ENSC 427: Communication Networks
Spring 2015

Final Project:

"A decade of advancement: comparing the performance of various applications over 802.11b & 802.11n WiFi using Riverbed Modeler"

http://www.sfu.ca/~jaridw/main.html

Team #13
Vani Choubey (vchoubey@sfu.ca) — 301162616
Henry Hein (hhein@sfu.ca) — 301201424
Jarid Warren (jaridw@sfu.ca) — 301197954

Table of Contents

List of Figures & Tables

1. Abstract

In this report, we aimed to analyze the difference a decade made in Local Area Networks (LAN). Specifically, the first public WiFi standard 802.11b (1999) and the most recent standard available through the simulator we used (Riverbed Modeler) 802.11n (2009). We made a coffee shop scenario with six (6) users to create a load on the router (either 802.11b or 802.11n WiFi). Three (3) of the users downloaded peer-to-peer (P2P) traffic in every scenario and the other three (3) accessed the application to be analyzed: YouTube HD video stream over HTTP, VoIP calling over RTP or File Transfer over RTP. One (1) of the nodes was mobile and 7.5m from the router, the second was fixed at a distance of 5m from the router, and the last one was also fixed and 15m away from the router. In each of the six (6) scenarios (802.11b/802.11n and YouTube/VoIP/FTP) we analyzed average throughput and average delay. Our results proved our goal of showing how much more advanced 802.11n is compared to 802.11b; however, throughput was not significantly different. Average delay was the major difference showing how much user experience was effected. Lastly, we showed that closer distance to the router does not necessarily improve throughput.

2. Introduction

As portable Internet devices (PID) have evolved in the past couple of years, more emphasis has been placed on Internet services to provide: video and audio streaming, the world wide web, File Transfer, in addition to Voice of IP (VoIP) as the availability and price from Internet Service Providers (ISP) improve and decline respectively. Among the protocols to leverage these technologies are Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP) and Real-Time Protocol (RTP). These protocols are often used to implement applications that are accessed primarily through Wireless Fidelity (WiFi) hotspots in the home, on colleges campuses in addition to most public businesses.

We aim to analyze the performance of these applications by comparing the first widely spread version of WiFi, 802.11b, and the most recent iteration available through Riverbed Modeler, 802.11n, to gain a greater perspective in how much these technologies have advanced over the past decade.

To do this, we will create two identical scenarios (one with 802.11n, the other 802.11b) with a server subnet located in San Jose that is connected to a client in Vancouver. The server will provide the client with the particular application data being requested in each scenario: YouTube, VoIP, or File Transfer. The client, will resemble that of a coffee shop with a total of six (6) users connected to the router at a single time.

The three applications were chosen strategically to identify performance differences in both WiFi iterations. YouTube high-definition (HD) streaming was originally introduced in 2009, VoIP gained huge popularity in 2004, and RTP saw its last major revision in 1999. The introduction of

YouTube HD and the newest RTP line up exactly with the introductions of 802.11n and 802.11b respectively; and VoIP lies directly in between. With this foresight, we assumed each technology would handle applications specific to their era better than the other.

Three (3) of those users will maintain a P2P connection throughout every scenario. This serves to essentially load the LAN to further highlight the performance difference achieved with WiFi in the past ten (10) years.

The three (3) remaining nodes will run each of the three applications and will have their average throughput (bit/s) and average delay (s) observed and compared. Two of the nodes will be fixed, where one (1) is five (5) meters from the router and the other is fifteen (15) meters away. The last node, will be mobile using Riverbed Modeler's random waypoint algorithm and start 7.5 meters away from the router.

To summarize: three (3) different applications (YouTube, VoIP, File Transfer), two (2) parameters (average throughput and average delay) and two (2) different versions of WiFi will be observed for a total of six (6) scenarios and twelve (12) graphs to analyze performance.

2.1 Fundamental Concepts
2.1.1 802.11 WLAN or WiFi
WiFi or Wireless Fidelity is a certification assigned to group of wireless LAN devices that follow IEEE 802.11 standards to connect with each other in a relatively small geographical area, such as home or office (as shown in figure 1). This kind of wireless technology allows its user to conveniently and quickly access resources from the Internet [2] [3].

802.11 standard was introduced to world of networking back in 1991. It is a set of specifications defined for Medium Access Control and Physical Layer (PHY). 802.11 PHY provides services like FHSS, DSSS and infrared PHY, with a 1 Mb/s. 802.11 MAC on the other hand, provides carrier sensing multiple access with collision avoidance (CSMA/CA) service. After 802.11 release, different amendments were released in order to get better performance [1]. The most recent amendment available through Riverbed is 802.11n which will be comparing to the first widely adopted version, 802.11b, in our project.

The 802.11b standard was first released in 1999 and introduced on the market with the Apple iBook. This standard providing users with speeds of up to 11 Mbits/s via infrared signals in the 2.4 GHz band using the Complementary Code Keying (CCK) modulation scheme[1]. The standard also took advantage of the aforementioned CSMA/CA as the MAC method. Although some of the channel's total capacity is sacrificed with this method, reducing speeds to about 5.5 Mb/s, reliability is ensured under poor environmental conditions with the Transmission Control

---

[1] http://en.wikipedia.org/wiki/IEEE_802.11b-1999

Protocol (TCP). With the best effort service User Datagram Protocol (UDP), speeds can go up to 7.1 Mb/s with header overhead. 802.11b was replaced by 802.11g in 2003 that saw increases in average throughput and reductions in price.

A decade after 802.11b in 2009, 802.11g was replaced by the third major revision of WiFi: 802.11n. This new WiFi protocol took advantage of multiple antennas to increase average throughput to up to 600 Mb/s. With multiple-input multiple-output (MIMO), frame aggregation, security improvements and dual bands (2.4 and 5 GHz) 802.11n was the single largest improvement to the WiFi standard since its inception.

Figure 1: 802.11 WLAN. [7]

2.1.2 Transport Protocols

The various applications we'll be testing over WiFi utilize one of three major protocols: Hypertext Transfer Protocol (HTTP) for web pages (including YouTube video playback), File Transfer Protocol (FTP) to move large blocks of data, and Real-Time Transfer Protocol (RTP) to implement applications such as VoIP and audio from commercial streaming services.

Defined by the Internet Engineering Task Force (IETF) in 1999, HTTP "is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web."[2] Hypertext is made in such a way that pages are accessible via links ("hyperlinks"), and HTTP is the protocol used to transfer hypertext from one Internet Protocol (IP) node to another. Although HTTP is usually built on top of TCP, it can also be used as a "best-effort" service on top of UDP. Both versions use IP as the network layer to send packets back and forth.

---

[2] http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

HTTP operates in a "request/response" fashion between the server (requested page) and client (web browser). The server responds to requests by sending Hypertext Markup Language (HTML) documents that are displayed on the client's web browser; HTML can include photos, video or text. Uniform Resource Locators (URLs) are used to find and access servers for request.

Originally introduced in 1971, FTP "is a standard network protocol used to transfer computer files from one host to another host over a TCP-based network."[3] FTP, like most protocols, works on a client-server model that requires authentication to initiate (such as a login with password); however, it uses separate control and data connections for the client and server. Unlike HTTP, FTP requires a TCP connection for each transfer, making it much slower although more secure.

FTP was originally made to be used on a command-line interface and was shipped on UNIX, Linux and Windows operating systems. Today, it is built into many productivity applications in addition to servers, mobile devices, computers and even hardware.

