How's My Network Online Games

Entertainment Applications for Network Performance Measurement

The field of computer networks is especially relevant as our world becomes more and more digital. The best way to strengthen our knowledge in this area and make apt goals regarding it is to collect network data. How's My Network (http://hmn.cs.wpi.edu) is a project devoted to testing network connections. It not only provides helpful feedback to its users about their network connections, but also stores the data it gathers for future study. The subject of this project is the addition to the How's My Network project of a Website that hosts several online games that run network tests while being played (http://hmn-games.cs.wpi.edu). Our approach to collecting network data is to host online games that users can enjoy while visiting our Website. While they play the games, our network tests are executed in JavaScript that runs inconspicuously in the background. In order to get feedback on our Website as well as gather network data, we issued a survey, primarily filled out by students of WPI. Our analysis of the survey responses gives breakdowns for each question and summaries for the open response questions. Impressions of our games were generally lukewarm, and most experienced little to no lag from the network tests while playing the games. There were a few bugs with one of our games, Tetríssimus, but nothing game-breaking. Then we analyze the network performance data for both hosts and the ten servers we tested with. The WPI origin server stood out as having the greatest average throughput by far at 13.21 Mbps. The Japan server had the largest average round-trip time at 464.98 ms, and the Minnesota server had the most jitter by far with a coefficient of variation of 0.64. The limited scope of this project left some room for improvements, and the feedback from the survey also gives us direction for the future.

Computer Science MQP
Advised by Mark Claypool

Alec Mitnik
4/26/2012

Table of Contents

1. Introduction

The field of computer networks is especially relevant as our world becomes more and more digital. People want speed and reliability in their network connections, but to improve current networks we need to better understand their current capabilities. For instance, what is the bottleneck to performance? Throughput? Round-trip time?

The best way to strengthen knowledge of current network performance and make apt goals regarding it is to collect network data. There are unfortunately few sources of openly available, comprehensive data for residential networks. Existing sites that gather network data do not accommodate residential networks, while others are not very comprehensive.

How's My Network (http://hmn.cs.wpi.edu), abbreviated as "HMN," is a project devoted to gathering performance data on the network connections for end-users. HMN not only provides helpful feedback to its users about their network connections, but also stores the data it gathers for future study. How's My Network strives to be easy and fun to use. A Website is the chosen medium because it does not require downloading external applications, and there are few impediments to its use.

The subject of this project is the addition to the How's My Network project of a Website that hosts several online games that run network tests while being played (http://hmn-games.cs.wpi.edu). This endeavor hopes to draw people in and give them motivation to devote some of their time for testing. To do this we found and adapted a couple online games in addition to a game we coded from scratch. We have three games so far, but intend to expand our collection in the future.

Our approach to collecting network data is to host online games that users can enjoy while visiting our Website. While they play the games, our network tests are executed in JavaScript that runs inconspicuously in the background. Tests begin when the page for a game is loaded, and they continuously repeat after minute-long breaks. We measure three aspects of network performance: throughput, round-trip time, and jitter. We get these measurements by recording the time it takes to download specifically sized objects. We made a bitmap image file of size 100,000 bytes to download for measuring throughput, and a 43 byte bitmap image file to measure round-trip time. We test with ten different servers from around the world, and use our objects or objects of similar size.

In order to get feedback on our Website as well as gather network data, we conducted a user study, primarily participated in by students at WPI. A survey gathered demographic information such as age and gender, and also information such as time spent playing online games and interest in testing network connections. The survey then asked the participants to visit the Website and play all three games. After that, the participants gave feedback on their experience, with suggestions for improvement. In addition to the data gathered from the survey responses, we also gathered 1270 network test entries to analyze for this project.

Our analysis of the survey responses gives breakdowns for each question and summaries for the open response questions. Most participants considered themselves very familiar with computers/technology. Impressions of our games were generally lukewarm, and most experienced little to no lag from the

network tests while playing the games. There were a few bugs with one of our games, Tetríssimus, but nothing game-breaking. Among other things, it was suggested that we measure upload throughput as well as download throughput.

Next is our analysis of the network performance data for both hosts and the ten servers we tested with. The host analysis first considers performance over all ten servers, then only a geographically close server, and finally just a geographically far server. Each of the three performance measures are summarized with a cumulative distribution graph, a complementary cumulative distribution graph, and a table with the minimum value, maximum, mean, median, and standard deviation. Average throughput was greater overall for local servers than remote servers. Throughput for all servers was much higher, but that probably has to do with including the WPI origin server in that group. Round-trip time was larger for remotes servers and smaller for local servers, as expected. Jitter actually seemed to be smaller in general for remote servers, though they also had the largest jitter value, so the range of values is higher. The WPI origin server stood out as having the greatest average throughput by far at 13.21 Mbps. The Japan server had the largest average round-trip time at 464.98 ms, and the Minnesota server had the most jitter by far with a coefficient of variation of 0.64.

The limited scope of this project left some room for improvements, and the feedback from the survey also gives us direction for the future. We will continue to improve the How's My Network project in order to someday see its goal realized.

2. Background and Related Work

Current tools for network measurement aren't easily accessible for typical users.[5] Some tools, such as Dimes[4] and DipZoom[2], require the installation of software despite providing high quality data. If not installations, then tools require permanent contributions such as PlanetLab[7] and Archipelago[6]. The tests that are easily accessible, such as Speedtest[1], which is available through a Web browser, are not helpful for gathering data for research purposes. Another tool, Gomez[3], does not provide that data it gathers to its users or people who would want to study it.

Another issue is that the raw network performance data that these tools provide do not help most users understand how their network performance affects the applications they use. Network performance affects anyone who accesses the Internet, yet only those professed in computer networks would be able to make sense of the feedback these tools provide. These tools should also make an effort to explain how certain networks perform for different applications, such as VOIP or online games.

Games With a Purpose (http://www.gwap.com/gwap/), abbreviated "GWAP", is one of the inspirations for this project. GWAP uses online games to gather data for technological purposes. The behavior and responses that players exhibit while playing the games are used to train computers to understand how to accomplish similar tasks. We also set out to create games that have a purpose, and take a similar approach by using online games to draw in visitors while simultaneously collecting the data we seek.

3. Approach

This section describes our approach to developing a Website for the How's My Network project. The Website's purpose is to host online games, and then, while those games are played, run network tests and collect network data. We implemented three JavaScript games for the site in order to provide users with some choices while staying within the scope of this project, and went on to create a high score database in order to have high score tables for the games. High scores were implemented to enhance user experience and interaction, as well as provide greater replay value. Then we created the network tests, using JavaScript, which measure the player's network performance. The data from these tests are collected and stored in a database for further research.

3.1. Online Games

Our approach to gathering network data was to use online games to draw people in and keep them occupied, while network tests could be run on their machines. Thus, our first task was to acquire some online games. We were not sure how the game and the tests would interact, so we sought out JavaScript games to allow easy integration between the games and our tests. As it turned out, the network test code runs independently from the games, so other formats are future possibilities as well.

We wanted to start with simple, well-known games that would interest people visiting the site and be familiar to them. As the project grows, we expect to add more games to our collection. We employed two techniques to acquire games: coding a game from scratch, and adapting an open source game. Three games are currently hosted on our site: a sliding block puzzle made using the first approach, and implementations of Mine Sweeper and Tetris that were acquired with the second approach.

3.2. Sliding Block Puzzle

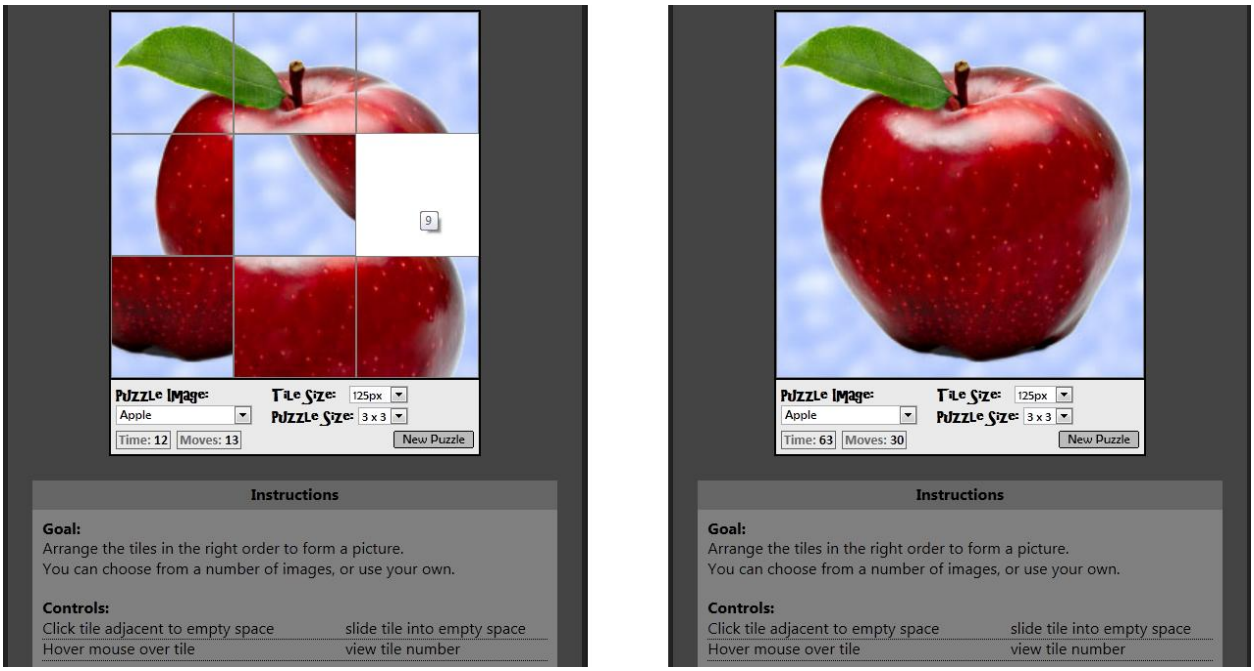

Figure 1. Sliding Block Puzzle

Figure 1 shows images of the sliding block puzzle game. The goal is to arrange the tiles correctly to form a picture. The player can choose the puzzle size (from 3x3 to 8x8), the tile size, and the image for the puzzle. Both seconds passed and moves made are recorded as your score. The player can start a new puzzle with the selected settings by clicking the "New Puzzle" button.

A lot of effort went into taking a simple concept and embellishing it in many ways. First, the numbered tiles became pictures. Pictures were not always clear about the correct tile order, so the tile numbers were made accessible through the hover text for each tile. Players can choose between a selection of different pictures. Players can also use any image on the Web, as well as any image that can be uploaded onto our server. The last option in particular turned a simple, standalone file into a more complicated client/server process.

Uploading an image file causes the page to redirect to a PHP file that processes the upload. If successful, the image file is named "MyImage" appended with an incrementing number stored in a flat file. The PHP file then redirects back to the sliding block puzzle Webpage. File uploading is a complex process, and because it redirects the page, any running network tests get interrupted and restarted. This code was implemented before any other client/server interaction, and before XMLHttpRequests were implemented to send data to the server without redirecting the page. Perhaps XMLHttpRequests could be utilized here, so that uploading would not require redirecting the page, but unfortunately there was not enough time to explore this option.

A significant challenge was figuring out how to best shuffle the initial tile layout. We decided to settle with making a proportionally large number of random valid moves (the number of tiles times 20) at the start of the game. It would have been preferred to completely randomize the layout, simply taking a few measures to ensure that it would always be solvable; however, any algorithm attempted for this strategy did not succeed for every different puzzle size. Perhaps this can be attempted again if a working algorithm is found.

