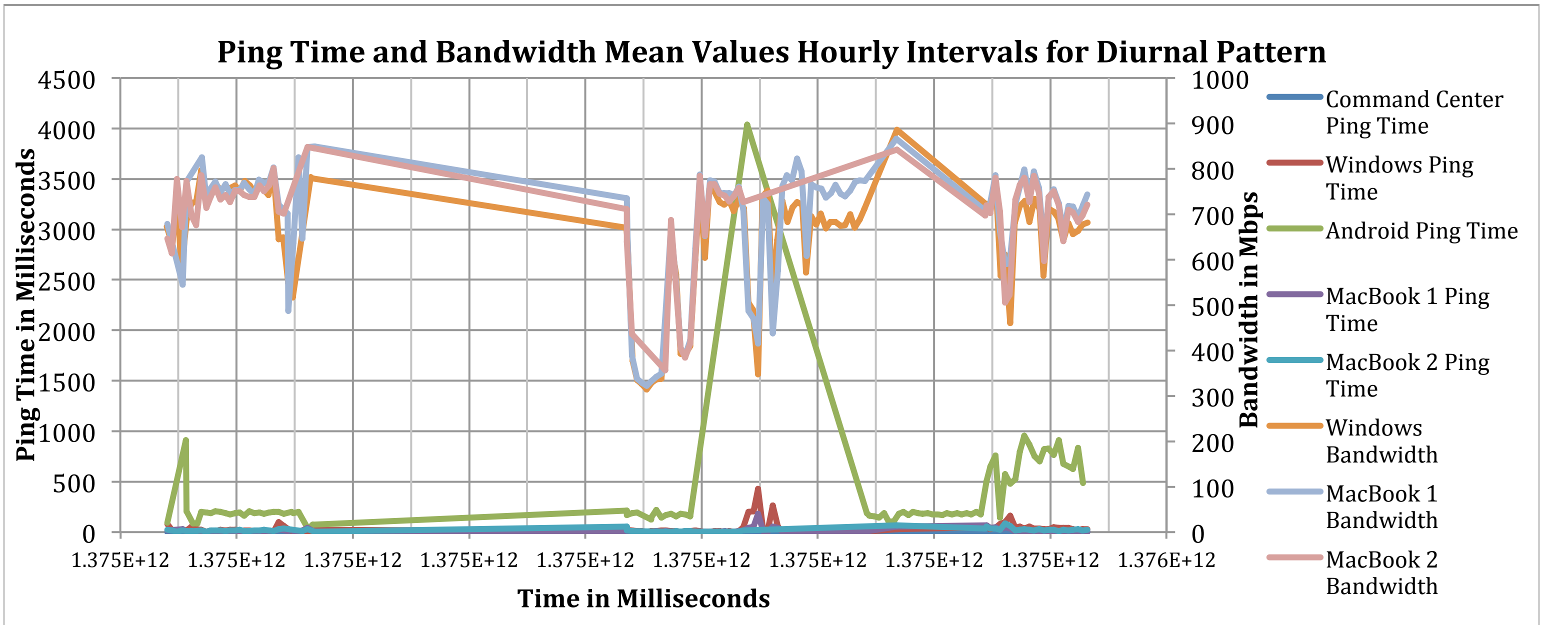
Each individual device contained a program that measured the current bandwidth. For the laptops, this program was remotely activated by the command center, and then the command center retrieved the information. For the tablet, the tablet user controlled when the data was collected, but the bandwidth data file was retrieved by the command center. A program to determine which programs and applications were running on the laptops was also put on the laptops. Like the bandwidth program, the command center remotely activated the program, and then retrieved the data file. The ping time data was collected through a program used by the command center without remote access. This ping time data was then used to make a visualization of the network.

Experiments
Our testing was comprised of seven days of continuous network measurements that recorded the current ping time and bandwidth of five devices: two MacBook Pro laptops, one PC laptop, one iMac desktop, and one Android tablet. It also collected which programs were running on the laptops. The measurements were taken every five minutes.

We made controlled changes to the configuration of the network by repositioning the devices to be closer or farther away from the router in order to emulate the movement of such devices within a typical household. Additionally, we disconnected and reconnected various devices at different times of the day to emulate the random combinations of devices that may be connected to a home network at any given moment. Finally, we experimented with opening up potentially intensive applications such as Spotify and buffered YouTube videos in various quantities on various combinations of devices to see how stress affects network performance.