Lastly, RTP which was published in 1996 by the IETF, is used to push end-to-end real-time streaming media over IP networks. "RTP facilitates the transfer of real-time data. Information provided by this protocol include timestamps (for synchronization), sequence numbers (for packet loss and reordering detection) and the payload format which indicates the encoded format of the data."[4] Unlike the aforementioned protocols, RTP mostly utilizes UDP as the transport layer to ensure readiness over reliability. For this reason, TCP is not used as the connection establishment in the protocol slows delivery. TCP was designed to support an array of multimedia protocols including H.264, MPEG-4, MJPEG, MP3 and MPEG with the ability to add additional formats without rewriting the protocol.

RTP also employs a sub-protocol, Real-Time Transport Control Protocol (RTCP) which specifies Quality of Service (QoS) feedback in addition to synchronization between media streams. Overall, RTCP accounts for approximately 5% of overall RTP traffic.

2.2 Riverbed Modeler Implementation

The overall network topology and the server subnet setups are identical for both versions of WiFi. However, the configuration for client subnets are different. By referencing [5], we created the network topology model, shown in Figure 2. The network topology consists of the IP Network in Portland located between the server subnet in San Jose and the client subnet in Vancouver. Server and client subnets are connected to the IP cloud using 45 Mb/s, Digital Signal 3 (DS3) coaxial cables.

---

[3] http://en.wikipedia.org/wiki/File_Transfer_Protocol

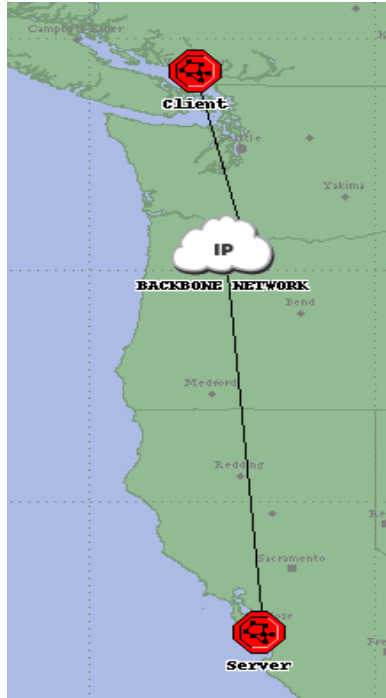[4] http://en.wikipedia.org/wiki/Real-time_Transport_Protocol

Figure 2: Overall network topology.

2.2.1 Server Layout

Inside of the server subnets as shown below in Figure 3, we imagine a scenario in an office environment where the server, local client, and router are connected by 100 Mb/s (100BaseT) twisted pair Ethernet cables. The router is connected to the IP network by a 45 Mb/s, Digital Signal 3 (DS3) coaxial cable. The server contains the information (YouTube videos, files, VoIP calls) to be sent to the clients in Vancouver and a local client is setup for troubleshooting and traffic validations.
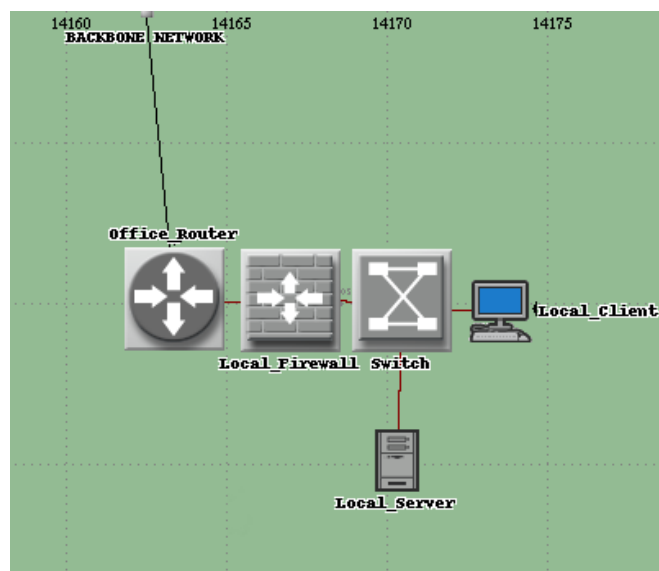

Figure 3: Server subnet topology.

Figure 4 below shows the attributes from Local_Server in figure 3 above. The important parameter to note in figure 4 is "**Application: Supported Services**". Our local server only handles one application and **P2P (peer-to-peer file sharing)** background application at a time; so, we assign one specific service defined in **Application Definition (YouTube,VoIP,FTP)** at the start of each simulation. Only **FTP** is shown in figure 5 as an example.

Figure 4: Local server application definition.

Figure 5: Applications Supported Services by the local server

2.2.2 Client Layout

For the client layout shown below in figure 6, we have set up a coffee shop scenario with a router, a mobile device and five (5) stationary laptops. The router is connected to the server via a backbone IP network that is connected to the server subnet in San Jose. The table below shows the characteristics and locations of all devices for both 802.11b and 802.11n configurations.

Table 1: Detailed device information from figure 6.

Device name	x-coordinate (m)	y-coordinate (m)	distance from router (m)	Supported Applications
Mobile_1_1	0	7.5	7.5	VoIP/YouTube/FTP

Fixed_0	2.5	0	2.5	VoIP/YouTube/FTP
Fixed_1	-15	0	15	VoIP/YouTube/FTP
Fixed_2	0	-10	10	P2P
Fixed_3	10	0	10	P2P
Fixed_4	7.07	-7.07	10	P2P

The radius of the hexagonal cell shown in figure 6 represents the indoor wireless range. The radius for 802.11b and 802.11n are 35 meters and 70 meters respectively. In our simulation scenarios, three (3) stationary devices are dedicated to P2P file sharing application to generate background traffic. A mobile device and two (2) stationary devices are dedicated to run one of VoIP, YouTube, or FTP applications we defined in section **2.2.4** to **2.2.6**.
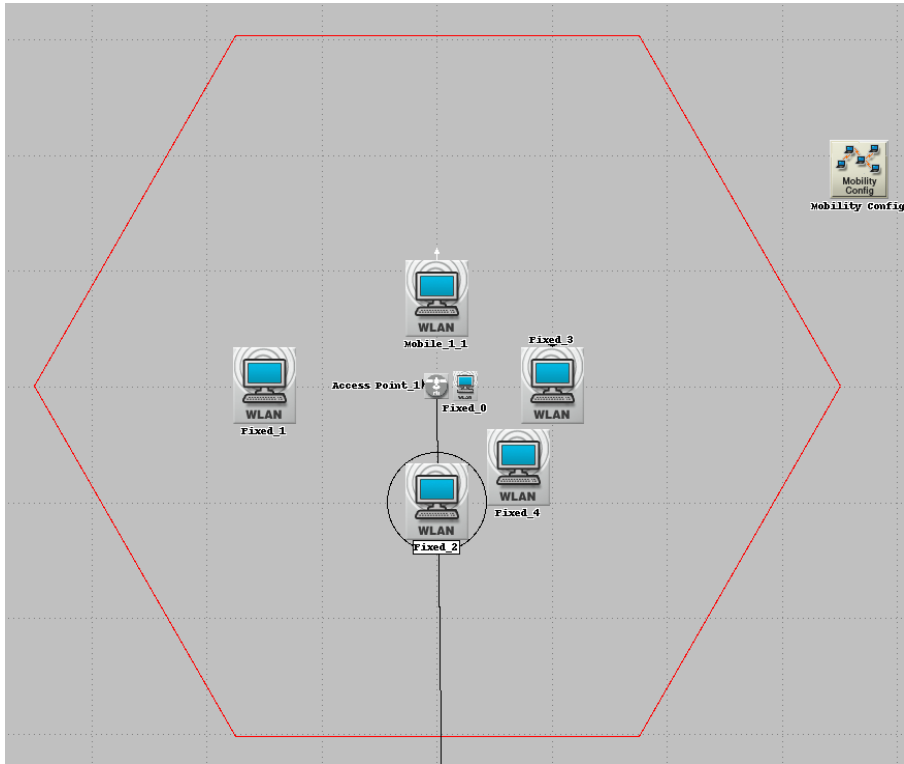

Figure 6: WiFi client subnet topology.