3.3. Buscayasminas (Mine Sweeper)

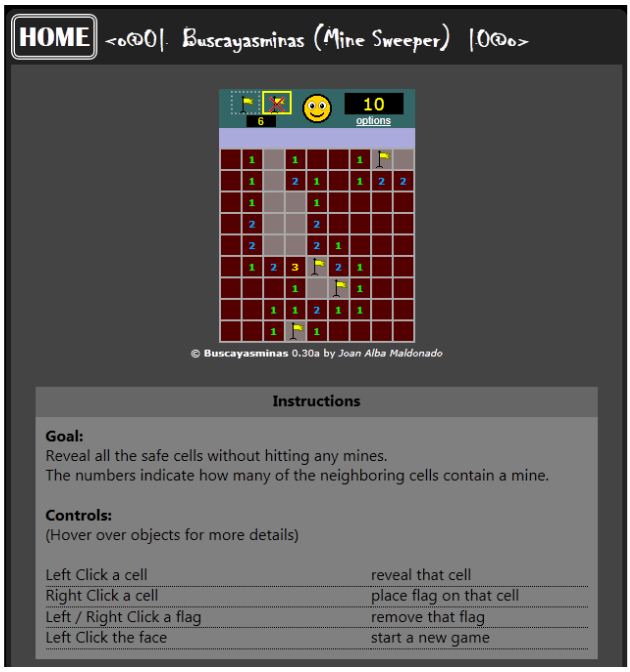

Figure 2. Buscayasminas (Mine Sweeper)

Figure 2 shows an image of Buscayasminas (Mine Sweeper). The goal is to reveal all the safe cells in the grid without uncovering any mines. The player can select from three difficulty settings. The number of seconds passed is recorded as your score. The player can start a new puzzle by clicking the face at the top.

Buscayasminas (original version at http://www.downv.com/Windows/download-Buscayasminas-10124844.htm), by Joan Alba Maldonado, is an implementation of Mine Sweeper and was found through an online resource for open source JavaScript games. It was chosen for use due to its popularity and relative simplicity. It was challenging to adapt the code, however, because it was all written in Spanish. Through logical reasoning and some help from Google Translate, this game was successfully integrated into our Website. Its layout was altered to have relative rather than absolute positioning so that the game would not only ever appear in the top left corner of the page, its sprites were all redesigned to something more generic and polished, and the puzzle size options, which originally let you choose the dimensions of the puzzle and the number of mines, were simplified to support only a beginner, intermediate, or expert setup.

3.4. Tetríssimus (Tetris)

Figure 3. Tetríssimus (Tetris)

Figure 3 shows images of Tetríssimus (Tetris). The goal is to stack the pieces to form complete rows, which then disappear, without letting the stack ever reach the top. The points earned by placing pieces and forming complete rows are recorded as your score. There is a button that lets the player pause them game, or start a new game after a game over.

Tetríssimus (original version at http://sourceforge.net/projects/tetrissimus/), also by Joan Alba Maldonado, is an implementation of Tetris, and was also chosen for use due to its popularity and relative simplicity. Also like Buscayasminas, all of its code was written in Spanish, yet in this case there were lots of interface messages displayed in Spanish as well, which needed to be translated. Its layout was converted to be relative rather than absolute positioning so that the game would not only ever appear in the top left corner of the page, and it was slightly redesigned to look more polished and user friendly.

Very little of the functional game code was altered, although it was changed to randomize the horizontal position of new falling pieces. Factors such as time constraints and the language barrier prevented further modifications, and the code works for the most part, but as our survey results in section 4.2 revealed, there are a few bugs with the game. The game was functional enough to be playable, but could be fixed and improved in the future.

3.5. High Scores

We implemented a high score system for the site in order to engage the players and give them motivation to keep playing and improving. We keep a set of high scores for each game in a database, using MySQL database for this purpose. This idea required code to read in and present the high scores

to the player, as well as allow them to submit their own scores. This was accomplished with PHP to read in the high scores when the page first loads, and JavaScript to calculate and submit the scores of the players.

Implementing high score submission with JavaScript was a decision to emphasize performance and simplicity over security and true accuracy. We saw the consequences of this when the first two high scores for one of our games were credited to "BUT" in first place, and "SEX" in second. It is highly unlikely that the scores tied to those initials were earned. Perhaps further measures could be taken to keep the scores honest in the future, if devoting time to the cause is deemed worthwhile.

3.6. Network Testing Implementation

We host the games on a Website coded in PHP, HTML, and JavaScript. Since we wanted the tests to run while the games were being played, JavaScript was the natural choice for running our network tests. The tests could be confined to a single JavaScript file and imported into the pages hosting each game. The network tests run during gameplay and send the results to a database while also displaying them for the user by using Highcharts graphs (http://www.highcharts.com/).

There are three types of network measurements performed: throughput, round-trip time (RTT), and jitter measured as coefficient of variation in the RTT (the standard deviation divided by the mean). The value taken for each RTT test is the average of ten, in order to get more consistent results. The tests are set to run as soon as the page is loaded, wait for a minute when finished, and then repeat.

Our tests are measured with the time it takes to download an image from a specific server. We add a GET variable "n" to the image file address and set its value to the current date-time in order to prevent caching. At first we only tested using the origin server for our Website, but we later expanded to testing nine other locations around the world to give more comprehensive results. We modified our initial test design to perform each of the three tests on each of the ten servers, send that data, and then wait for a minute before repeating.

The throughput test relies on a large image, while the RTT test uses an image confined to a single network packet. We generated bitmap images of specific sizes for this purpose: 100,000B for throughput and 43B for RTT, and were able to use them in most cases. Some servers were unable to host the images, and so we use similar sized images that they already hosted instead. Perhaps at some point all our testing servers could be convinced to host these image files.

3.7. Website Layout

Each game is hosted on its own Webpage and linked to from the main page. We developed PHP template files that are imported into the top and bottom of each game's page, giving each game page a consistent style and layout. Below each game is a collapsible element containing the instructions for the game, because even though these games are well known, there are still people who would need to know how to play, particularly our implementation. We may also host lesser known games in the future.

The game and instructions are sectioned off, and underneath is another section containing a small survey asking about network information. Including user input in regards to network information enhances the data, but the survey is strictly optional and can be ignored. Underneath the survey section is a third section containing the network performance results, as shown with three dynamically generated graphs. These are Highcharts graphs that match our three network tests of throughput, round-trip time, and jitter.

This layout functions well for our purposes, and our survey results in section 4.2 show that users were largely neutral about the design. There were some suggested changes however, such as placing the network performance results beside the game rather than below, that could be taken into consideration for future implementation.

4. Survey Evaluation

We created a survey using Google Docs (http://docs.google.com) as part of a user study to get feedback on our project and collect network data to analyze. We chose Google Docs because it did not limit the number of survey responses we could gather, and the responses were stored in a spreadsheet that could be easily exported. The survey was open for one week, and only the data gathered during that time is used for analysis.

4.1. Survey

Our survey starts with four demographic questions on the first page so we can understand what who our participants are.

The second page asks three rating questions on the next page regarding technology, online games, and level of interest in the topic of our project. This helps us understand the investment or understanding the participants would have while using our Website. The options for rating questions are a single plus character "+" all the way up to five plus characters "+++++". We use these options to be as unbiased and unambiguous as possible.

The third page asks the participant to visit our Website at http://hmn-games.cs.wpi.edu/ and play each of the three games for about 3 minutes. This would be long enough to gather data and give an impression of the site and each game, while staying short enough so as not to be a burden to the participants.

The fourth page asks three more rating questions, this time regarding impressions of our games and their experience playing them. This section helps to steer us for our future work on the project.

The fifth page asks a couple of open response questions about impressions of our Website and any problems with the games.

The sixth page asks two more open response questions regarding recommendations for adding any games or network tests.

The seventh and final page allowed the participant to give their name had they been offered any sort of class credit for taking our survey.

4.2. Results

There were 49 survey takers in total; however, one of them neglected to answer any of the questions. All questions were optional, so the number of responses to each question varied, and was not necessarily 49. We had largely targeted Computer Science and IMGD/Game Development majors at WPI, so the results reflect the diversity within those departments. Percentages are rounded to whole numbers, so might not add up to 100.

1. Gender:

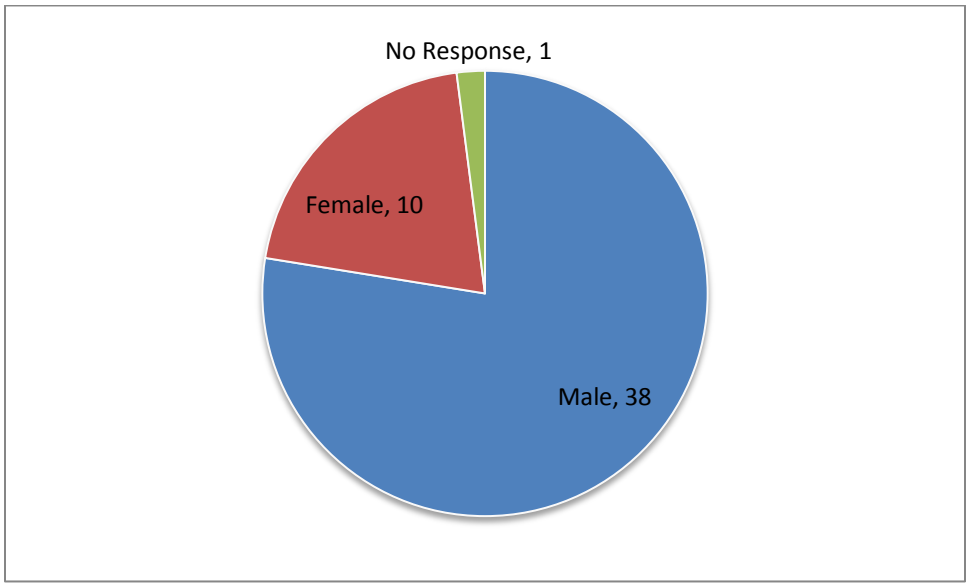

Figure 4. Distribution of Gender Responses

Selected Response	Number of Responses	Percentage of Responses
Male	38	78%
Female	10	20%
No Response	1	2%

Table 1. Distribution of Gender Responses

The different colored sections of Figure 4 represent the proportion of different responses, including “No Response,” representing a lack of a response. Table 1 shows the distribution of responses.

Nearly everyone indicated their gender. The majority of our test subjects appear to have been male. However, the fraction of women responses is about double the fraction of women within our target group.

2. Age:

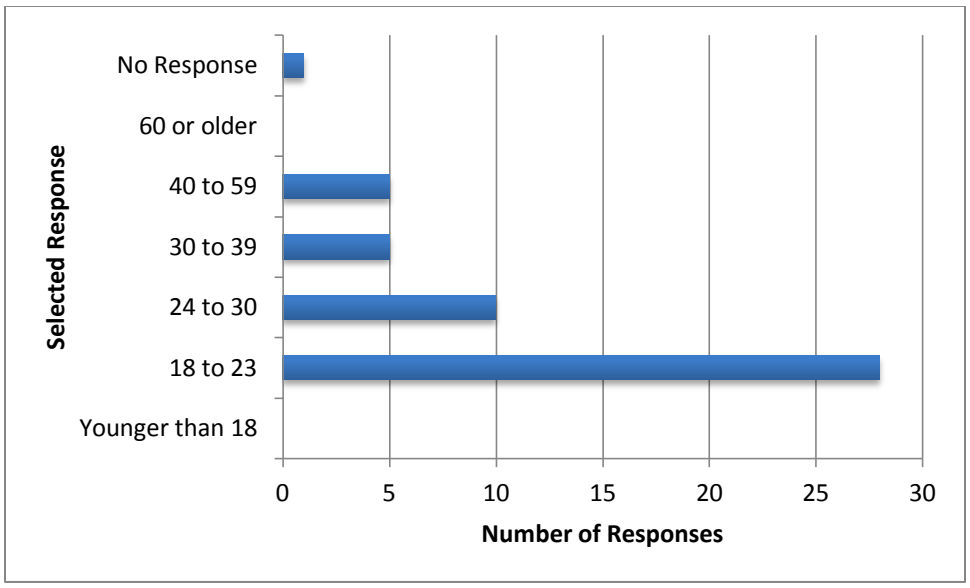

Figure 5. Distribution of Age Responses

Selected Response	Number of Responses	Percentage of Responses
Younger than 18	0	0%
18 to 23	28	57%
24 to 30	10	20%
30 to 39	5	10%
40 to 59	5	10%
60 or older	0	0%
No Response	1	2 %

Table 2. Distribution of Age Responses

The different bars on Figure 5 represent the proportion of different responses, including "No Response," representing a lack of a response. Table 2 shows the distribution of responses.

Nearly everyone indicated their age. All of our test subjects appear to have been between 18 and 59 years old. The majority were 18 to 23, which is the age range of most college students, and reflects our target of WPI students.