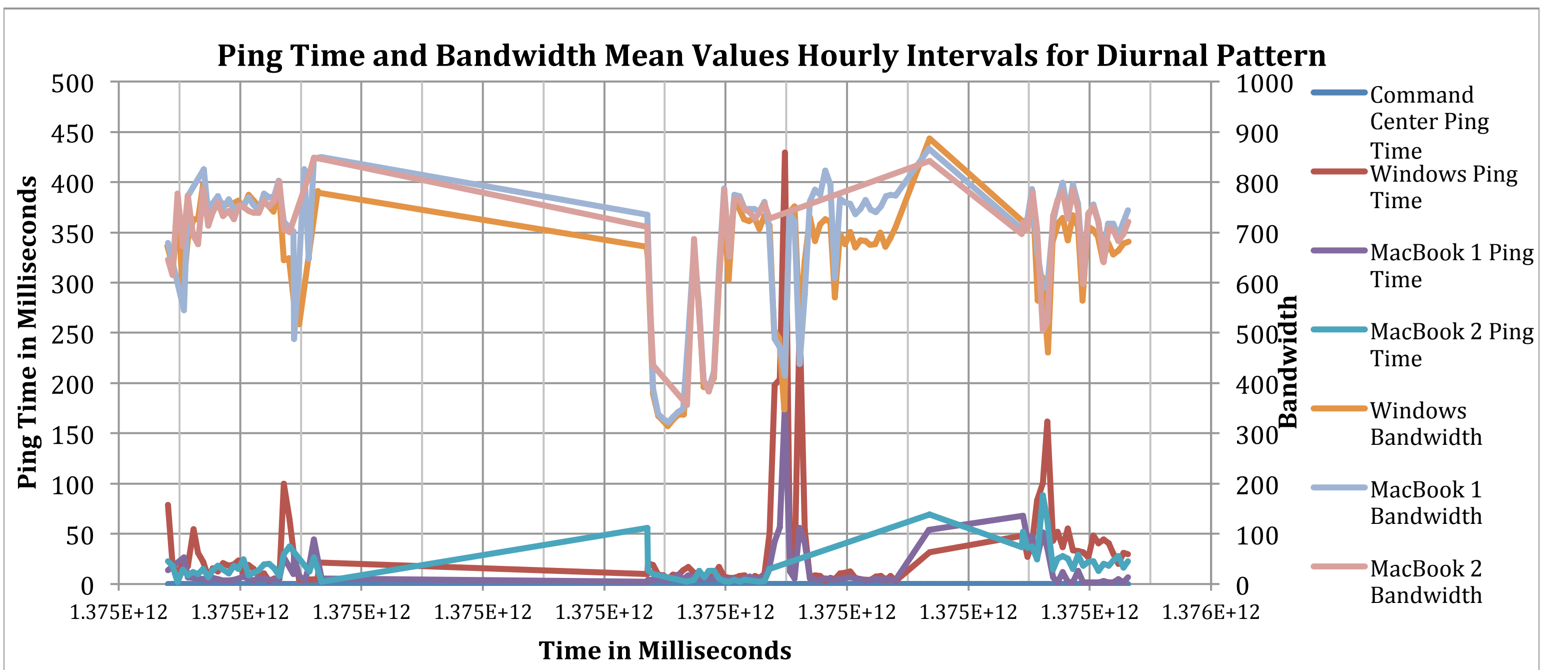
Results

We have two graphs that provide what we believe to be the most important data relevant to the state of the network. The tick marks on the x-axis have intervals of 24 hours, starting at 12:00 a.m. Thursday, July 25 and ending at 12:00 a.m. Saturday, August 3. The long, stagnant lines toward the left of the graph represent the Saturday and Sunday in which we did not gather data.



The Android tablet has a much longer ping time than the laptops. Since the tablet performed under the same conditions as the laptops, we think that its worse ping time could either be because it is a different operating system, or simply it could be because it is a tablet.

To observe the ping time and bandwidth for the laptops more closely, we have another graph without the Android tablet data.