In all our simulation scenarios, the mobile device moves in a random trajectory within the cell at average human walking speed (1.4 m/s) as seen in figure 7 and 8, denoted as **Speed(meters/seconds) - constant (1.4)**. The major difference between figure 7 and 8 is **x_min, y_min, x_max, y_max** parameters. The minimum and maximum for x-axis/y-axis sets the

bounds in which the device will move (i.e. a person on the mobile device walking inside the coffee shop)

Figure 7: Mobile node 802.11b parameters

Figure 8: Mobile node 802.11n parameters.

Figure 9 shows an example workstation setup for stationary devices using VoIP application. Figure 10 shows an example workstation setup for the mobile device using VoIP application. In order to run an application on a device during simulation, we configured each workstation's **Application: Supported Profiles** attribute specific to each simulation scenario.

Figure 9: Example Fixed workstation setup.

Figure 10: Example Mobile workstation setup.

All the nodes (Router, workstations) inside the client subnet are configured to **Physical Characteristic: HT PHY 5.0 GHz** for 802.11n standard at **52 Mb/s** base data rate up to **480 Mb/s** data rate. For 802.11b standard the nodes are configured to **Physical Characteristic: Direct Sequence** and the data rate is **5.5 Mb/s**. This is seen in figure 11 & 12. One attribute that differentiates a router from a workstation is **Access Point Functionality**. For a router, this attribute is set to **Enable** and **Disable** for the workstations. **BSS Identifier** for all devices is set to **0** in order for all the routers and workstations to work together.
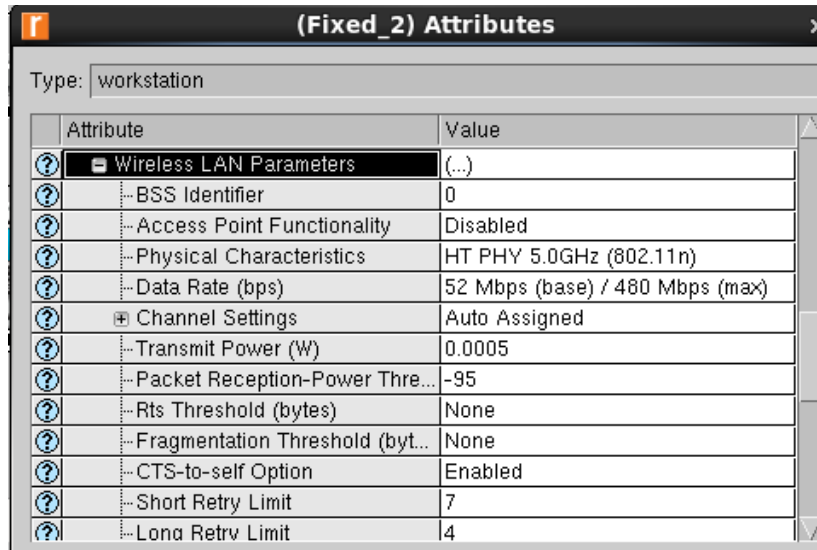
Figure 11: 802.11n workstation & router setup.
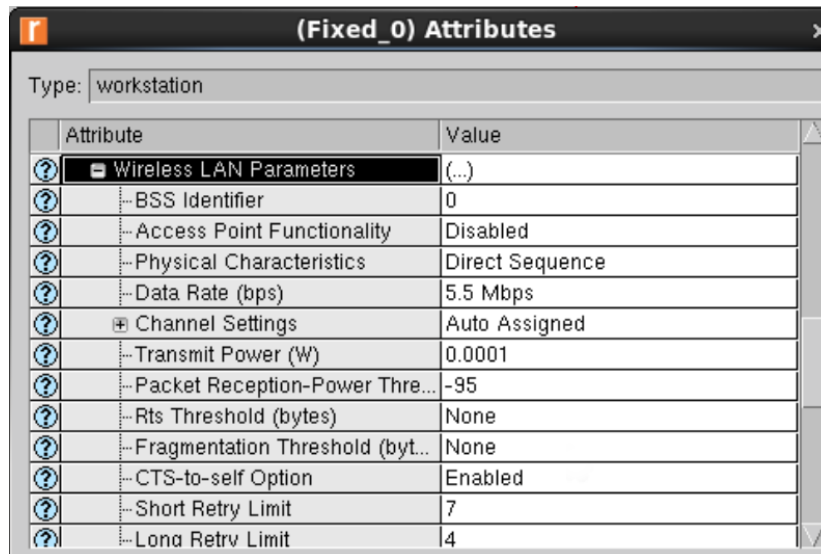
Figure 12: 802.11b workstation & router setup.

2.2.3 P2P Background Traffic Configuration

We selected riverbed's predefined P2P **Application Definition** called **High Traffic** in figure 13; analyze its average throughput between 802.11b & 802.11n shown, in figure 15.

Figure 13 : P2P application description.

High Traffic default file size is between 0.1 MB to 10 MB shown in figure 14. By looking at the size of the file, we know this is a scenario where users inside the coffee shop are downloading a song on an app.

Attribute	Value
Inter-Request Time (minutes)	exponential (30)
Requested File Size (bytes)	uniform_int (100000, 10000000)
File Popularity	uniform_int (1,5)
Leecher Probability	0.0
RSVP Parameters	None
Type of Service	Best Effort (0)

Figure 14: P2P table.
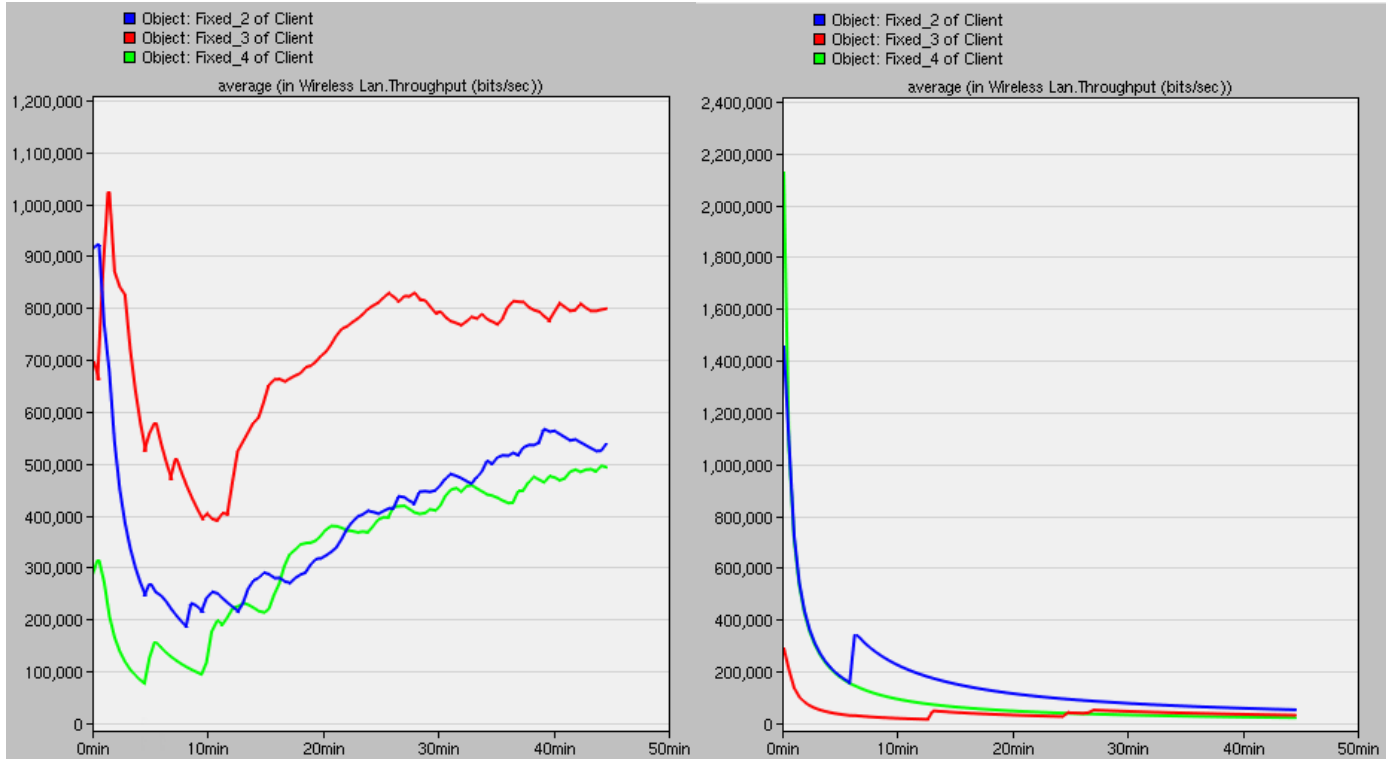
Figure 15: P2P average throughput (bit/sec), 802.11b on left, 802.11n on right.