3. Occupation:

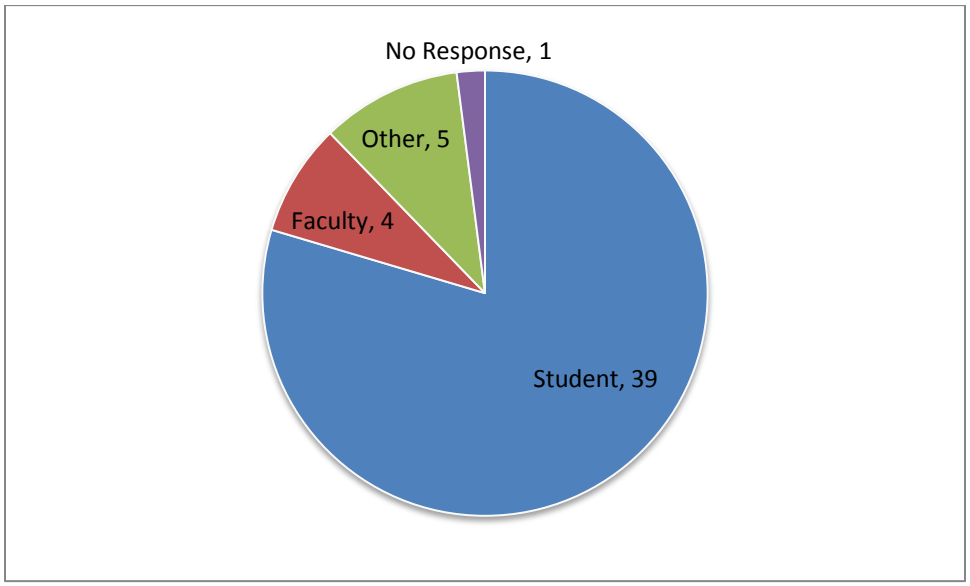

Figure 6. Distribution of Occupation Responses

Selected Response	Number of Responses	Percentage of Responses
Student	**39**	80%
Faculty	**4**	8%
Other	**5**	10%
No Response	**1**	2%

Table 3. Distribution of Occupation Responses

The different colored sections of Figure 6 represent the proportion of different responses, including "No Response," representing a lack of a response. Table 3 shows the distribution of responses.

Nearly everyone indicated their occupation. The large majority identified as students, reflecting our target of WPI students, although this shows that there were faculty and others as well.

4. Major/Department:

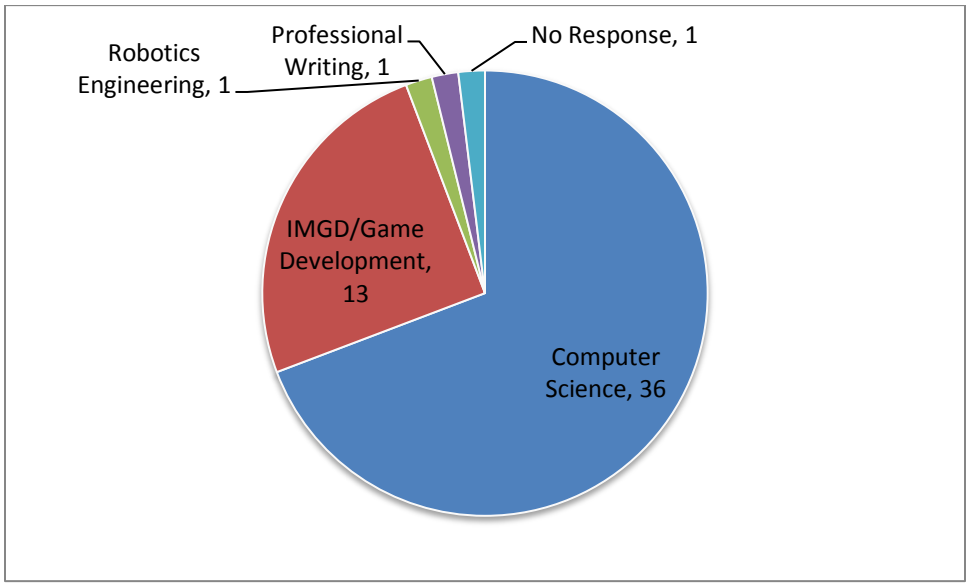

Figure 7. Distribution of Major/Department Responses

Selected Response	Number of Responses	Percentage of Responses
Computer Science	36	73%
IMGD/Game Development	13	27%
Robotics Engineering	1	2%
Professional Writing	1	2%
No Response	1	2%

Table 4. Distribution of Major/Department Responses

The different colored sections of Figure 7 represent the proportion of different responses, including "No Response," representing a lack of a response. Participants were allowed to select multiple responses, so percentages add up to more than 100%. The provided responses were "Computer Science," "IMGD/Game Development," and a third option in which they could write the subject of their choosing. Table 4 shows the distribution of responses.

Nearly everyone indicated their major/department. The majority selected Computer Science, and a fair amount selected IMGD/Game Development as well, reflecting our target of students with those majors.

5. *Familiarity with computers/technology:*

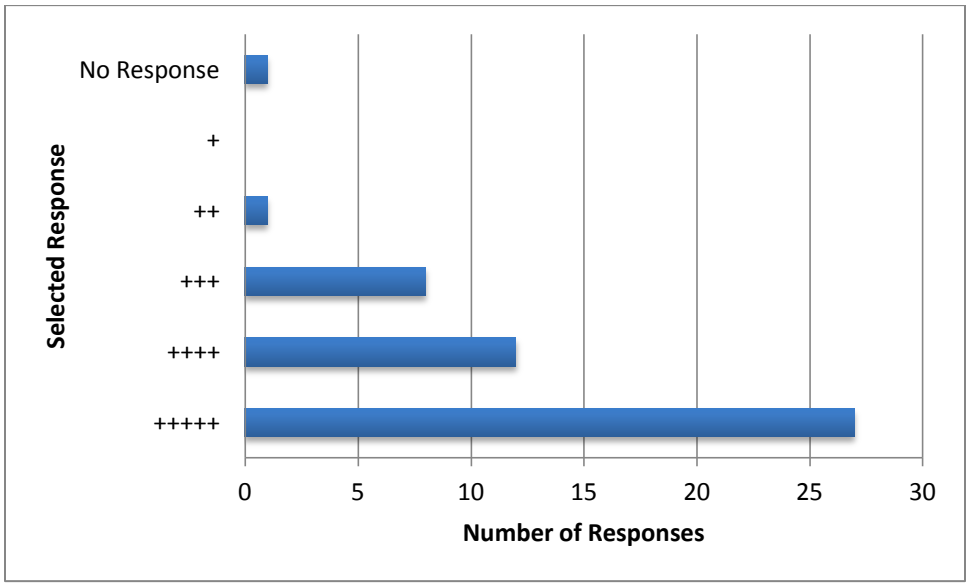

Figure 8. Distribution of Familiarity with Computers/Technology Responses

Selected Response	Number of Responses	Percentage of Responses
+++++	27	55%
++++	12	24%
+++	8	16%
++	1	2%
+	0	0%
No Response	1	2%

Table 5. Distribution of Familiarity with Computers/Technology Responses

The different bars on Figure 8 represent the proportion of different responses, including "No Response," representing a lack of a response. The use of symbols as response choices was meant to minimize ambiguity and maximize objectivity. Table 5 shows the distribution of responses.

Nearly everyone gave a response. Most of our test subjects appear to be very familiar with computers and technology, which reflects our target of WPI students. No one chose the least familiar option of a single symbol.

6. *Time spent playing online games:*

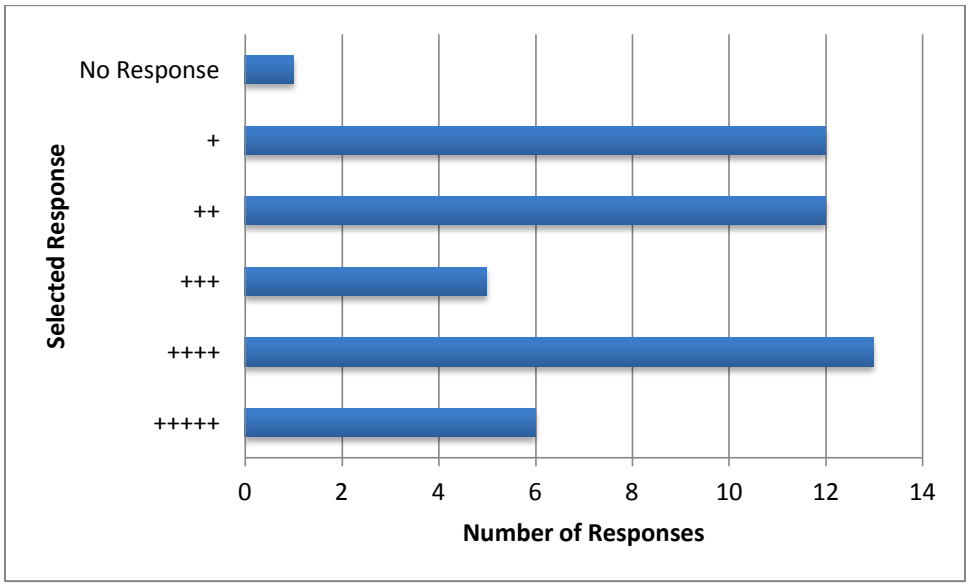

Figure 9. Distribution of Time Spent Playing Online Games Responses

Selected Response	Number of Responses	Percentage of Responses
+++++	6	12%
++++	13	27%
+++	5	10%
++	12	24%
+	12	24%
No Response	1	2%

Table 6. Distribution of Time Spent Playing Online Games Responses

The different bars on Figure 9 represent the proportion of different responses, including "No Response," representing a lack of a response. Table 6 shows the distribution of responses.

Nearly everyone gave a response. The responses show a wide range of devotion to playing online games among the participants, demonstrating that such a hobby does not necessarily correlate with familiarity with computers and technology.

7. *Interest in testing your network connection:*

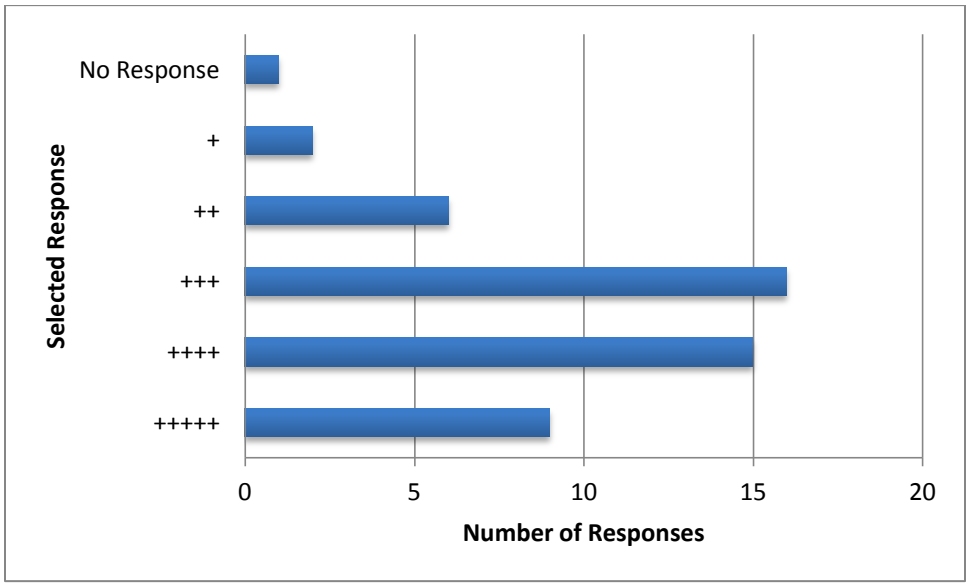

Figure 10. Interest in Testing Your Network Connection Responses

Selected Response	Number of Responses	Percentage of Responses
+++++	9	18%
++++	15	31%
+++	16	33%
++	6	12%
+	2	4%
No Response	1	2%

Table 7. Interest in Testing Your Network Connection Responses

The different bars on Figure 10 represent the proportion of different responses, including "No Response," representing a lack of a response. Table 7 shows the distribution of responses.

Nearly everyone gave a response. The majority of the participants appear to be at least somewhat interested in testing their network connection, which likely correlates with familiarity with computers and technology.