There visually appears to be an inverse correlation between ping time and bandwidth in that the ping time increases when the bandwidth decreases. The Windows laptop has the largest spikes, so it is the device that is most affected by poor network conditions. And in the second half of the graph, the Windows laptop often has a lower bandwidth than the other laptops. Another observation is that around noon each day, the bandwidth would decrease, and the ping time usually increased a little bit.

In looking at the bandwidth and ping time data and corresponding it with what experiments we were running, we were not able to come up with many conclusions. It appears that when video is running, the ping time spikes, meaning that the devices take longer to respond.

During the first two drops in bandwidth and the first two increases in ping time (on the first Thursday and Friday), the devices were running videos and other data consuming applications such as Spotify. On Thursday, all of the devices on the network were running video, but on Friday, only one of the devices on the network was running video.

After the weekend, the largest and longest bandwidth dip occurred. This happened when we moved the devices further from the router. Most of them were in other rooms. However, none of them were running applications. It was interesting to note that we kept them in these spots throughout the night, but the bandwidth increased and the ping time decreased. The next day, we kept them in these farther locations, but also ran video. This had the largest effect on the ping time - ping time dramatically increased. However, bandwidth did not have such a dramatic change. The bandwidth did decrease, but it was not as dramatic as the day before when there were no videos running. When they were left overnight in the faraway locations without running any applications or videos, their ping times decreased and bandwidth increased.

On Thursday, the devices were moved to be next to the router, and they were not running any programs. The bandwidth and ping times were fairly constant. Overnight, the laptops were turned off. When they were turned on again, they were set up next to the router and they all were running video and the Macs were running Spotify. This increased ping time and decreased bandwidth. However, overnight they were sitting next to the router without anything running, but unlike the other nights, the bandwidth has two spikes and one dip, and the ping time was higher than normal.

The data stops at around 8:30 a.m. on the second Friday.

We were not able to come up with many conclusions about the bandwidth and the experiments. The bandwidth always seemed to drop around noon, no matter where the devices were, or what applications were running on the devices. However, one thing to note is that the bandwidth decreased more when the devices were further away from the router. But, when it wasn't around noon, even when the devices were the same distance away, the bandwidth went to what appears to be normal levels.

Conclusion and Future Work

We created a test bed, with a command center, three laptops, one tablet, and tools to calculate and retrieve measurements. We then collected data for seven days, and came away with a few conclusions. First, certain devices are more susceptible to network conditions than others and have slower performance, as was the case with the Windows laptop and Android tablet. Second, slower ping times are possibly caused by the devices running videos and Spotify. Third, bandwidth always decreased around noon, no matter what was running or distance.

For future work, we think it would be interesting to run more experiments controlling packet loss. However, before this is done, the algorithm to determine if devices are on the network would need to be modified. Right now, it basically assumes that there is no packet loss on the network, which for our network worked, but it is not a reasonable assumption for all networks.

Another experiment that would be good to look at would be to collect more video quality data at the same time as bandwidth and ping time data. We did collect some video quality data, but unfortunately it was not enough to be able to make any conclusions or see any patterns. Also, it was difficult to compare to the ping time and bandwidth data because there are only a few data points for the video quality analysis.

For another tool development suggestion, we think it would be a good idea to try to find out a way to not have the IP addresses of the devices that can be remotely accessed hard coded into the programs on the command center. We do not know if this is possible and could not come up with anything ourselves, but this step would make the program much more user friendly for potential home users.

We think that Nmap has the possibility of being a useful tool. Nmap could be used to find which devices are on the network, and it could also give ping time. It can also report much more data, so there are probably other uses for it in the command center.

As mentioned earlier, the Android tool that measures network information such as bandwidth needs to be modified to update the information continuously, not only when opened. Or, possibly come up with a completely different strategy for measuring this data. This information will be useful in understanding the extent to which Android tablets are affected by network conditions in comparison to laptops.

It might be a good idea to develop more measurement tools for other applications such as Spotify. We did try to create an app to measure Spotify, which did not work, but maybe with more time it could work, or it could be possible to take measurements from different applications.

We suggest looking more closely at the data and the analysis of the data and trying to come up with a singular "health score" that incorporates the bandwidth, ping time, and other potentially useful data. This will help the command center know whether or not the devices are experiencing good network quality. This is an important next step in making the network self-healing.