2.2.4 YouTube Application & Traffic Configurations

YouTube uses TCP and HTTP protocols to deliver video streaming over IP stated from [4]. So we created **HTTP application (YouTube_1080P)** and modified the **application definition** to replicate the simulation behaviours from YouTube streaming in 1080P resolution shown in figures 16, 17 and 18 below.
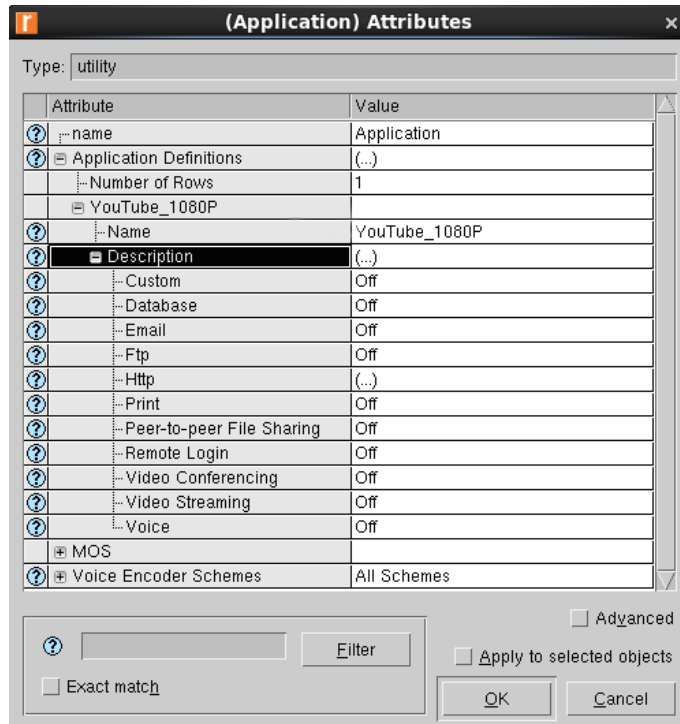
Figure 16: YouTube_1080P HTTP application description.

YouTube uses H.264, MPEG-2 or MPEG-4 for video encoding with frame rate in the range of 15-30 fps. In Figure 16, the **page interarrival time** is defined as the time it takes to refresh the page (Inverse of the frame rate), where the values are from **0.03333 seconds/frame** to **0.06667 seconds/frame**. This is shown in Figure 17 below.
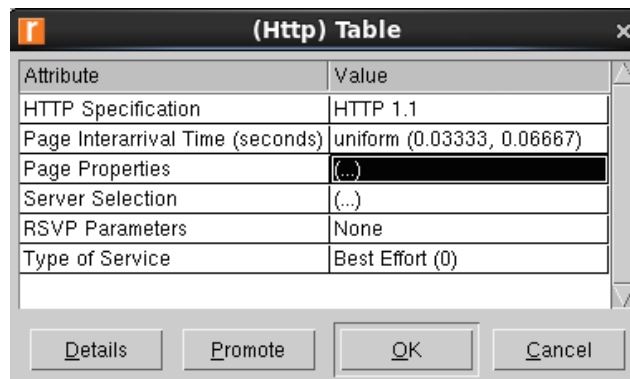
Figure 17: HTTP table.

In **automatically loaded page objects table** inside **page properties** in Figure 18, we defined the **object size** to **109227 bytes** that represents the file size in a single video frame. We chose H.264 video formatting for YouTube code and its corresponding bit rate is 25 Mb/s. The file size was determined by converting 25 Mb/s to 3.125 Mb/s then multiplying by 1/30 second/frames (YouTube 1080p frame rate) to get **109227 bytes/frame**. We also assumed the user is watching

the video in fullscreen so we selected **number of objects** to **single object**, as shown in Figure 18 below.

Figure 18: Automatically loaded page objects table.