8. *Fun of playing the games:*

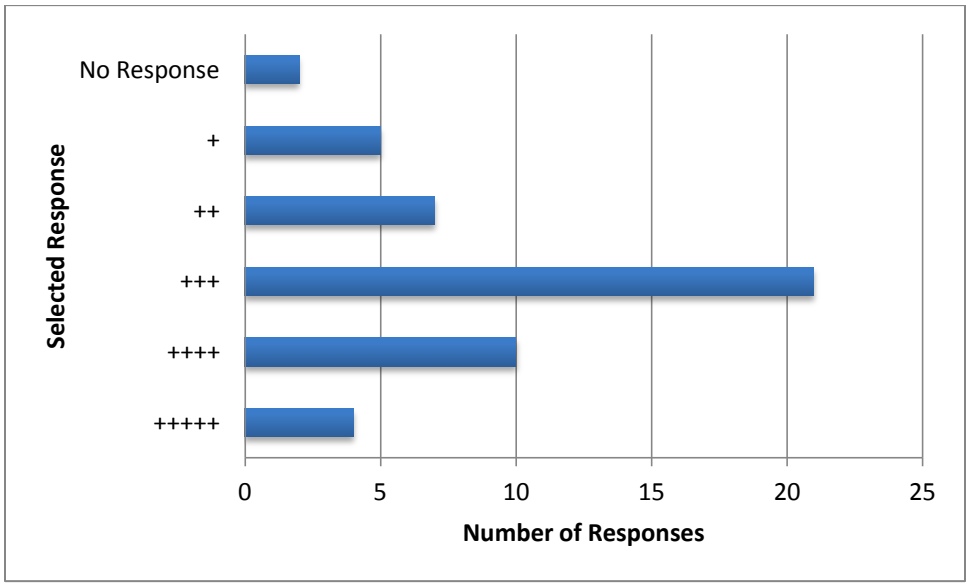

Figure 11. Fun of Playing the Games Responses

Selected Response	Number of Responses	Percentage of Responses
+++++	4	8%
++++	10	20%
+++	21	43%
++	7	14%
+	5	10%
No Response	2	4%

Table 8. Fun of Playing the Games Responses

The different bars on Figure 11 represent the proportion of different responses, including "No Response," representing a lack of a response. Table 8 shows the distribution of responses.

Nearly everyone gave a response. It seems that overall the participants did not feel too strongly about the games at How's My Network one way or the other.

9. *Noticeability/lag from network tests while playing the games:*

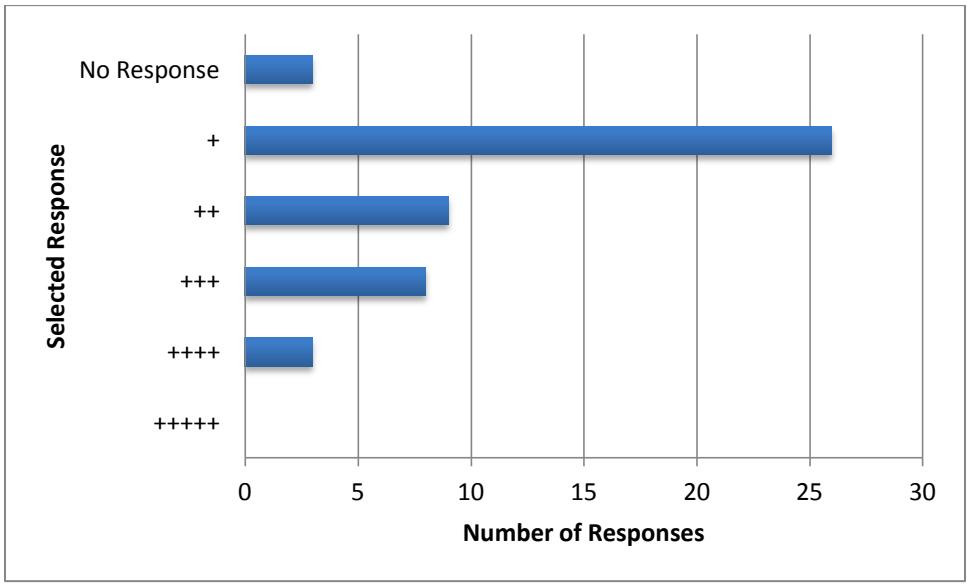

Figure 12. Distribution of Noticeability/Lag from Network Tests While Playing the Games Responses

Selected Response	Number of Responses	Percentage of Responses
+++++	0	0%
++++	3	6%
+++	8	16%
++	9	18%
+	26	53%
No Response	3	6%

Table 9. Distribution of Noticeability/Lag from Network Tests While Playing the Games Responses

The different bars on Figure 12 represent the proportion of different responses, including "No Response," representing a lack of a response. Table 9 shows the distribution of responses.

Only a few participants neglected to respond. Most of the participants appear to have experienced little to no lag while playing the How's My Network games, which bodes well for the implementation of the network tests, however there is room for improvement because some of the participants did experiencing some lag. No one chose the greatest lag option of five symbols.

10. Usefulness of network test results:

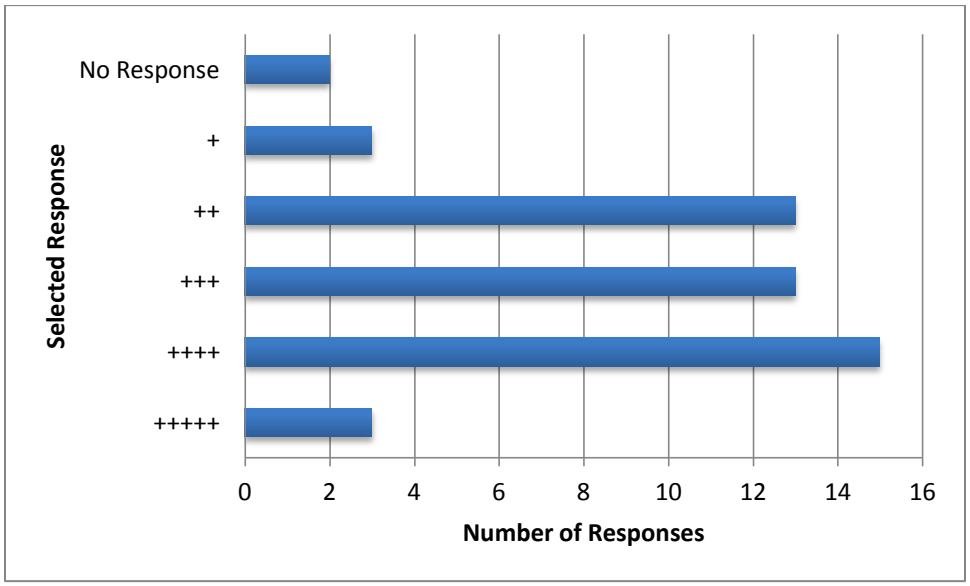

Figure 13. Distribution of Usefulness of Network Test Results Responses

Selected Response	Number of Responses	Percentage of Responses
+++++	3	6%
++++	15	31%
+++	13	27%
++	13	27%
+	3	6%
No Response	2	4%

Table 10. Distribution of Usefulness of Network Test Results Responses

The different bars on Figure 13 represent the proportion of different responses, including "No Response," representing a lack of a response. Table 10 shows the distribution of responses.

Nearly everyone gave a response. Most of the participants do not seem to have considered the network test results especially useful or useless, but merely somewhat useful.

11. What did you think of the site design/layout?

This was an open response question. Most participants chose to respond. The overall response was that the site design/layout was plain and simple. Some felt this made it boring or unappealing, while others considered it clear and easy to use.

There were a few remarks on the font for the section titles being confusing or hard to read, and some people suggested putting the test results up next to the game so they could both be viewed at once.

12. Did you find any problems with the games?

This was an open response question. Most participants chose to respond. The overall response was that there were no significant problems, although there were a few that mentioned lag or expecting different controls. A couple participants complained that the games were boring, as well.

There were some mentions of issues regarding Tetríssimus in particular. These complaints ranged from lag, to the graphics, to problems with the controls. It certainly was our most ambitious game, and we relied on the preexisting code, which apparently had flaws and room for improvement.

One participant took issue with the scoring system for the sliding block puzzle: *"Apple Sliding Puzzle is a bit unfair. Its [sic] all RNG based to get a near perfect number of moves for the high score board. That said I got a 6 which is pretty good."* It seems that the participant was unaware that the puzzle image could be changed from the default of that of an apple. More importantly, it is true that the number of moves required to complete the puzzle varies each time, since the tiles are shuffled randomly. This is, for practical purposes, unavoidable. To help keep the scoring fair, there is a separate score for the time it takes to complete the puzzle. We use both scores, seconds taken and number of moves, so that people can choose whether to play haphazardly yet fast, or slowly but cleverly.

13. Are there any specific games you suggest we add?

This was an open response question. Roughly half of the participants chose to respond, although some of them were unhelpful, simply stating "*no.*" There were multiple helpful suggestions for games to add, such as Solitaire, Sudoku, Pac Man, Asteroids, Bejeweled, Snake, and Pong.

14. Are there any additional network tests or changes that you suggest?

This was an open response question. Roughly half of the participants chose to respond, although most of them were unhelpful, simply stating "*no*."

One suggestion was to expand our throughput measurement to test upload speed as well as download speed. This was something we had not been able to include, but it is certainly something to possibly implement in the future. Another suggestion was to measure packet loss, although we are unsure if this can be accomplished in the same manner as our other tests.

15. Extra Credit

The final question allowed students to provide their names in order to receive extra credit for taking the survey, had extra credit been offered for one of their classes. There were several useless responses along the veins of "*I wasn't!*" However, three people did provide their names, while a fourth provided two unrecognized initials.

5. Network Data Evaluation

5.1. Network Data Approach

Our goal is to collect measurement data on the capabilities of end-user networks. To this end, we run a JavaScript program to perform network tests on visitors of How's My Network, which forms the data into JSON format before calling an XMLHttpRequest to asynchronously store the data into a MySQL database. We continually repeat our tests after a minute of waiting each time in order to get more measurement samples. The network tests are done by measuring the download time of specifically sized objects from ten different servers around the world. These locations include WPI in Worcester, MA, USA (the origin server for How's My Network); Australia; Singapore; Japan; Oregon, USA; Minnesota, USA; Boston, MA, USA; The United Kingdom; France; and Norway.

Each entry into the database includes the date and time of the test performed, the IP address of the host whose network was measured, a random cookie to identify all tests performed from the same Webpage in the same sitting, the actual network performance results, the number of the data entries amongst all entries with the same host and cookie, the results of an optional survey on the same Webpage as the game regarding the user's network, the host's user agent which includes the browser type and version along with the operating system, and finally a version number which is used to allow for association of data gathered to specific scripts. For this report, all data analyzed comes from the same script version.

The network performance results include, for all ten servers, the size of the large object downloaded to test throughput, the time to download which can be used to compute throughput, the size of the small object, presumably contained within a single network packet, used to test round-trip time, the time to download the object each time for ten downloads, representing round-trip time and from which jitter can be calculated, and the version number of the script used to perform the network tests (which was constant for all data in this study). When a test's download fails, the value for the measurement is marked as "ERROR" and excluded from any data analysis. Only big object downloads (for throughput) experienced errors, resulting in 17 "ERROR" throughput values out of a total of 12,700 (1270 test iterations with 10 servers per test iteration), or 0.13%.

Our analysis consists of data from a client's perspective, and data from a server's perspective. The client portion analyzes the average throughput, round-trip time, and jitter for each host. It first does this considering all ten servers used in testing, referred to as the aggregate analysis. Then it does a local analysis, only using data from the local server for each host. We assign the *local server* to be the geographically closest server to the client that is not in the same location as the client, and discounting

the origin server at WPI as well. Finally, it has the remote analysis, which only uses data from the *remote server* for each host, which is assigned as simply the server that is farthest geographically from each host. This allows us to observe any differences in how networks perform overall, with specifically local servers, and with specifically remote servers. The server portion only does an aggregate analysis that considers all hosts, since the uneven geographical distribution of hosts would be difficult to categorize for each server.

