Effects of Common Building Construction Materials on TCP Throughput, FTPS and HTTP Response Time

Patrick Vest
Computer Science
University of Alabama
Tuscaloosa, AL
pcvest@ua.edu

Tyler Conger
Computer Science
University of Alabama
Tuscaloosa, AL
tconger1@crimson.ua.edu

Jacob Pearson
Computer Science
University of Alabama
Tuscaloosa, AL
japearson5@crimson.ua.edu

ABSTRACT

In this paper, we will be analyzing how well wireless network signals propagate through various common construction materials and the materials' effect on different network protocols. This analysis will be conducted by measuring the impedance of a wireless signal after enclosing a receiving device in a box built from various common construction materials. These tests will determine how well the signal is able to pass through the material box. We will be measuring the TCP throughput as well as the number of packets that were dropped, out of order, or corrupted during transmission. We also will be testing the response time of an HTTP request between the two devices, as well as response time for FTPS; the time it takes to transfer a file between the two devices. The various materials that will be used include drywall, brick, and an aggregate of these materials, including electrical conduit to represent the walls inside of a residence or commercial property. The various traffic between the two devices: the receiver; encased a material box, and the router; left in open air, will be measured using tools such as Wireshark, JPerf, and iPerf. The total percentage number of damaged, unordered, or lost packets, as well as the bandwidth with each different material will be compared to a control test where no material box was used, and the router and receiver had a direct line of sight allowing them to easily transmit data without significant interference between the two.

KEYWORDS

Bandwidth – The theoretical maximum amount of data that can be sent through the network link per second

Brick – Small rectangular blocks of fire dried clay that are used as a building material.

Drywall – A type of wood board made from wood pulp and various other materials, commonly used in interior walls of homes and buildings.

FileZilla - a free and open-source, cross-platform FTP application, consisting of FileZilla Client and FileZilla Server. Both server and client support FTP and FTPS.

FTPS – File Transfer Protocol Secure, a secure extension of FTP that encrypts data with TLS and allows secure transfer of files between devices across the web.

HTTP – Hyper Text Transfer Protocol, an application layer protocol used for data communication across the web.

Insulation Foam – Our insulation foam used was Pink Panther hard cell rigid insulation foam. The same kind that is used within interior and exterior wall insulation projects.

JPerf – A Java-based GUI wrapper for the iPerf software developed to increase the ease of use of iPerf relative to running iPerf from the command-line of the host operating system.

Laser Distance Measure – A device that emits a laser beam out of one side and is able to measure the distance from the device to the end of the laser over long distances.

Linux – An open-source operating system based on the Linux kernel, many different versions have been released and remade.

Mock Exterior Wall – The mock exterior wall, or simply mock wall, is a construction created to mimic the effects and composition of that of a real wall. The mock exterior wall will be created with a wooden box built specifically to house the Raspberry Pi computer. Surrounding the wooden box is a slightly larger box built from Pink Panther hard-celled insulation foam. This foam box is then placed behind a mock wall of bricks. Through these bricks will be a section of electrical conduit. All of these characteristics were created to mimic that of a real exterior wall.

Raspberry Pi – A small single board computer that allows all the functionality of a larger computer in a smaller and compact space, running a Linux based operating system.

TCP Drop Rate – TCP drop rate, while not formally defined by the TCP protocol, for the purposes of this report, means any segment the percentage of segments that arrived corrupted or out of order in each given TCP session.

Throughput – The measured amount of data that is being sent through the network link per second

Transmission Control Protocol (TCP) - A secure protocol that ensures that all packets are delivered in-order

iPerf – An open-source utility for transmitting TCP packets from one host running the software in client mode and another host running the software in server mode to receive traffic.

1 Introduction

In modern homes and offices, various types of materials are used both on exterior walls as well as interior walls, and these materials can create varying amounts of network signal impedance. This can cause instability in the network connection and dead zones throughout a building that can be frustrating for internet consumers. As the world has become increasingly interconnected, having a strong wireless internet signal in every room has become more and more important and thus the need has arisen to use building materials in homes and offices that will allow consumers to stay connected without facing the issues caused by dead zones. Testing the effects of obstructions caused by commonplace materials that are used in everyday construction is a necessary step to ensuring seamless wireless connectivity. In fact, this topic has been highly researched in the past. This study aims to analyze the impact of these materials on individual network protocols, as some protocols have more stringent requirements than others regarding elements such as bandwidth and delay. This knowledge is useful in determining placement of key devices such as a router or wireless access point, as placement of one of these devices behind a signal blocking material can impede the device's ability to reach maximum usage. In addition, it is important to determine what protocols should be used on a network that could experience high impedance.

To test the amount of signal degradation that is caused by these construction materials, we will be using various software tools that will be running on both the receiver and the transmitter to monitor the rate at which data is able to flow between the two devices, also known as bandwidth or throughput. We will also be measuring the percentage of data packets that are dropped, sent out of order, or corrupted, which indicates that the information being transmitted over the wireless signal is having more difficulty being sent. Using this information, we can compare each of the materials to one another to understand which materials are the least impactful to signal degradation and the ones that are the most damaging to the signal. The software tool that will be used for this detection is Wireshark which is able to identify packets that are sent out of order, lost, corrupted, or otherwise need to be retransmitted. To transmit and receive traffic between the two devices we will be using the tools JPerf which is a GUI (Graphical User Interface) extension of iPerf, a tool used for network performance measurement, FileZilla FTP server, and an HTTP server that we created in this course.

The hardware devices that will be used to conduct these tests include a Raspberry Pi 3b and a Linksys AC1750 v2. In our testing environment the Raspberry Pi 3b will be representing a typical user who may be receiving all different kinds of traffic through the web, such as when a user streams video or requests a web page. The second device is a laptop which will be sending data to the Raspberry Pi device. This traffic will be monitored and tracked by the aforementioned software for recording our data and providing interface for us to see and analyze the results in real time.