2.2.5 VoIP Application & Traffic Configuration
We created **VoIP application definition**, which utilizes RTP, as shown in Figure 19 below. We used **G.711 (silence)** audio codec that uses Pulse Code Modulation (PCM) standard for Internet Protocol. **G.711** audio codec digitizes analog audio signal and outputs 64 kb/s digital signal, which can be seen below in Figure 19. **G.711** employs Packet Loss Concealment (PLC) technology that by minimizing packet loss increases performance.
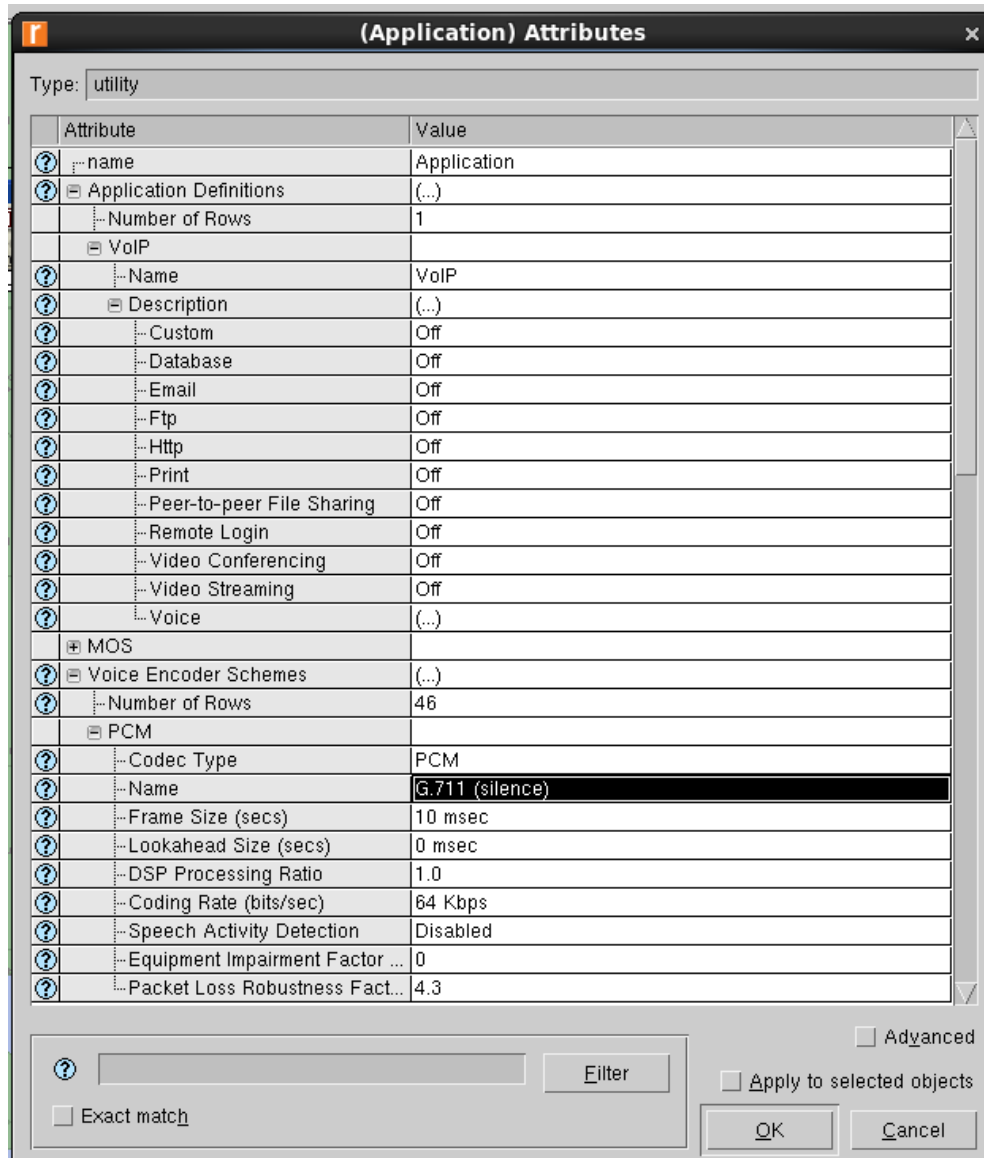
Figure 19: VoIP application definition.

The parameters in **G.711** and **G.711 (silence)** are very similar; however, **G.711 (silence)** codec accounts for the silence periods throughout the conversation and reduces the signal bandwidth accordingly. This allowed us to simulate a realistic phone call scenario where there are long period of silence in both ends of the call.

Figure 20: Voice table.

2.2.6 File Transfer Application & Traffic Configuration

We imagine a scenario where users in the coffee shop are downloading a .pdf file. We created the scenario in **Application Definition** called **FTP** in figure 21.

Figure 21: FTP Application Definition.

We defined a .pdf with file size of 50 kB as seen in figure 22.

Figure 22: FTP Table.

2.2.7 Profile Definition

To simulate two different applications in each scenario, we created two (2) profile configurations: **YouTube & P2P** as seen in figure 23. In all our scenarios, two (2) profile configurations are ran simultaneously according to table 1 above. All of our simulations were conducted in the same manner, to observe the effect of background traffic on the performance of user defined application definitions. The list below is the important changes we made to concurrently run two (2) applications in one (1) scenario.[5]

- Number of rows = two (2) profiles
- Start Time Offset (seconds) = No Offset - Application ran right as simulation begins
- Number of Repetitions = Unlimited - Sends application data until simulation is over
- Operation Mode = Simultaneous - This runs two (2) profile configurations at once!
- Start Time (seconds) = constant(0) - Each application starts its simulation time at t = 0
- Duration (seconds) = End of Simulation - The profiles are ran over and over until the end of simulation time as defined by user

---

[5] All Profile Configurations are identical. See Figure 22 below.

Attribute	Value
⑦ └label color	black
⑦ ⊟ Profile Configuration	(...)
⑦ └Number of Rows	2
⊟ YouTube_profile	
⑦ └Profile Name	YouTube_profile
⑦ ⊟ Applications	(...)
⑦ └Number of Rows	1
⊟ YouTube_1080P	
⑦ └Name	YouTube_1080P
⑦ └Start Time Offset (seconds)	No Offset
⑦ └Duration (seconds)	End of Profile
⑦ ⊟ Repeatability	(...)
⑦ └Inter-repetition Time (s...	exponential (300)
⑦ └Number of Repetitions	Unlimited
⑦ └Repetition Pattern	Concurrent
⑦ └Operation Mode	Simultaneous
⑦ └Start Time (seconds)	constant (0)
⑦ └Duration (seconds)	End of Simulation
⑦ ⊞ Repeatability	(...)
⊟ P2P	
⑦ └Profile Name	P2P
⑦ ⊟ Applications	(...)
⑦ └Number of Rows	1
⊟ P2P	
⑦ └Name	P2P
⑦ └Start Time Offset (seconds)	No Offset
⑦ └Duration (seconds)	End of Profile
⑦ ⊟ Repeatability	(...)
⑦ └Inter-repetition Time (s...	exponential (300)
⑦ └Number of Repetitions	Unlimited
⑦ └Repetition Pattern	Concurrent
⑦ └Operation Mode	Simultaneous
⑦ └Start Time (seconds)	constant (0)
⑦ └Duration (seconds)	End of Simulation
⑦ ⊞ Repeatability	(...)
⑦ └hostname	
⑦ └minimized icon	circle/#708090
⑦ └role	

Figure 23: Example profile definitions.

3. Performance

To analyze performance we took a look at two parameters:

1. Average throughput in bits/s: this shows the amount of incoming data achieved for each workstation during the simulation. Although this is important to see how much data can be consumed, it isn't a direct parameter of user experience.
2. Average delay in seconds: this shows the total average delay from the server, to the end workstation. This is a significant parameter for real-time applications such as video streaming and VoIP as high average delays will result in poor user experience.

3.1 802.11b WiFi
3.1.1 YouTube
The 802.11b YouTube simulation was the longest of the six (6) total simulations we did for each scenario at over one (1) hour. As a result, we only did a simulation lasting ten (10) minutes for both version of WiFi. The results for average throughput and average delay can be seen below in figure 24.

Figure 24: 802.11b YouTube average throughput (bits/sec) on left, average delay (s) on right.

The left hand of the figure, representing throughput, shows how insensitive 802.11b was to movement; in fact, the mobile node outperformed the stationary nodes reaching a peak throughput of 400 kb/s, and settling somewhere around 200 kb/s. The stationary nodes both peaked at about 175 kb/s and reached a steady-state of about 100 kb/s.

The average delay, on the right of the figure, really shows how incapable 802.11b is with a modern application like HD YouTube. The average delay steadily rises for the course of the simulation to the point where it reaches a full second — obviously unacceptable for a user to watch a video on demand.