5.2. Host Analysis

The network information we gather and store includes the IP address of the measured host. Typically, IP address is a good identifier for an individual *host*, but there are cases, such as with the use of NAT boxes, where two different hosts can have the same IP address. In order to identify unique hosts within our data, we choose to assume that different hosts with the same IP address will have different user agents. This may not always be the case, and it is possible for a single host to switch between browsers or even operating systems, but it is a reasonable assumption for the purposes of this study. The results of our network tests during our survey analysis revealed 68 unique IP addresses. We conclude, because 4 of the IP addresses are tied to different user agents at different times, that our test results actually reflect the networks of 72 unique hosts.

When a host enters the Webpage for one of our games, a five alphanumeric character cookie is randomly generated in order to tie all network tests performed to that particular session of gameplay. We considered five characters to be enough to keep the cookies unique, although it turned out that our test data contained two different hosts with the same cookie, so more characters might be recommended for the future. We define a *session* as all the gameplay of an individual user after that user enters the Webpage for a game and until the Webpage is refreshed or the user exits the Webpage. All network data gathered from the same session contains the same cookie. During a session, the user's network is tested for throughput, jitter, and round-trip time across ten different servers from around the world, and the results of those tests are stored in a database. We call this a *test iteration*, because it results in a single database entry. For as long as a session continues, a new test iteration begins one minute after the previous test iteration (the first test iteration begins right away). Each test iteration within a session has a session number, starting with 1, and continually incrementing with each iteration. The results of our network tests include 173 unique cookies, implying 173 sessions in total. The results also show a total of 1270 database entries, meaning 1270 test iterations took place.

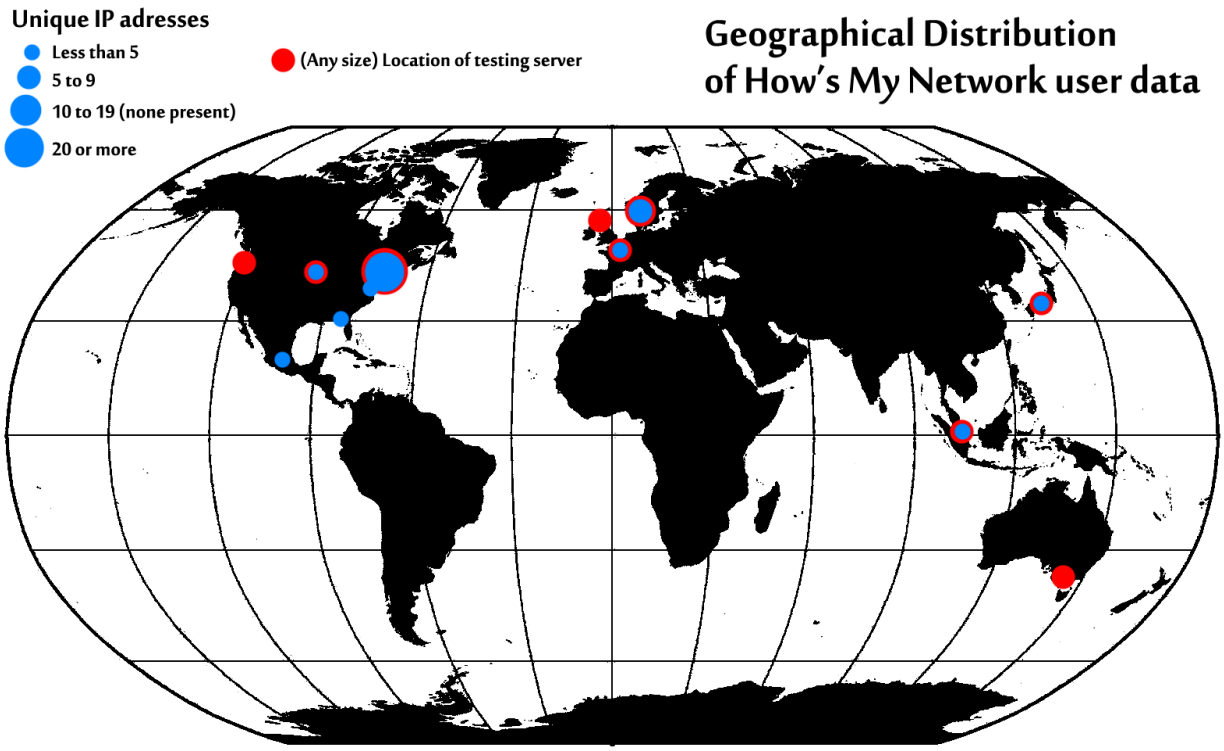

Figure 14. Map of server and host distributions

Figure 14 shows in red the geographical distribution of the ten servers used for network testing. It also shows in blue the geographical distribution of the IP addresses of the hosts that were tested. The large majority of hosts were located in Massachusetts, with several hosts originating in Norway, while the remaining locations each represent fewer than five hosts. Most of the hosts were around the area of our testing servers, as the Universities hosting those servers were our primary test subjects, but a few hosts came from Washington D.C., Florida, and Mexico as well.

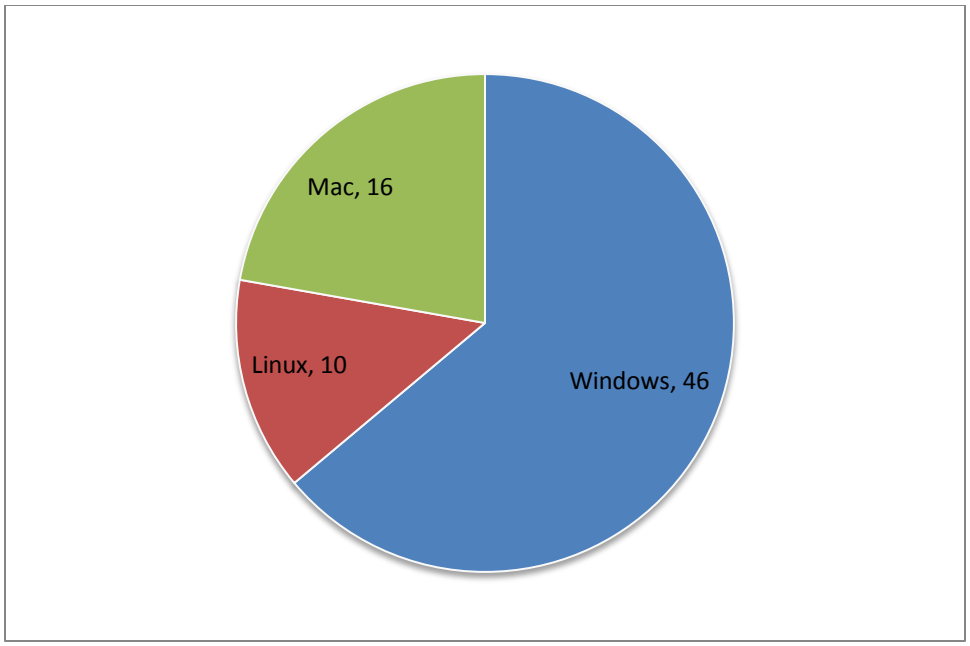

Figure 15. Distribution of Operating Systems

Figure 15 shows the distribution of operating systems among the hosts. The majority of hosts used the Windows operating system.

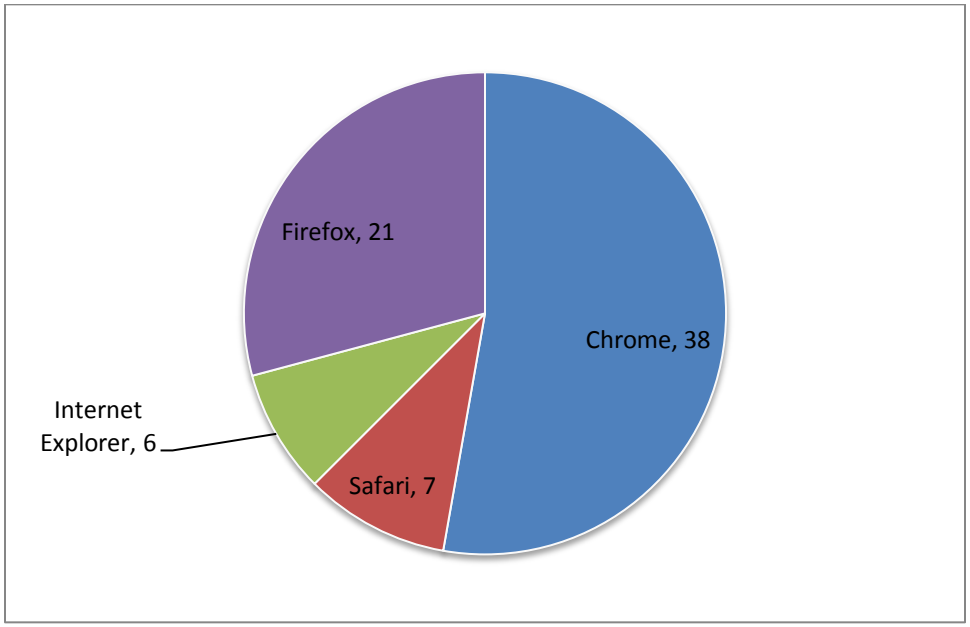

Figure 16. Distribution of Browsers

Figure 16 shows the distribution of browsers among the hosts. Over half of the hosts used Chrome, with Firefox being the next most common browser.

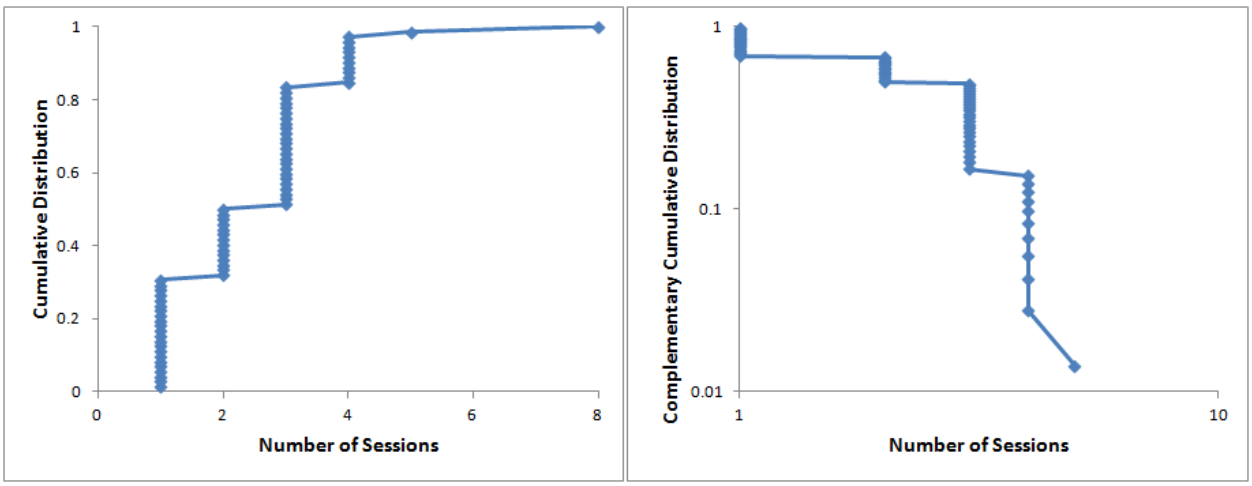

Figure 17. CDF and CCDF of Sessions per Host

Figure 17 shows the number of sessions per host as a cumulative distribution and a complementary cumulative distribution. It appears that there was a rather even spread from 1 to 5 sessions, with a single host having 8 sessions. The participants of the survey were only asked to do 3 sessions.