2 Procedure

We strictly followed these steps during our testing procedures to obtain our results:

Hardware Setup Procedure:

1. A Raspberry Pi 3b will be used as the server for our tests, and a laptop will be used as the client. The Raspberry Pi will run Debian Linux, and the laptop will run Windows 11 22H2.
2. Two tables will be placed 40 meters apart on opposite ends of an obstruction-free hallway. The 40-meter distance will be measured using a laser distance measure to ensure accuracy.
3. On one table will be the client device hardware. This includes the Linksys AC1750 v2 router, connected via a Cat 5e Ethernet cable.
4. On the opposing table will be the server device hardware. This includes the Raspberry Pi which will be connected to the 2.4 GHz band of the Linksys router. The Raspberry Pi will be operated using a wired mouse and keyboard in order to reduce any possible wireless interference.

Software Setup Procedure:

5. For the software, identical versions JPerf, and Wireshark 4.0.0 will be installed on the Raspberry Pi and Windows laptop.
6. For the FTPS trials, FileZilla will also be installed on the laptop, in order to host an FTPS server.

Setup Procedure for Each Obstruction Material:

7. For the control test group, trials will be conducted with no obstructions surrounding the Raspberry Pi, and no obstructions between the Raspberry Pi and the router.
8. For the brick test group, trials will be conducted with one layer of bricks encasing the Raspberry Pi computer on all sides.
9. For the drywall material test group, trials will be conducted with one layer of drywall encasing the Raspberry Pi on all sides.
10. For the mock exterior wall test group, trials will be conducted by encapsulating the Raspberry Pi on all sides in three layers of construction materials. The Raspberry Pi will be placed in a wooden box constructed of plywood. Then the plywood box will be surrounded by a box created of hard cell insulation foam layer. Finally, the layer of insulation will be surrounded by an outer layer of bricks. Within the bricks there will be a length of electrical conduit running through them to mimic that of a wall found in an actual building or home.

TCP Trial Procedure:

11. For each construction material surrounding the Raspberry Pi, 10 individual trials will be conducted. Each trial will consist of the client sending 128 Megabytes of data over a raw TCP connection.
12. JPerf will be run on the Raspberry Pi in server mode with a TCP window size of 0.060 Megabytes, with the maximum segment size determined by the operating system(s). With the maximum segment size unchanged between trials, it can and will be used as an independent variable through our tests; as we are more concerned with corrupt or undelivered segments than the speed of segment delivery.
13. JPerf will be running on the laptop in client mode, and JPerf will be running on the Raspberry Pi in server mode. Then the laptop will connect with the Raspberry Pi through JPerf.
14. Wireshark will be running on the Raspberry Pi to track the TCP messages sent and received from the server.

HTTP Trial Procedure:

15. For the HTTP trials, we will be conducting 20 trials for each construction material surrounding the Raspberry Pi. For each of these trials, we will load an image file of size 4,803 Kilobytes over a web server. This web server is going to be hosted by the laptop using webServerV2.py from Project 2.
16. Wireshark will be running on the Raspberry Pi to capture packets sent to and from the Raspberry Pi and laptop. This is necessary to calculate the response time.
17. Once the image finishes loading on the web page on the Raspberry Pi, the Wireshark capture will be stopped.
18. The response time will be calculated using the Wireshark record. We will calculate the difference between the time the HTTP GET request was sent and the time of the HTTP 200 OK message.

FTPS Trial Procedure:

19. For the FTPS trials, we will be conducting 10 trials for each construction material surrounding the Raspberry Pi.
20. On the laptop, FileZilla will be used to host a server on the laptop.
21. On the Raspberry Pi, FileZilla client will be used to connect to the laptop server.
22. For each trial, Wireshark will be running in the background on the Raspberry Pi to capture packets sent to and from the Raspberry Pi and laptop. This is necessary to calculate the response time.
23. A file of size 193 Megabytes, which is on the Raspberry Pi, will be uploaded to the server.
24. After the file finishes uploading the Wireshark capture will be stopped, and the file will be deleted from the server before starting the next trial.
25. The response time will be calculated using the Wireshark record. We will calculate the difference between the time of the first packet of data sent and the time of the HTTP 200 OK message.

3 Results – Raw TCP

3.1 Control

The first group of trials that were conducted were a series of TCP transfers with a size of 128 Megabytes each between the two devices. In total, ten trials were conducted to get an average throughput and drop rate percentage of the connection from the TCP client to the TCP server.

Below are charts of the results of the raw TCP session with no signal obstruction. Figure 1 shows the throughput of each of the ten trials. The dotted red line represents the average throughput of the trials. The second chart shows the TCP drop rate of each individual trial. The dotted red line represents the average drop rate of all ten trials.

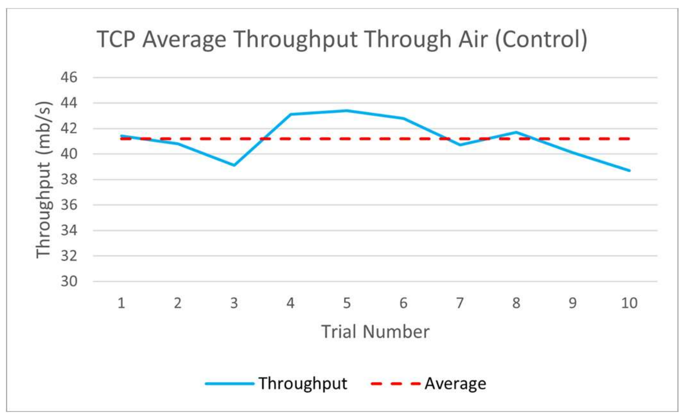

Figure 1 - The average throughput of the TCP connection through a hallway with no impedance. The average line is pictured in red and the results from this graph will be used as a comparison to all other tests.