3.1.2 VoIP
The VoIP simulation took about 20 minutes to complete a 30-minute call. The results of average throughput and average delay can be seen below in figure 25.
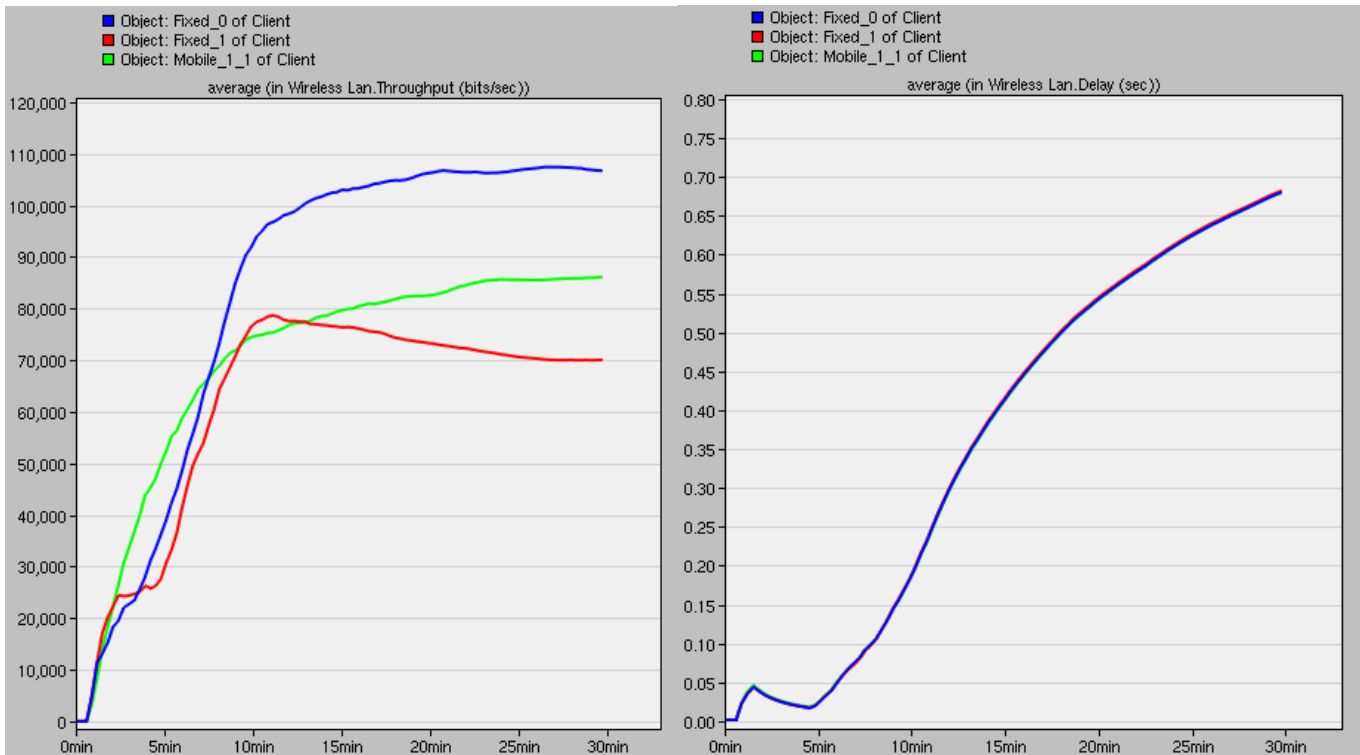
Figure 25: 802.11b VoIP average throughput (bits/sec) on left, average delay (s) on right.

From the figure on the left hand side it is evident that long term, distance had the most significant effect on the average throughput. The closer the node was to the router, regardless of mobility, the better the average throughput. The workstation closet, Fixed_0 (blue on the figure), achieved a steady state value of approximately 110 kb/s. The mobile node (green on the figure) was less at about 85 kb/s followed by the distant node that reached approximately 70 kb/s.

Again, the modern application crippled 802.11b as shown by the average delay on the right hand side of the figure. Like YouTube, the average delay increased throughout the simulation reaching a value of 0.7 seconds. At that rate, average delays in transmission during the phone call would indeed be evident and affect the customer's performance.

3.1.3 File Transfer

Lastly for 802.11b, File Transfer was the fastest of the three applications to simulate, completing in under 10 minutes to replicate a transfer lasting 25 minutes. The results for average throughput and average delay can be seen below in figure 26.

Figure 26: 802.11b File Transfer average throughput (bits/sec) on left, average delay (s) on right.

From the left hand side of the figure, you can see that the average throughput for File Transfer reacted inversely to distance (the exact opposite of VoIP). The fixed workstation furthest away achieved the highest average throughput, settling at a value of approximately 1.05 Mb/s. The mobile node reached approximately 700 kb/s whereas the closest fixed node was the slowest at about 550 kb/s.

As expected, the average delay was much more acceptable with the oldest application. Although the average delay raised the entire simulation, it appears to begin to settle at a value less than 0.1 s, which for File Transfer, is an acceptable rate.