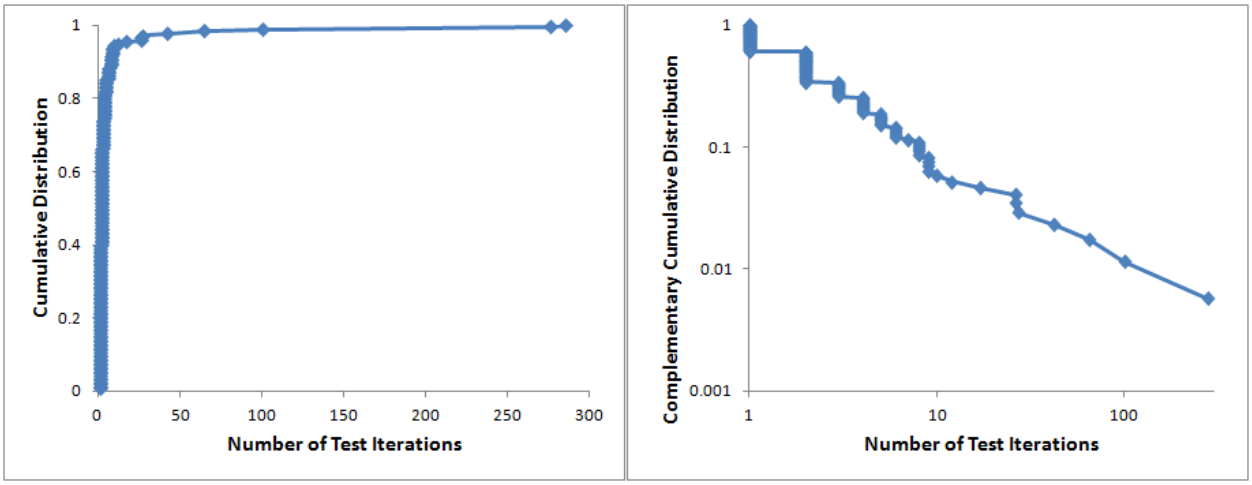

Figure 18. CDF and CCDF of Test Iterations per Session

Figure 18 shows the number of test iterations per session as a cumulative distribution and a complementary cumulative distribution. The large majority of sessions had very few test iterations, often only one. A few sessions had very many test iterations, however, with 285 being the most. That means one person stayed on a game's Webpage long enough for 285 test iterations to complete. With one minute between each test iteration, and keeping in mind that the test iterations themselves take some time, that amounts to well over 4.73 hours! Perhaps this person simply left the Webpage open while doing other things.

5.3. Client-Centric Analysis

5.3.1. Aggregate

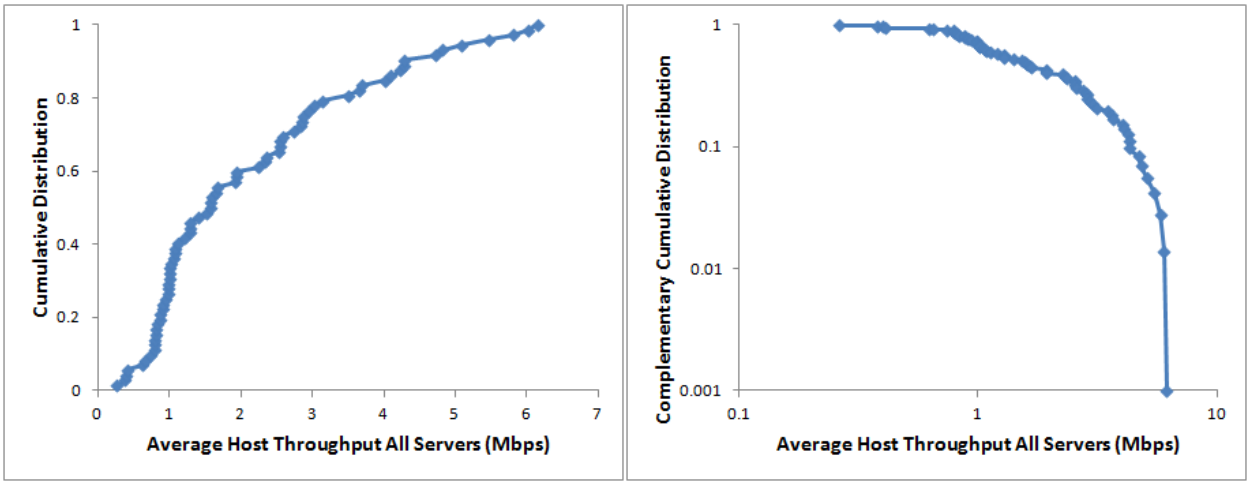

Figure 19. CDF and CCDF of Average Host Throughput for All Servers

Min	Max	Mean	Median	Standard Dev.
0.26	6.15	2.13	1.58	1.54

Table 11. Breakdown of Average Host Throughput for All Servers (Mbps)

Figure 19 shows the average throughput over all servers for each host as a cumulative distribution and a complementary cumulative distribution. Table 11 shows a breakdown of the throughput averages consisting of the minimum and maximum values, the mean, the median, and the standard deviation. The throughput averages are in Megabits per second. There is a fairly even spread of values from almost 0 Mbps to just over 6 Mbps, skewed somewhat towards the lower end.

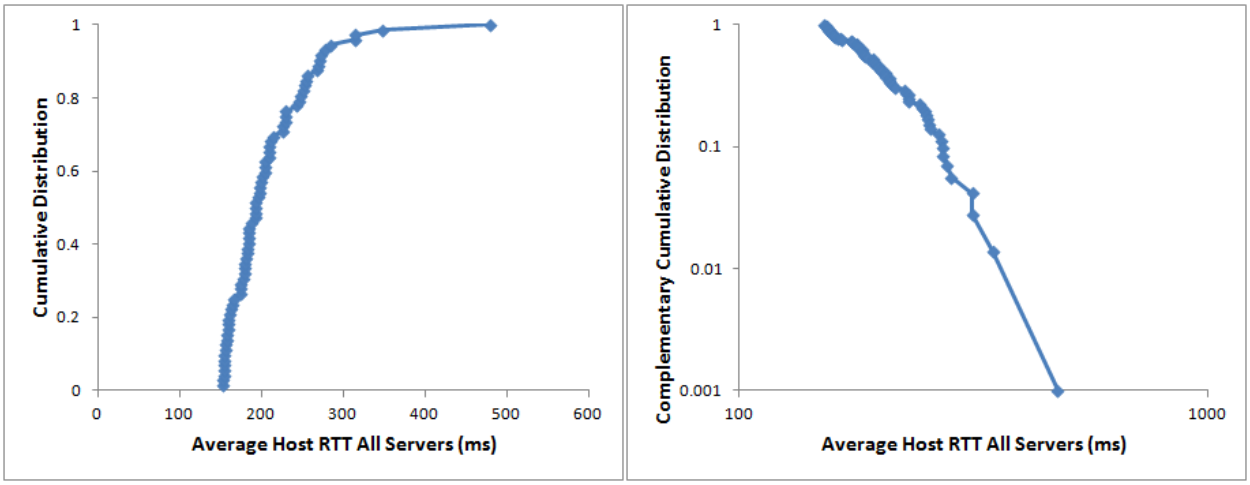

Figure 20. CDF and CCDF of Average Host RTT for All Servers

Min	Max	Mean	Median	Standard Dev.
151.86	478.92	207.7	192.72	55.13

Table 12. Breakdown of Average Host RTT for All Servers (ms)

Figure 20 shows the average round-trip time (RTT) over all servers for each host as a cumulative distribution and a complementary cumulative distribution. Table 12 shows a breakdown of the RTT averages consisting of the minimum and maximum values, the mean, the median, and the standard deviation. The RTT averages are in milliseconds. The RTTs were mostly spread between 150 ms and 300 ms, with a couple going beyond that and one at almost 500 ms.

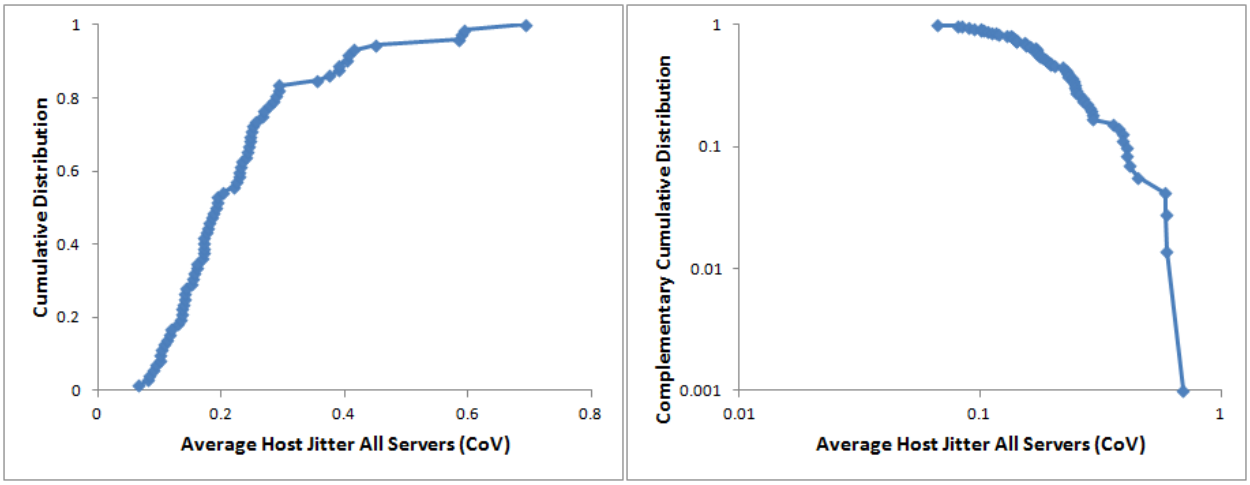

Figure 21. CDF and CCDF of Average Host Jitter for All Servers

Min	Max	Mean	Median	Standard Dev.
0.07	0.69	0.23	0.19	0.13

Table 13. Breakdown of Average Host Jitter for All Servers (CoV)

Figure 21 shows the average jitter over all servers for each host as a cumulative distribution and a complementary cumulative distribution. Table 13 shows a breakdown of the jitter averages consisting of the minimum and maximum values, the mean, the median, and the standard deviation. The jitter averages are measured as coefficient of variation (CoV), which is the standard deviation divided by the mean, and is unitless. There is a mostly even spread from about 0.1 to 0.3, with several hosts averaging around 0.4, a few at about 0.6, and one at almost 0.7.

5.3.2. Local

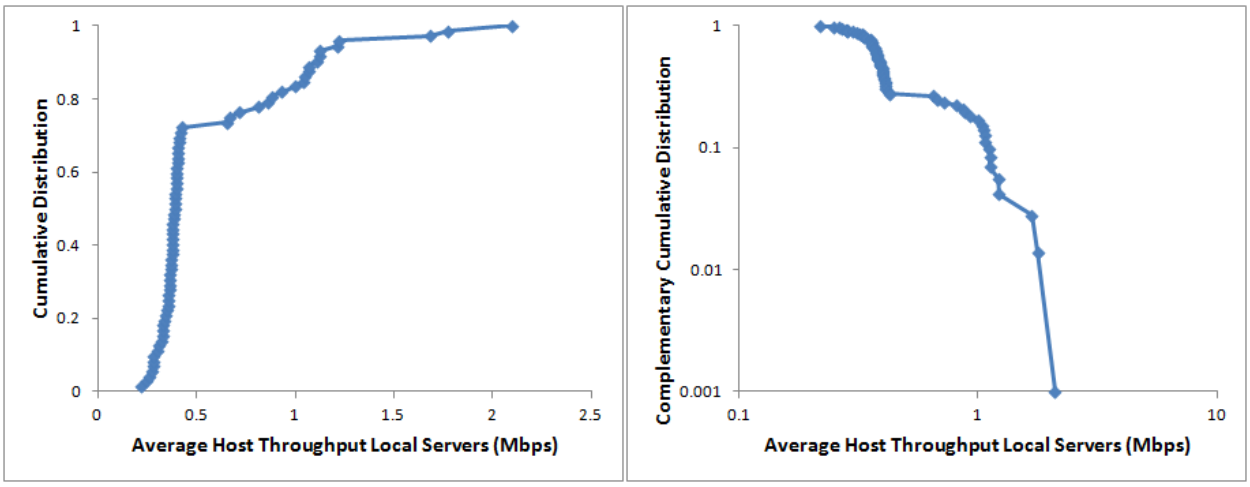

Figure 22. CDF and CCDF of Average Host Throughput for Local Servers

Min	Max	Mean	Median	Standard Dev.
0.22	2.1	0.57	0.39	0.39

Table 14. Breakdown of Average Host Throughput for Local Servers (Mbps)

Figure 22 shows the average throughput over local servers for each host as a cumulative distribution and a complementary cumulative distribution. Table 14 shows a breakdown of the throughput averages consisting of the minimum and maximum values, the mean, the median, and the standard deviation. The throughput averages are in Megabits per second. For local servers, many hosts lie just under 0.5 Mbps, with the rest spreading out at up to 2.1 Mbps.