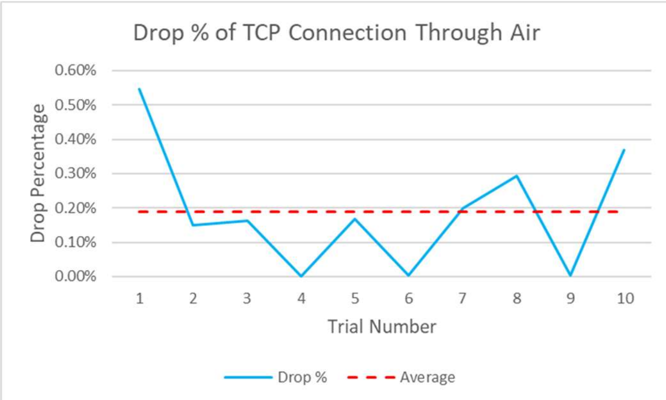

Figure 2 - Comparing the percentage of packages that were dropped, damaged, or arrived out of order for each individual trial. This covers each trial for the control, no blocking material to cause interference.

3.2 Bricks

Similar to the control, a series of TCP transfers of 128 Megabytes of data were sent through the brick encasement. Ten trials were conducted to get an average TCP drop rate value and throughput of the connection from the TCP client to the TCP server while the device was encased bricks.

Below are charts of the results of the raw TCP session with bricks acting as a signal obstacle between the router and the receiving host. The first chart shows the throughput of each of the ten trials with the dotted red line representing the average throughput. The second chart shows the TCP drop rate of each of the ten trials with the dotted red line representing the average drop rate.

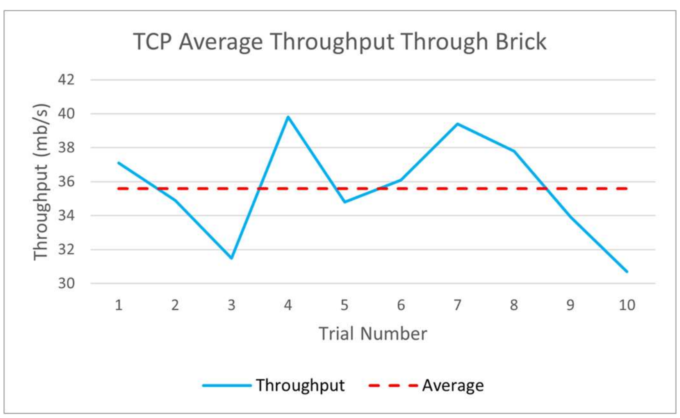

Figure 3 - The average throughput of the TCP connection through a brick obstruction. The average line is pictured in red, and the per-trials results are shown on the blue line.

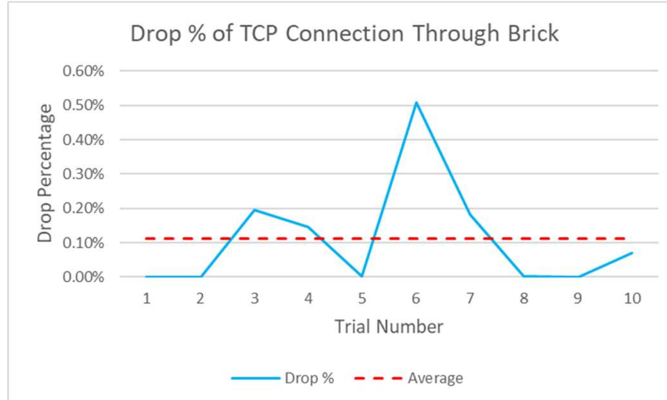

Figure 4 - Comparing the percentage of packages that were dropped, damaged, or arrived out of order for each trial. This covers each trail for the brick material.

3.3 Drywall

As seen in previous TCP transfers, we are sending 128 Megabytes of data through the drywall surrounding the Raspberry Pi device. We conducted ten separate trails of this to get an average value for the drop percentage and for the throughput of the connection from the client to the server.

Below are two graphs that showcase the results of the TCP session with drywall acting as a signal blocker in between the router and the receiver. In each figure, we have a dotted red line representing the overall average of the throughput and drop rate, respectively throughout all trials. The first chart shows the average throughput, and the second chart shows the drop rate of each of the ten trials.

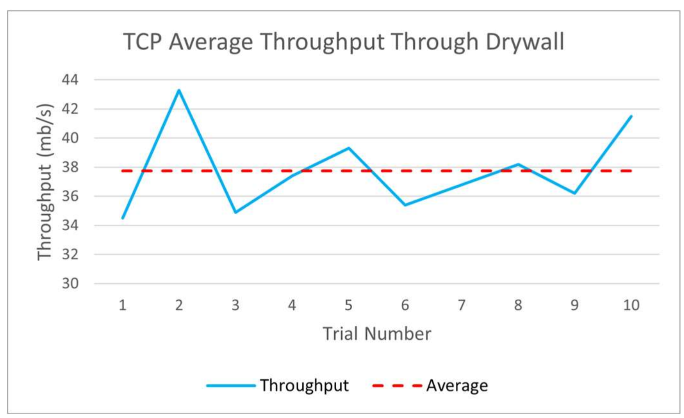

Figure 5 - Comparing the total throughput in megabits per second to the current iteration of the trail we are conducting, this data was captured while the Raspberry Pi was surrounding with a box of drywall to imitate a wall in a home or other building.