3.2 802.11n WiFi
3.2.1 YouTube
Unlike the 802.11b simulation, the 802.11n simulation for YouTube took under five (5) minutes to complete for a 10 minute HD stream. The results for the average throughput and average delay can be seen below in figure 27.
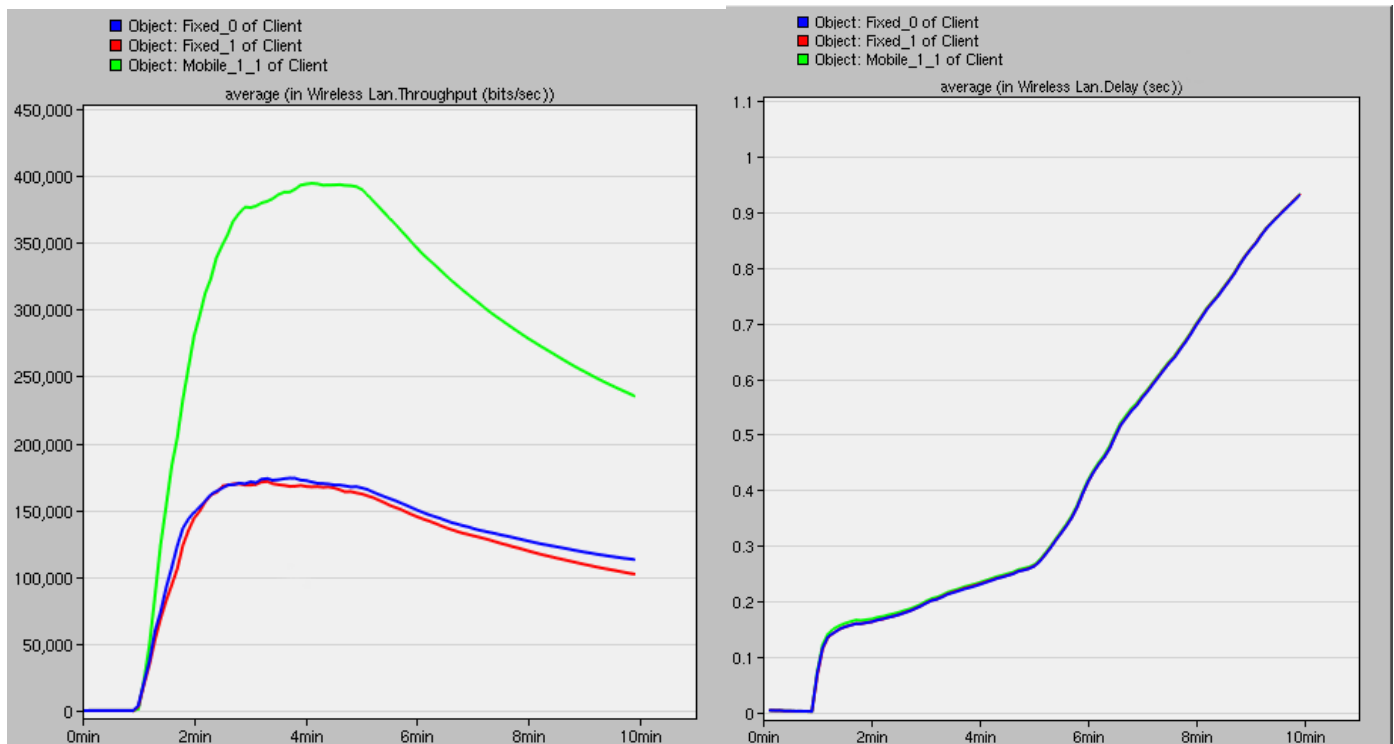
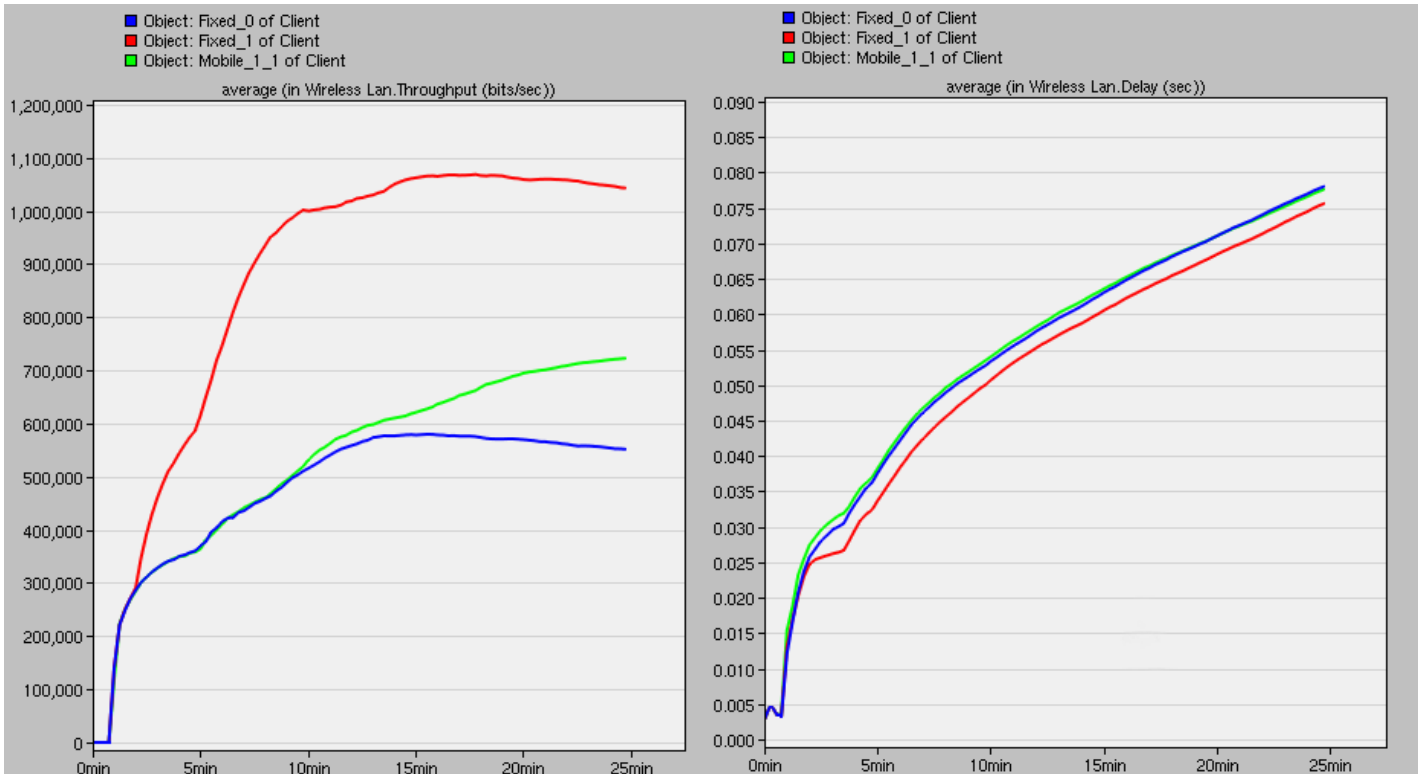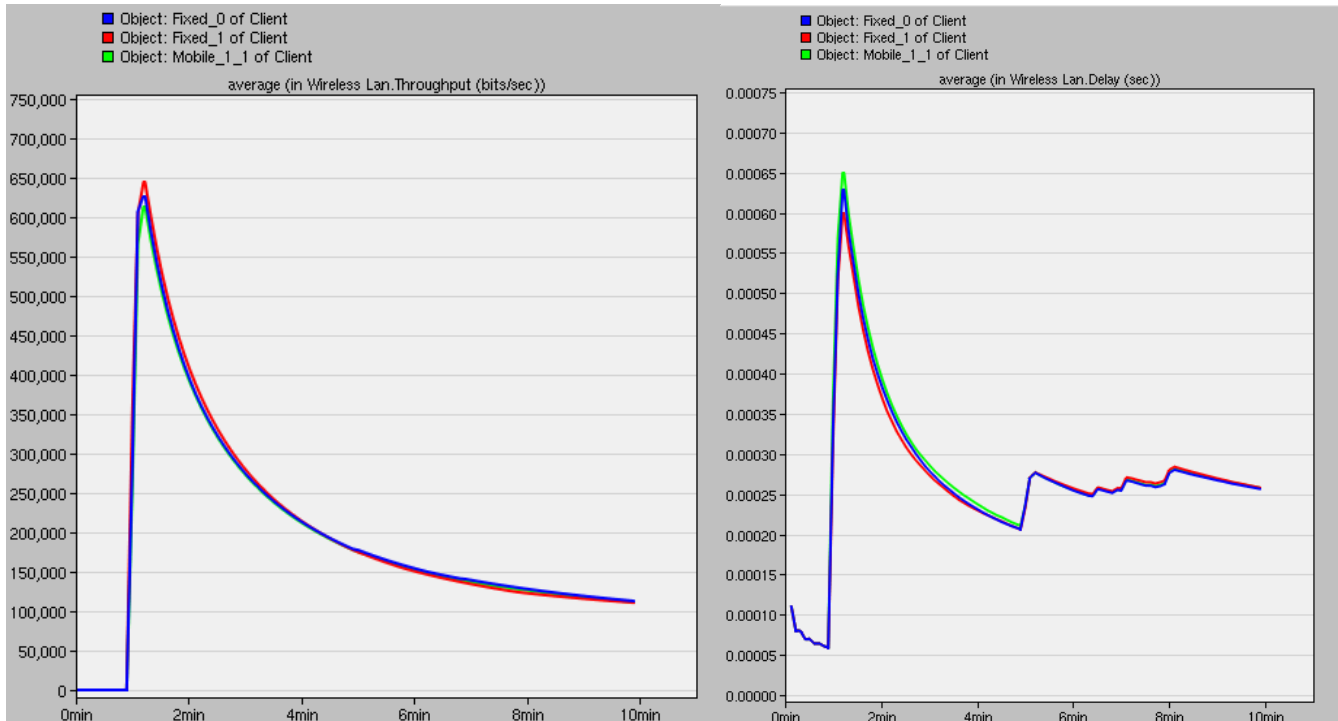

Figure 27: 802.11n YouTube average throughput (bits/sec) on left, average delay (s) on right.

From the left hand side of the figure, you can see that throughput was consistent regardless of distance or mobility. Average throughput reached a maximum value of about 650 kb/s and reached a steady-state value of approximately 100 kb/s.

The right hand side shows how fast 802.11n handles the HD YouTube stream. Average delay reached a maximum of 0.65ms and approached a steady-state value around 0.25ms. Performance like this would deliver fantastic user experience with little to no lag.

3.2.2 VoIP
VoIP simulation time only took under 10 minutes to complete, however, it failed to reach the entire 30-minute call simulation as the machine ran out of memory. As a result, figure 28 below only shows up to 23 minutes for the average throughput and delay.