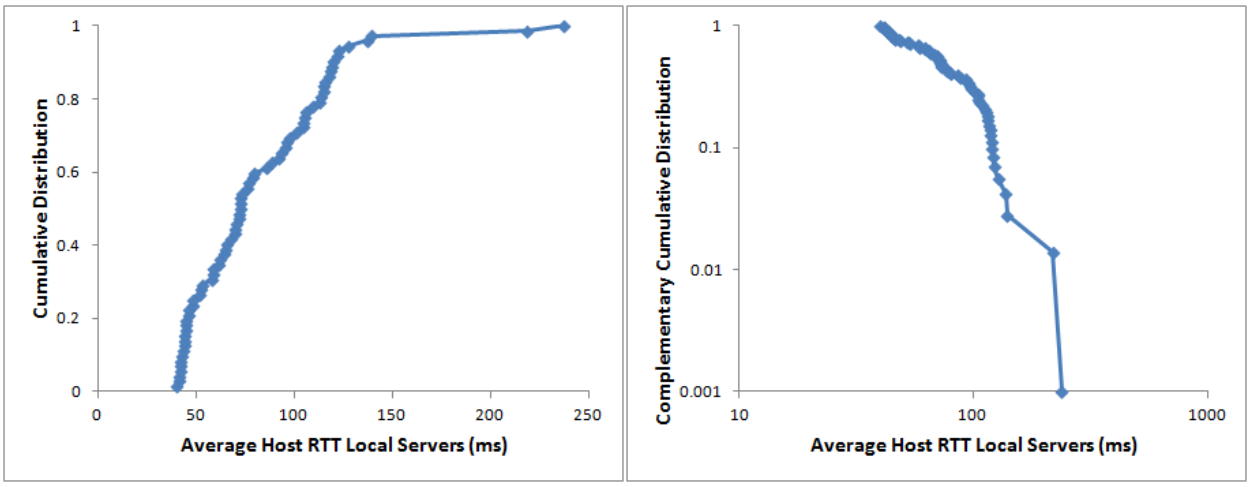

Figure 23. CDF and CCDF of Average Host RTT for Local Servers

Min	Max	Mean	Median	Standard Dev.
40.06	237.30	81.88	72.33	38.03

Table 15. Breakdown of Average Host RTT for Local Servers (ms)

Figure 23 shows the average round-trip time (RTT) over local servers for each host as a cumulative distribution and complementary cumulative distribution. Table 15 shows a breakdown of the RTT averages consisting of the minimum and maximum values, the mean, the median, and the standard deviation. The RTT averages are in milliseconds. For local servers, the RTTs are evenly spread from about 40 ms to about 140 ms, with a few values jumping up to over 200.

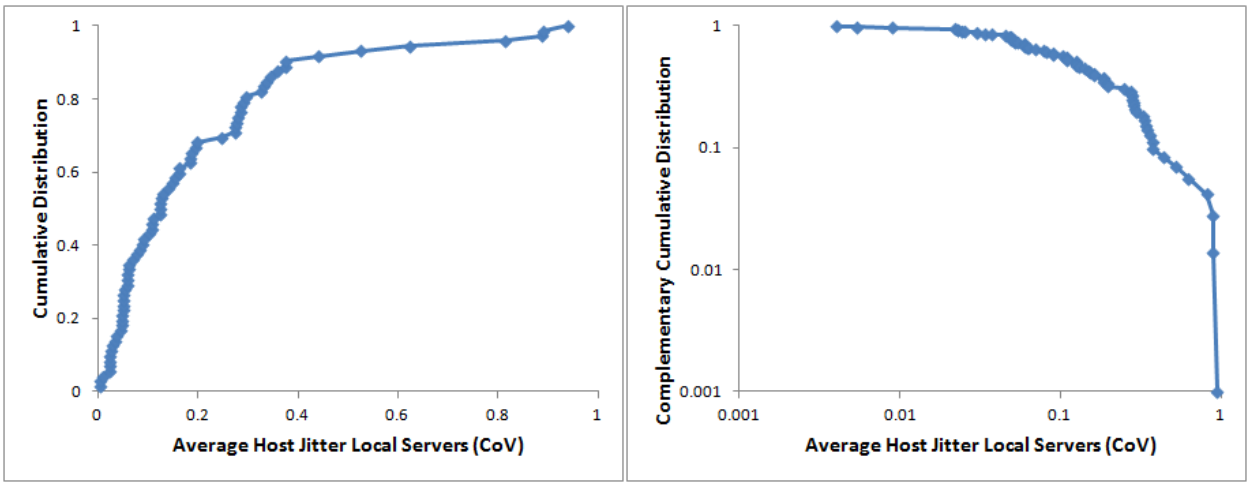

Figure 24. CDF and CCDF of Average Host Jitter for Local Servers

Min	Max	Mean	Median	Standard Dev.
0	0.94	0.2	0.12	0.21

Table 16. Breakdown of Average Host Jitter for Local Servers (CoV)

Figure 24 shows the average jitter over local servers for each host as a cumulative distribution and a complementary cumulative distribution. Table 16 shows a breakdown of the jitter averages consisting of the minimum and maximum values, the mean, the median, and the standard deviation. The jitter averages are measured as coefficient of variation (CoV), which is the standard deviation divided by the mean, and is unitless. For local servers, just as with all servers, most jitter averages were evenly spread in the low end of the range, with some portion jumping off into higher values. With fewer samples than the aggregate analysis, the jitter averages for local servers have a greater range.

5.3.3. Remote

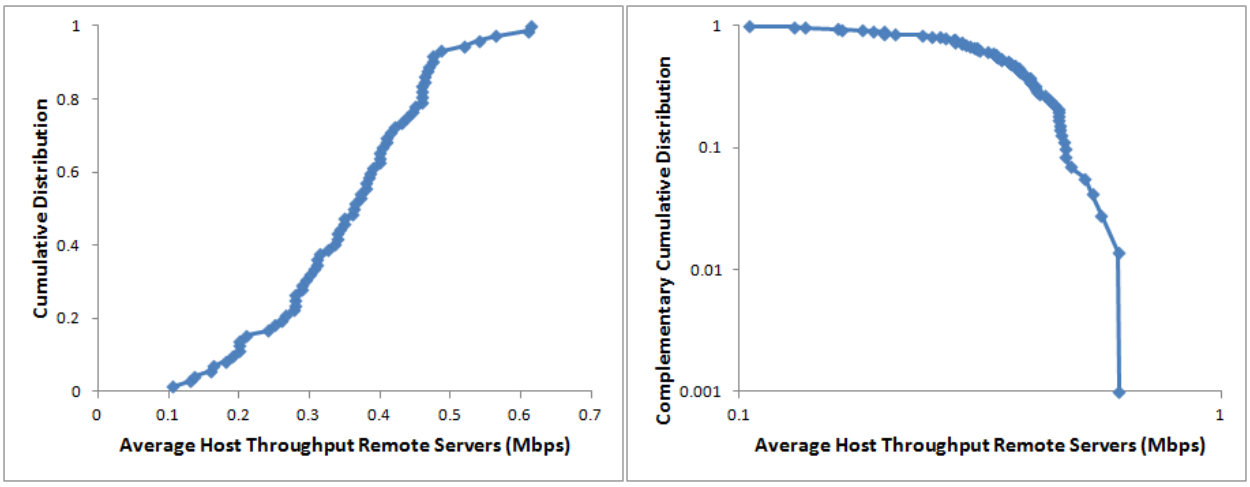

Figure 25. CDF and CCDF of Average Host Throughput for Remote Servers

Min	Max	Mean	Median	Standard Dev.
0.11	0.61	0.36	0.36	0.11

Table 17. Breakdown of Average Host Throughput for Local Remote (Mbps)

Figure 25 shows the average throughput over local servers for each host as a cumulative distribution and a complementary cumulative distribution. Table 17 shows a breakdown of the throughput averages consisting of the minimum and maximum values, the mean, the median, and the standard deviation. The throughput averages are in Megabits per second. For remote servers, there seems to be a rather even spread from about 0.1 Mbps to about 0.6 Mbps.

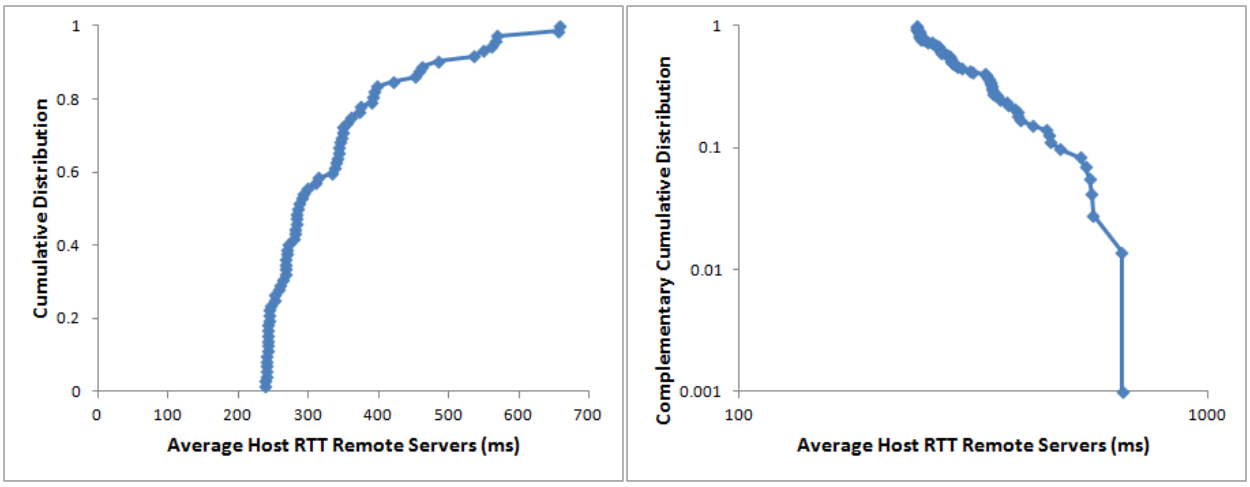

Figure 26. CDF and CCDF of Average Host RTT for Local Servers

Min	Max	Mean	Median	Standard Dev.
238.97	657.95	331.5	285.13	105.03

Table 18. Breakdown of Average Host RTT for Local Servers (ms)

Figure 26 shows the average round-trip time (RTT) over local servers for each host as a cumulative distribution and complementary cumulative distribution. Table 18 shows a breakdown of the RTT averages consisting of the minimum and maximum values, the mean, the median, and the standard deviation. The RTT averages are in milliseconds. For remote servers, the RTTs are largely between about 230 and 400 ms, with a gradually increasing trend up to about 660 ms.

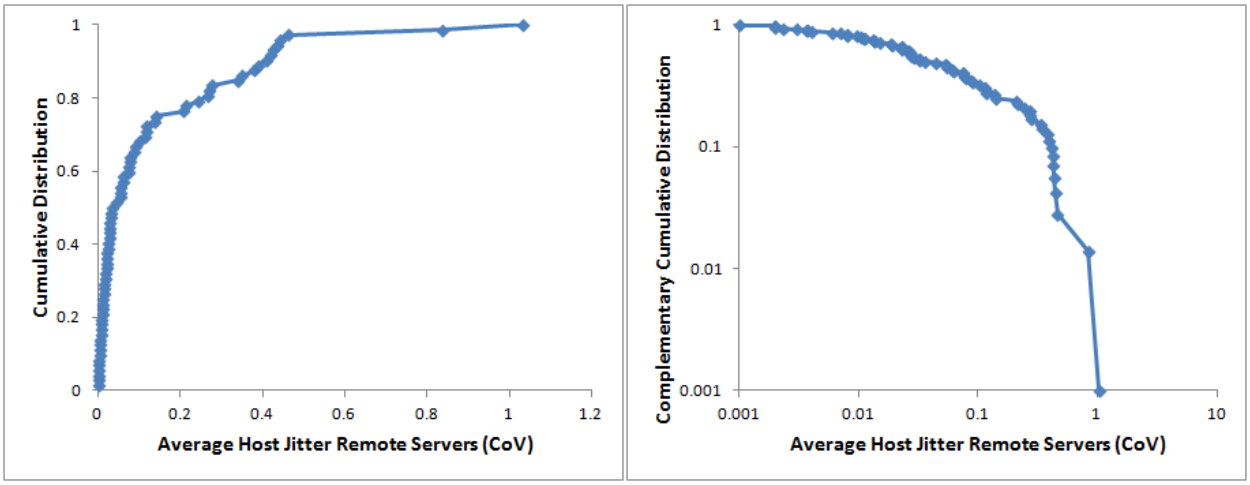

Figure 27. CDF and CCDF of Average Host Jitter for Local Servers

Min	Max	Mean	Median	Standard Dev.
0	1.03	0.13	0.04	0.2

Table 19. Breakdown of Average Host Jitter for Local Servers (CoV)

Figure 27 shows the average jitter over local servers for each host as a cumulative distribution and a complementary cumulative distribution. Table 19 shows a breakdown of the jitter averages consisting of the minimum and maximum values, the mean, the median, and the standard deviation. The jitter averages are measured as coefficient of variation (CoV), which is the standard deviation divided by the mean, and is unitless. For remote servers, just as with all servers and local servers, the jitter averages are skewed to the lower end, with a few values jumping out. Averages for remote servers have the greatest range of the three analyses; however, they have a noticeably lower mean than the other two. So connections to remote servers are likely to have less jitter, but can occasionally result in greater jitter.