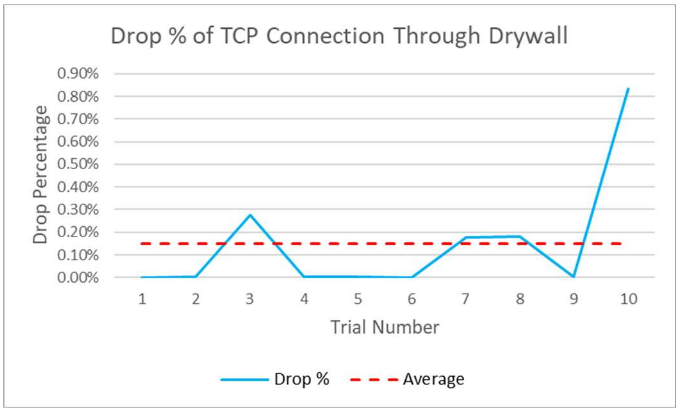

Figure 6 - Comparing the drop percentage of each trial over the total of ten trials conducted. This is through drywall which was surrounding the Raspberry Pi device in order to imitate a blocker that may occur in a traditional building.

3.4 Mock Exterior Wall

As we see in the previous trials, we send 128 Megabytes of data with the aforementioned multi-material box that is representative of a mock exterior wall separating the Raspberry Pi computer device from the router. With this setup, we conducted ten trials and recorded the drop packet percentage and the average throughput in megabits per second.

Below we can see the two graphs show the results of the trials. We have a dotted red line that is representative of the overall average after all the trials were conducted. The first of the two charts show the average throughput in Megabits per second and the secondary chart shows the drop percentage of packets dropped in each trial.

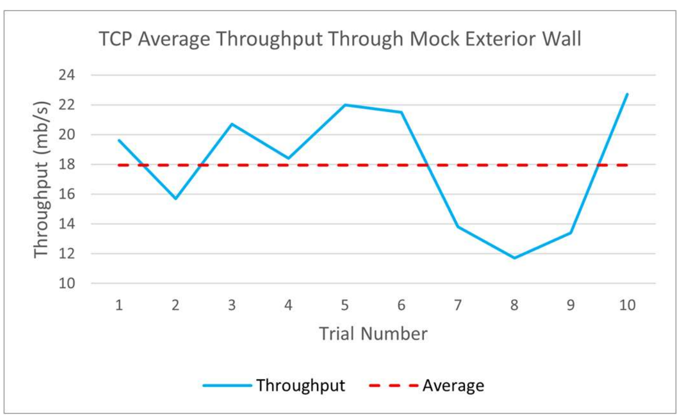

Figure 7 - The throughput in Megabytes per second for each of the 10 trials. Shown in the graph is the average throughput: also represented with a dotted red line. These trials were conducted while the Raspberry Pi was surrounded by the mock exterior wall box.

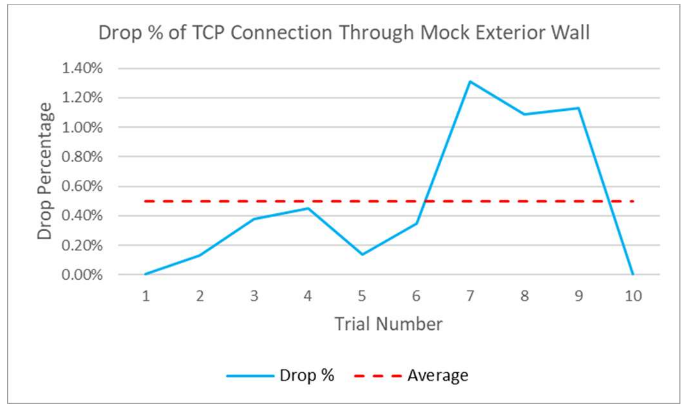

Figure 8 The drop percentage of each trial over the total of ten trials through the mock exterior wall. The mock exterior wall surrounded the Raspberry Pi computer in order to mimic the walls of a traditional building and all the materials that would be within those walls.

4 Analysis of TCP

4.1 Control - Raw TCP

According to Figure 1 and Figure 2, the average drop rate for the control trials is 0.19%, and the average throughput for the control trials is 41.18 Megabits per second. We expect these values to be the greatest compared to the other trial results because the signal was traveling directly between the client and host with no physical obstructions to impede the signal. A signal with physical obstructions would see a decrease in throughput, and an increase in corrupted packets or packets delivered out of order. These results will give us an accurate baseline to compare the effects that various physical obstructions have on the signal.

4.2 Bricks - Raw TCP

As can be seen in both Figure 3 and Figure 4, the average drop percentage for the TCP test through bricks is 0.11%, and the average throughput for TCP test through the bricks is 35.6 Megabits per second. Unsurprisingly, both the average throughput and average drop rate are worse than the corresponding measurements from the control tests. This higher drop percentage showcases that the brick structure around the Raspberry Pi degrades the wireless signal to the point that some of the packets are dropped. Given the thickness and density of a common brick, this result is within the bounds of our expected values for the brick tests.

4.3 Drywall - Raw TCP

Figure 5 and Figure 6 showcase the average throughput and average drop rate of packets respectively. As we see in Figure 5 above, the throughput through the drywall is on average 37.75 Megabits per second. This is generally in line with results from other tests as well as the data that we expected. We can see that the control has only a slightly higher throughput: of around 41.18

Megabits per second, we would expect this to be higher as there was no obstruction that could have blocked signal. It is interesting that compared to the brick test, which had a throughput of 35.6 Megabits per second, the drywall had an overall throughput of 37.75 Megabits per second. As can be seen in Figure 6 we are able to see that drop rate starts off low in fact having some trials with zero or near zero dropped packets however a significant drop in the later trials shows that the drywall material was able to cause some significant interference with the signal between the two devices. Granting us an overall average drop rate of about 0.15% overall. However, while this is slightly higher than the drop rate through brick material it is worth noting that the average was significantly affected by a high drop rate in the final trial, discarding the outliers to the data gives us a much more expectant average of around 0.07% which falls in line with previous estimations and along with other conducted trials of control and through brick material, allowing us to see that while there is some disruption caused by drywall it is less than that of a brick material and more than no material, the control.

4.4 Mock Exterior Wall - Raw TCP

As shown Figure 7 and Figure 8 the average throughput for the mock wall exterior is 17.95 Megabytes per second. Figure 8 also shows that the average drop percentage is 0.50% in terms of number of average packets dropped or damaged. We would expect this value to be higher as it includes an assortment of different materials instead of just a singular one that could cause signal loss. By analyzing the data, we can see that this value is higher than the drop rate in other trials of different materials. In terms of throughput the overall average was 17.95 Megabytes per second, which is significantly lower than that of previous trials. We would expect the throughput to be lower as there are now a multitude of materials that are causing signal blockage. This showcases just how much being on the opposite side of a wall from an access point can cause deterioration in signal.

5 Results – HTTP

5.1 Control

For the HTTP Control trials, we measured the response time between the Raspberry Pi's first HTTP GET request and the HTTP 200 OK message sent from the web server. Once again, the control trials were conducted to test the signal strength in an environment with no obstructions between the client and server. This data is showcased in Figure 9 below, with the dotted red line representing an average response time of 1.073 seconds.

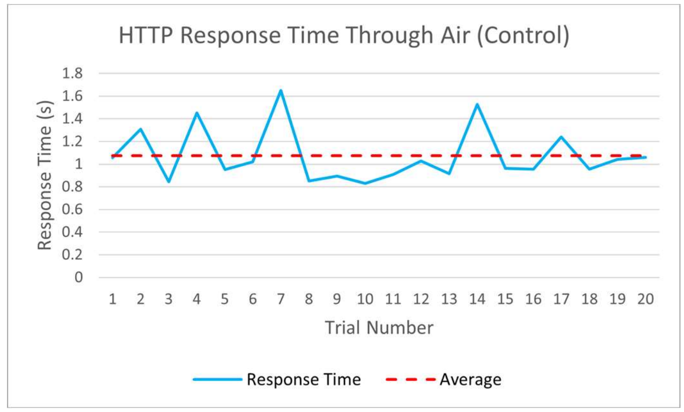

Figure 9 - Showcasing the HTTP response time over 20 individual trials in an environment with no obstructions. The dotted red line represents the average response time for all 20 trials.

5.2 Brick

For the HTTP response time we measured the time that it took for the Raspberry Pi to request an image from a web server, and then the time the Pi received the entirety of the file. This trial was conducted with the Raspberry Pi encased in bricks, following the same procedure as documented in Section 3.2.

Below is the chart of the results of the HTTP response time with bricks acting as a signal obstacle between the router and the receiving host. The red, dotted trendline represents the average drop rate.

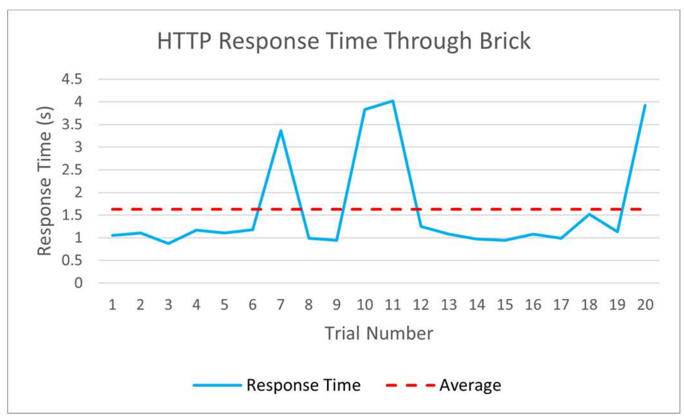

Figure 10 - Showcasing the HTTP response time over 20 individual trials through the brick box that was constructed around the Raspberry Pi. The red dotted line is representative of the overall average over all the trials.

5.3 Drywall

For the drywall round of testing for HTTP response time we set up a similar process that was described in the procedure to allow our device to be separate from the router and encased in the specified material, in this case a drywall box. For these rounds of testing, 20 HTTP connections were made and the response times were measured for sending a 4,803 Kilobyte image.

Below in Figure 11 we can see the results of those 20 trials where Tree.jpeg was sent between the two devices via an HTTP connection. The figure shows that generally this response time is around one second per trial with some trials deviating and having higher times than others.

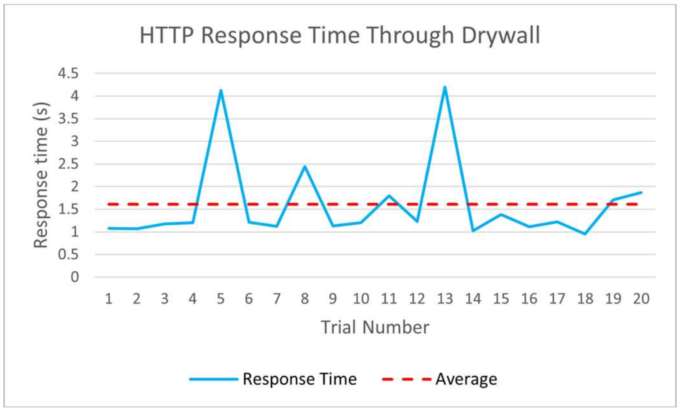

Figure 11 - The HTTP response time over 20 trials through the drywall construction material. The dotted line showcases the average response time overall after the 20 total trials.

5.4 Mock Exterior Wall

Below, in Figure 12 we can see the results of those 20 trials where Tree.jpeg was sent between the two devices via an HTTP connection. The dotted red line is representative of the overall average after the 20 trials were completed, whereas the blue line is representative of the response time in each individual trial conducted.