5.4. Server-Centric Analysis

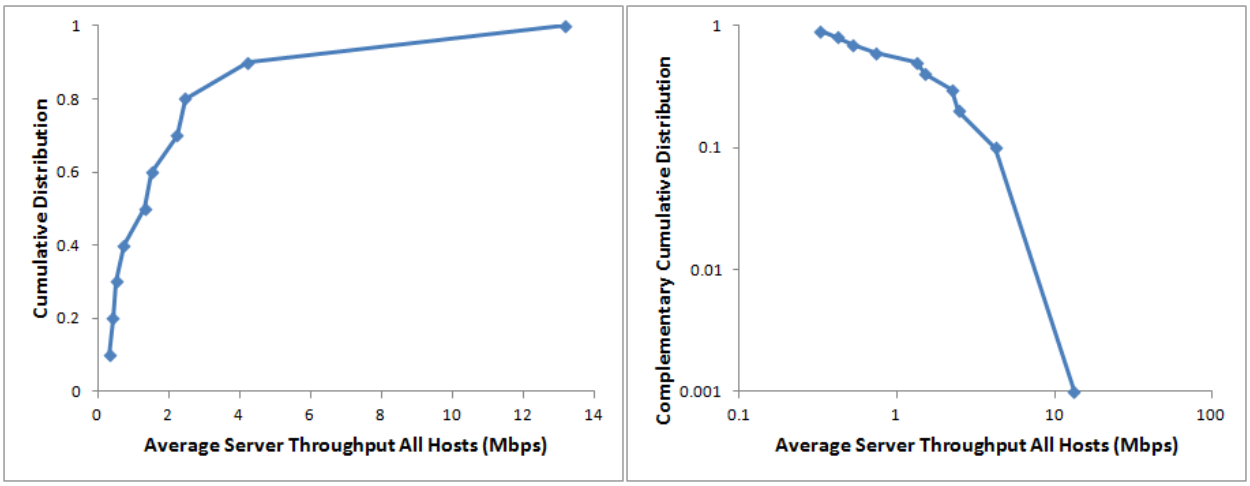

Figure 28. CDF and CCDF of Average Server Throughput for All Hosts

Min	Max	Mean	Median	Standard Dev.
0.33	13.21	2.69	1.41	3.88

Table 20. Breakdown of Average Server Throughput for All Hosts (Mbps)

Figure 28 shows the average throughput over all hosts for each server as a cumulative distribution and a complementary cumulative distribution. Table 20 shows a breakdown of the throughput averages consisting of the minimum and maximum values, the mean, the median, and the standard deviation. The RTT averages are in Megabits per second. Most servers performed at less than 3 Mbps, but one was at over 4 Mbps, and one, the WPI origin server, averaged over 13 Mbps.

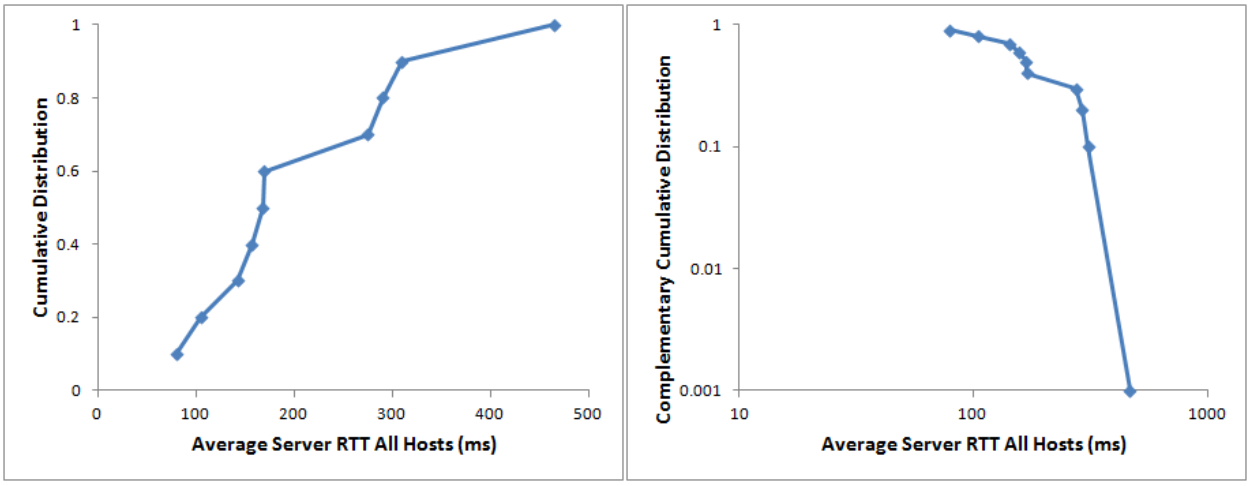

Figure 29. CDF and CCDF of Average Server RTT for All Hosts

Min	Max	Mean	Median	Standard Dev.
79.24	464.98	215.76	168.19	117.55

Table 21. Breakdown of Average Server RTT for All Hosts (ms)

Figure 29 shows the average round-trip time (RTT) over all hosts for each server as a cumulative distribution and a complementary cumulative distribution. Table 21 shows a breakdown of the RTT averages consisting of the minimum and maximum values, the mean, the median, and the standard deviation. The RTT averages are in milliseconds. The majority of servers stayed below 200 ms, but a small group averaged at around 300 ms, and one, the Japan server, averaged over 450 ms.

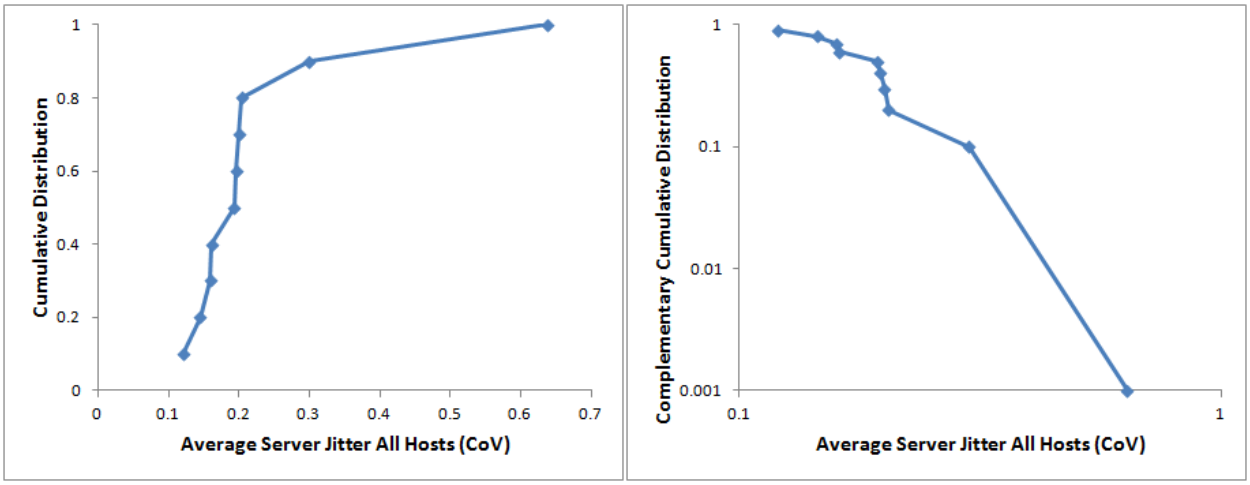

Figure 30. CDF and CCDF of Average Server Jitter for All Hosts

Min	Max	Mean	Median	Standard Dev.
0.12	0.64	0.23	0.19	0.15

Table 22. Breakdown of Average Server Jitter for All Hosts (CoV)

Figure 30 shows the average jitter over all hosts for each server as a cumulative distribution and a complementary cumulative distribution. Table 22 shows a breakdown of the throughput jitters consisting of the minimum and maximum values, the mean, the median, and the standard deviation. The jitter averages are measured as coefficient of variation (CoV), which is the standard deviation divided by the mean, and is unitless. Most servers had average jitters between 0.1 and 0.2, with one at around 0.3, but one, the Minnesota server, averaged over 0.6.

6. Conclusions and Future Work

Average throughput was greater overall for local servers than remote servers. Throughput for all servers was much higher, but that probably has to do with including the WPI origin server in that group. Round-trip time was larger for remotes servers and smaller for local servers, as expected. Jitter actually seemed to be smaller in general for remote servers, though they also had the largest jitter value, so the range of values is higher. The WPI origin server stood out as having the greatest average throughput by far at 13.21 Mbps. The Japan server had the largest average round-trip time at 464.98 ms, and the Minnesota server had the most jitter by far with a coefficient of variation of 0.64.

It might be interesting to do a more in-depth survey analysis, that connects responses to different but related questions, such as time spent playing online games and interest in testing network connection, in order to discover any correlations between the them.

As for the feedback from the survey, one suggestion was to redesign our Website. We could try different layouts, such as placing the network performance results next to the game so that both could be seen. We could also try to make the site look more polished and professional. Any reported flaws with the games should be addressed, such as Tetríssimus not handling the setting of pieces to allow a final movement to the side. Also, clickable button controls for Tetríssimus and similar games could be added for mobile device users. As it stands, there is currently no way to play Tetríssimus without a keyboard. There are many options for new games to add, and plenty of resources for open source JavaScript games that can be utilized. We might expand to hosting different formats rather than only JavaScript games, since there is no interaction between the game and the network tests. We should also consider adding upload throughput to our list of measurements, which we may be able to achieve using XMLHttpRequests. Packet loss was also suggested, though not as feasible to implement.

Seeing as how someone managed to put BUM as the top score with SEX right below it on one of our high score tables, perhaps some care should be put into the prevention of cheating. Cheating is currently possible through manually modifying the JavaScript variables for each game. This could be handled by keeping the scores calculated on the server, requiring logins to submit scores, or perhaps updating the high score tables to only show weekly top scores. Given the scope of the Website, however, it may be deemed acceptable to leave as is, since potential for abuse is minor and solutions may cost more time, effort, or resources than they are worth.

What should perhaps be the focus going forward is displaying network results in a more comprehensive, user-oriented manner. Users might not be able to understand how throughput etc. would affect different network uses such as VOIP or online gaming. Making network performance a more approachable topic would help garner interest in it, which would benefit both our project and the future of network performance in general.

7. References

1. Broadband reports.com Speed Test, http://www.dslreports.com/stest/.

2. Deep Internet Performance Zoom, http://dipzoom.case.edu/.

3. Gomez Peer Community, http://www.gomezpeerzone.com/.

4. The DIMES project, http://www.netdimes.org/.

5. K. Claffy, M. Crovella, T. Friedman, C. Shannon, and N. Spring, “Community-oriented network measurement infrastructure (CONMI) workshop report,” *SIGCOMM Comput. Commun. Rev.*, vol. 36,

no. 2, pp. 41-48, 2006.

6. Y. Hyun, “Archipelago Meas. Infrastructure,” *CAIDA-WIDE Workshop*, 2006.

7. L. Peterson, A. Bavier, M. Fiuczynski, and S. Muir, “Experiences Building PlanetLab,” in *USENIX Symposium on OSDI*, Seattle, WA, 2006.