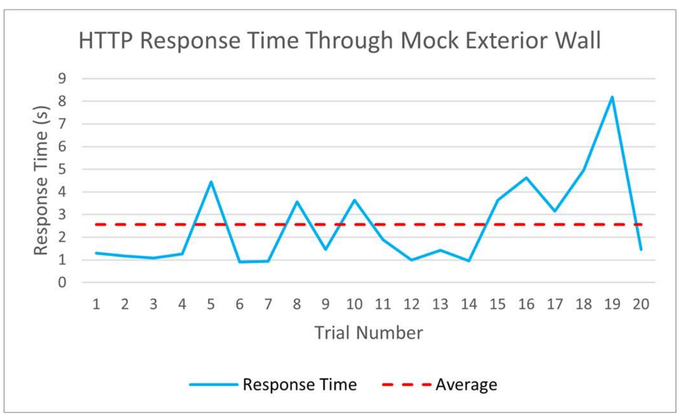

Figure 12 - The HTTP response time through 20 trials with the mock box and exterior wall surrounding the Raspberry Pi, the average value is showcased by the dotted red line.

6 Analysis of HTTP

6.1 Control

Figure 9 shows the results of the HTTP response time for 20 trials with no obstruction between the client and server. The average HTTP response time over the 20 trials is 1.073 seconds. This average response time serves as a base value to compare other results with obstructions impeding the signal.

6.2 Bricks

Figure 10 shows the results of the HTTP response time tests through the brick material. The average HTTP response time over the 20 trials is 1.626 seconds. This falls short of the control result by just over 34%. This shows that brick had a significant impact on the response time of the HTTP server. Consequently, the server must have had to retransmit a more significant portion of its packets that it did during the control experiments.

6.3 Drywall

Figure 11 shows the average HTTP response time in seconds. As we can see the average response time is 1.613 seconds. This is in-line with what we would expect as there is some material that is able to cause disruption in the signal, but the material is thinner and less dense than other materials used comparatively. We can also see that the standard deviation between trials is .942, which is caused by two major spikes in the data. The data is otherwise is fairly consistent throughout the entirety of the 20 separate trials. Compared to the control test, we saw a difference in response time of .540 seconds, which is one that we would expect from such a thin material being inserted between the Raspberry Pi and the router. As increasingly dense materials are added between the router and the Raspberry Pi, we expect to see an increase in slowdowns and overall response time.

6.4 Mock Exterior Wall

Figure 12 shows the results of the 20 trials. The average response time was 2.555 seconds. It should be noted, however, that the 19th trial was an extreme outlier, taking almost double the amount of time that the previous longest trial took. With such outlier excluded, it should be noted that the average response time is decreased to 2.258 seconds, which is marginally closer to the brick trials but not so far away as to constitute a concern for the experimental procedure. Based on this data, it can be concluded that such an exterior wall would have a significant impact on wireless network performance, as the mock exterior wall saw an average response time that was over 2.5 times that of the control.

7 Results – FTPS

7.1 Control

For the FTPS response time, we measured the amount of time it took for the Raspberry Pi to request a binary file from an FTPS server and receive the entire file. These trials were conducted with the Raspberry Pi in an open-air environment, following the procedure documented in Section 3.2.

Below is the chart of the results of the FTPS response time with no obstruction between the router and the receiving host. The dotted red line represents the average response time for all ten trials. The results for this control are incredibly consistent, increasing our ability to make informed and accurate conclusions about other trials.

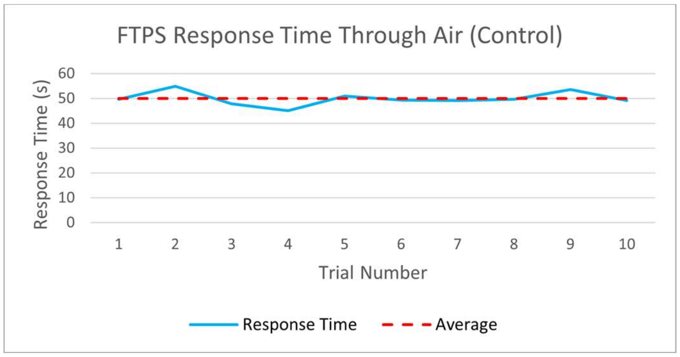

Figure 13 – The time taken for the binary file to be sent in each trial, represented by the blue response time line in seconds. The dotted red line is representative of the average between the ten trials after they were concluded.

7.2 Brick

For the FTPS response time, we measured the time it took for the Raspberry Pi to request a binary file from an FTPS server and receive the entire file. These trials were conducted with the Raspberry Pi encased in bricks, following the same procedure as documented in Section 3.2 and used a similar experimental procedure to the HTTP response trials.

Below is the chart of the results of the FTPS response time with bricks acting as a signal obstacle between the router and the receiving host. The red, dotted trendline represents the response time.

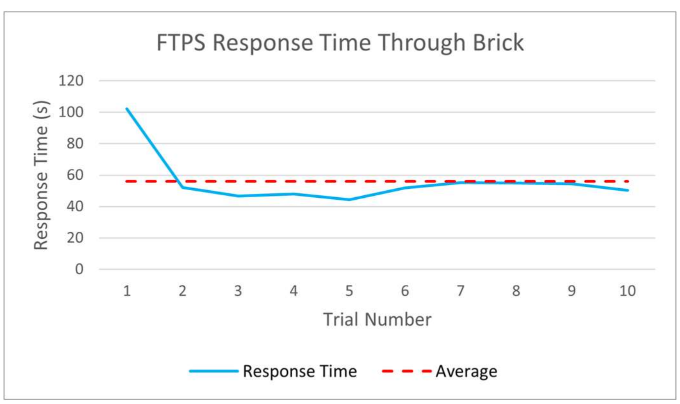

Figure 14 - Showcasing the FTPS response time over 10 trials with a brick box was constructed around the Raspberry Pi. The dotted red line is representative of the average for all the trials.

7.3 Drywall

In this round of testing, drywall was used to surround the Raspberry Pi as previously discussed in the procedure. However, this time we were testing the FTPS results in which a file was transferred between two devices running FileZilla over a wireless connection. Below we can see the figure associated in Figure 15. This showcases the response time for each of the ten trials and has a dotted red line that represents an overall average response time that was representative throughout the ten trials.

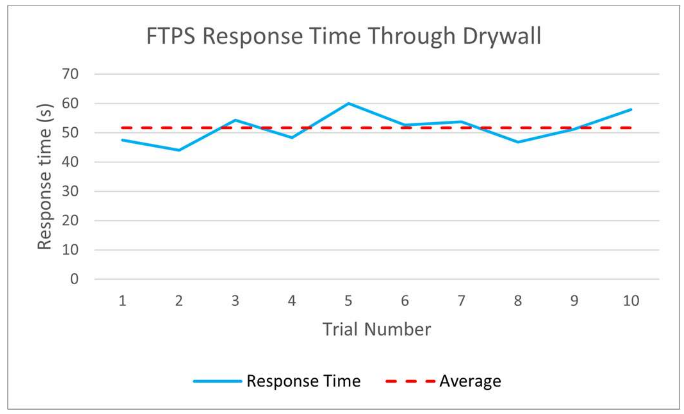

Figure 15 - Showcasing the FTPS response time through a drywall medium which caused interference between the Raspberry Pi and the router. The dotted red line is the average of the ten trials.

7.4 Mock Exterior Wall

For these set of FTPS trials, a mock exterior wall surrounded the Raspberry Pi as described in the procedure portion of the project. This wall included wood, insulation, bricks, and metal piping. As a file of consistent size is transferred between the two devices the response time for each transfer is recorded and represented in Figure 16; below, showcasing the varied times with each individual trial. The dotted red line is representative of the average response time after the ten trials were completed.

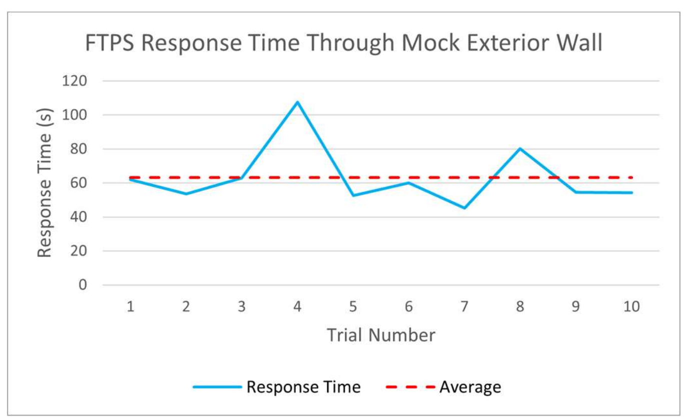

Figure 16 – The response time taking when transferring the specified file between the Raspberry Pi and the router with the mock exterior wall built separating the two devices. The dotted line is representative of the average response time of the ten trials after they were completed.

8 Analysis of FTPS

8.1 Control

Figure 13 shows the results of the FTPS response time tests in an open-air environment. The average FTPS response time over the ten trials with a 193 Megabyte file is 49.919 seconds. This will be used as a base value to compare the results of the other tests. This is the lowest response time compared to the other trials with a physical obstruction impeding the signal.

8.2 Brick

Figure 14 shows the results of the FTPS response time tests through the brick material. The average FTPS response time over the ten trials with a 193 Megabyte file is 56.028 seconds. This falls short of the control result by nearly 11%. This shows that brick had a significant impact on the response time of the FTPS server but not to the degree of significance that it has over the HTTP trials. Recall that the HTTP response time with the brick was 34% slower than the control. This disparity in proportional signal and connection degradation will be discussed in the discussion section.

8.3 Drywall

As can be seen in Figure 15, the average response time is 51.650 seconds. This is generally what would be expected with drywall material blocking the reception of the Raspberry Pi. Overall, the trials in this set followed our average line with no large outliers or unexpected data points that caused error. The data transfer time was fairly consistent, and did not change much between transfers. As we would expect, this response time of 51.650 seconds is only slightly higher than that of 49.919 seconds that the control took to transfer the file. The differences in the averages is only 1.731 seconds. As we would expect, a thin material such as drywall would not present nearly as much of a barrier for signal as would more dense materials or multiple layers of materials such as that described in the mock exterior wall tests.

8.4 Mock Exterior Wall

Figure 16 shows the results of the FTPS response time tests through the mock exterior wall. The average FTPS response time over the ten trials with a 193 Megabyte file is 63.219 seconds. This falls short of the control by nearly 26.643%. This shows that the mock exterior wall had a significant impact on the response time of the FTPS server. The mock exterior wall had a much more significant impact compared to the other obstruction materials during the FTPS trials.

DISCUSSION

Overall, we see that interference caused by the different materials does in fact cause some significant loss of data. However, with modern day networking technology and networking speeds, a singular interfering material blocking signal is not nearly as disastrous to signal quality as one may expect, especially using the 2.4 GHz band. Adding multiple walls and interference points would almost certainly have multiplicative effects where signal is degraded more and more through each wall or interference point. In the past with less powerful networking cards and routers, it may have been expected that signal degradation was more significant when passing through a singular material, as older networking equipment is surely equipped with lower quality receivers and transmitters.

It is also worth noting that each constructed box around the Raspberry Pi device needed a small hole to run cables through, specifically one cable for power to the device, one for output to a monitor setup nearby, and one for keyboard and mouse each to interact with the device. A wired keyboard and mouse were chosen specifically so that the wireless signal from the peripherals would not compromise the experiment results. The area that the cables were run through, while small, could be a factor that caused the experimental values to deviate from the expected values, albeit not significantly.

Lastly, as noted earlier in the report, while certain material performed better or worse than the control, as expected, the percentage of the decrease or increase in performance was not always consistent between protocols. This is likely due to the header information that is bundled with packets in HTTP, for instance, relative to the raw byte stream that is sent over the FTPS connection.

CONCLUSION

From analyses of the data gathered from our standard Internet Protocol testing, we can conclude that various material obstructions on a wireless signal will indeed have a detrimental effect on throughput, dropped packets, and response time. Results from raw TCP, HTTP, and FTPS all show that the control group, which was conducted in an open-air environment, outperformed the obstructed signal tests in every recorded metric. From these results we can also conclude that the mock exterior wall had the greatest impact for each of our tests. Furthermore, the brick wall had the second greatest effect on results. Finally, the drywall had the smallest effect on our test results.

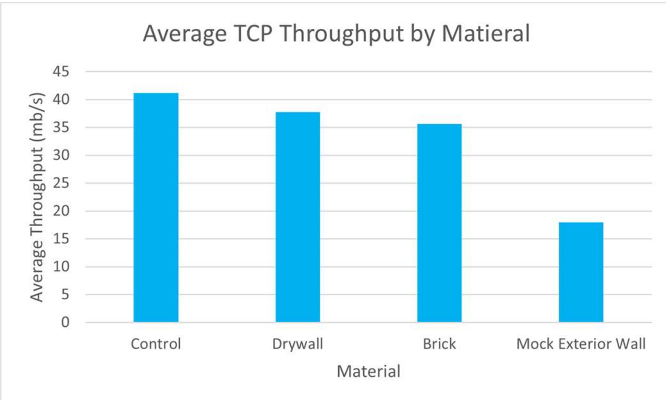

Figure 17 – A bar graph showcasing the average TCP throughput between each of the four separate materials that were tested. Each bar is representative of that materials average TCP throughput after the trials were completed.

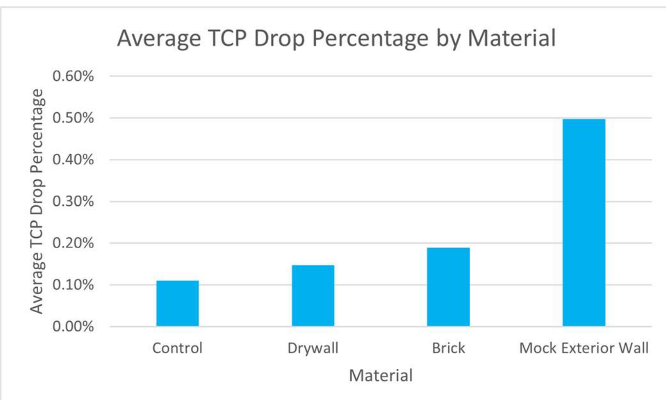

Figure 18 – Comparing the average TCP drop rate between the four separate materials that were tested. Each bar is representative of that materials average TCP drop rate after the trials were conducted.

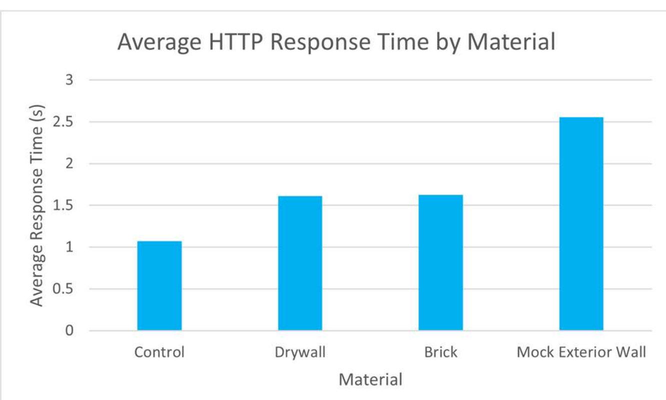

Figure 19 – The average HTTP response time separated by material. Each of the averages is in seconds and each separate bar represents a different material.

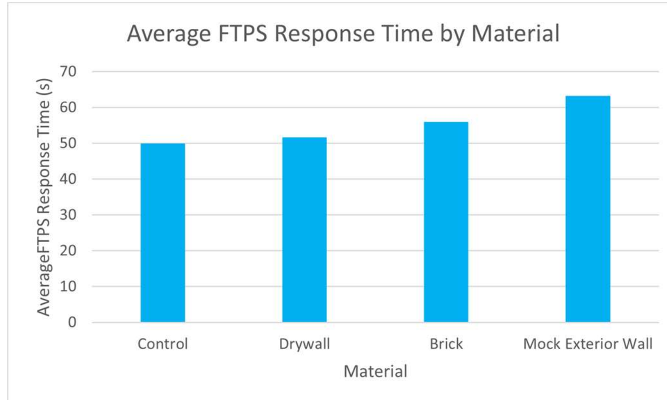

Figure 20 – The average FTPS response time separated by material. The averages are listed in seconds and is bar showcases a different material that was tested.

As shown in Figures 17 through 20, the magnitude of the effects of the different materials followed that of what intuition would suggest: The response times and throughput decrease in the order of control, drywall, brick, and mock exterior wall, while the TCP drop rate increased in the same order. This shows that the thickness of a material plays a role in the connection quality, but, more specifically, the density of the material, as intuition suggests, directly correlated with the connection quality.

This evidence suggests that access point placement is incredibly importing in buildings. After conducting these experiments, two things became clear:

- Placing access points in hallways allows for more interference from interior walls – and occupants of building are typically in the hallway very little of the day. Instead, access points should be placed within working rooms themselves. This will avoid most signal obstruction.
- If wireless internet connectivity is needed outdoors, it is best practice to place a weatherproof access point outdoors, rather than have the wireless signal propagate though the exterior wall of a building.

ACKNOWLEDGMENTS

Thank you to the ACME Brick Tile and Stone Tuscaloosa for providing much needed materials that were used in the construction of the various blocking walls between the router and the Raspberry Pi. Thank you also to Dr. Xiaoyan Hong for providing guidance and inspiration for and throughout this project.

REFERENCES

[1] J.F. Kurose and K. W. Ross, Computer Networking A Top Down Approach, 8th ed. Pearson Education Limited, 2020.