Figure 28: 802.11n VoIP average throughput (bits/sec) on left, average delay (s) on right.

The results for the average throughput on the left hand side of the figure show sensitivity to both distance and movement for VoIP over 802.11n. The furthest workstation (red in the figure) achieved the highest throughput at about 150 kb/s and still rising at the end of the simulation. The mobile and other fixed nodes both achieved average throughputs of approximately 80 kb/s and rising.

Average delay, on the right hand side of the figure, was incredibly low for all three nodes. Although there were slight differences between each node, they were in the hundredths of milliseconds. The steady-state and maximum average delays were about the same, ranging from 0.14 to 0.18ms. For a real-time phone call, this would give no noticeable difference to the end user.

3.2.3 File Transfer
Surprisingly, File Transfer took up to twice as long to simulate for 802.11n compared to 802.11b (10 vs. 20 minutes). The results for average throughput and average delay can be seen below in figure 29.

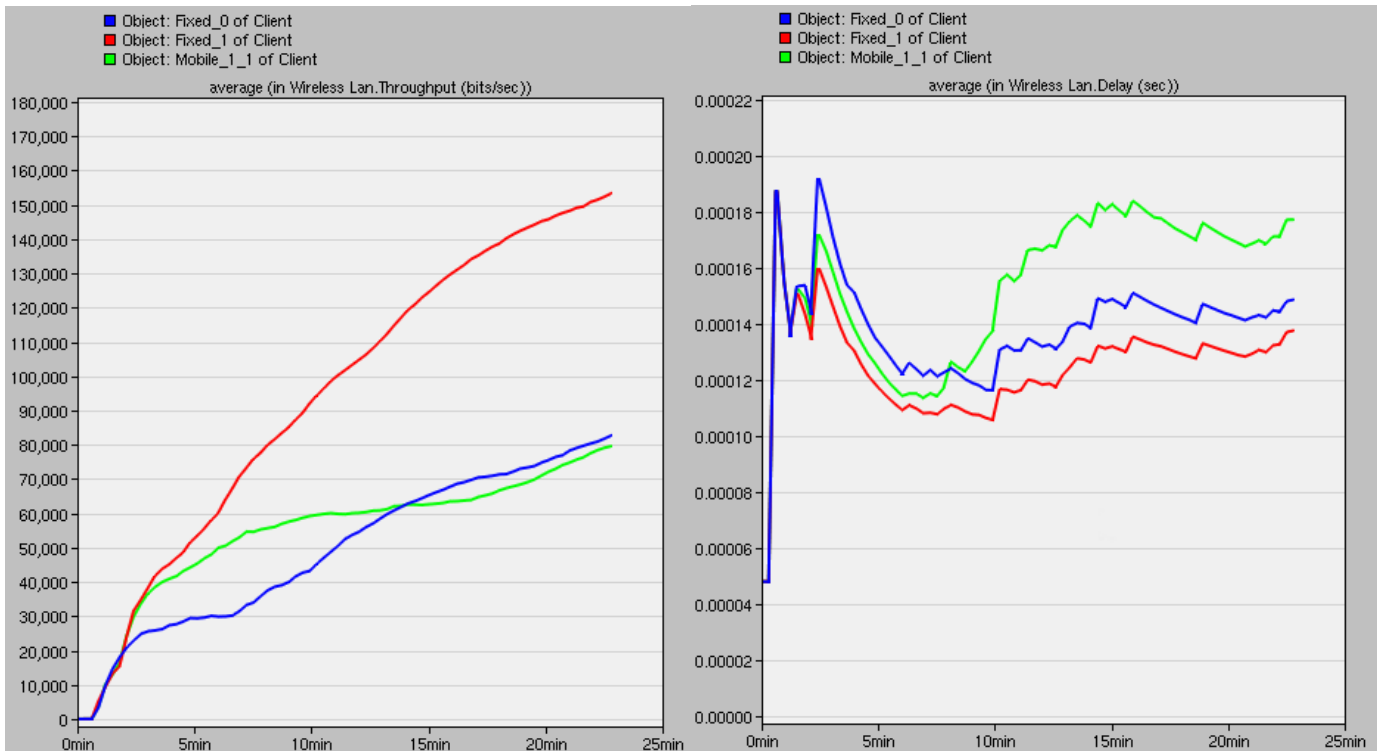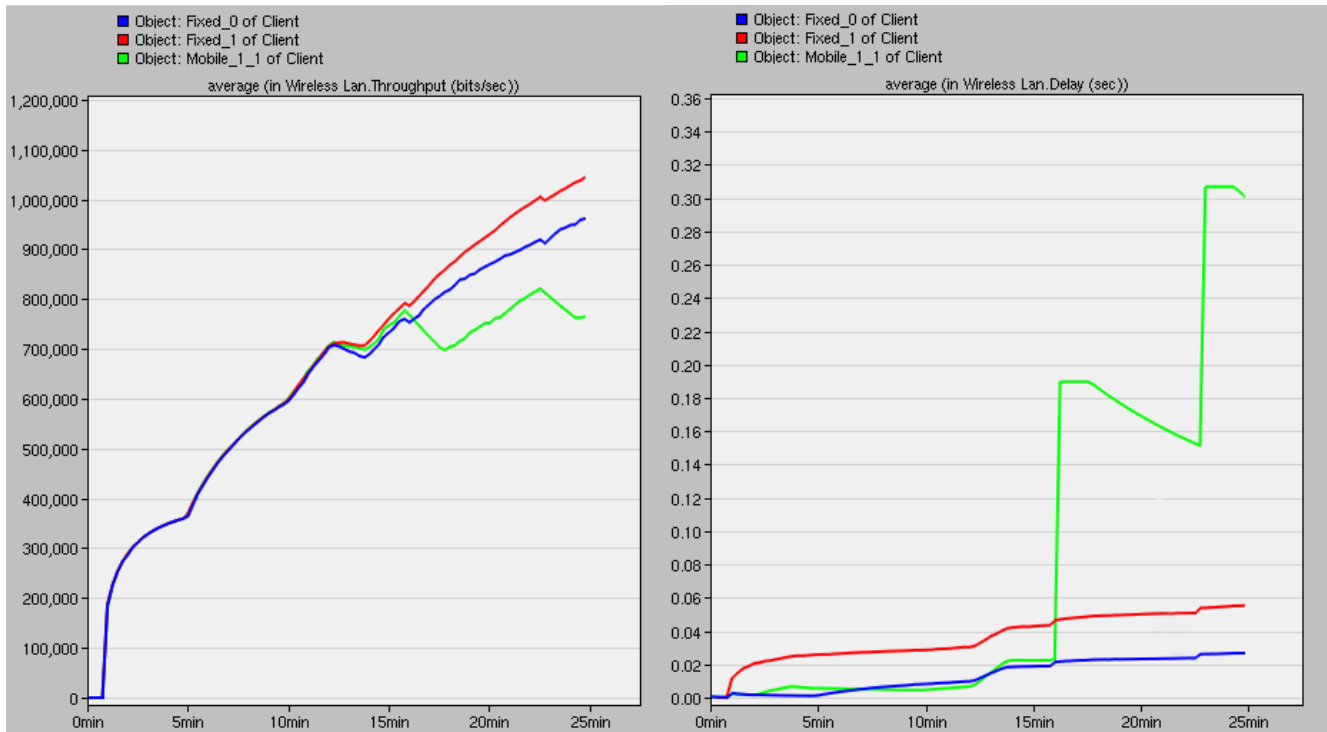Figure 29: 802.11n File Transfer average throughput (bits/sec) on left, average delay (s) on right.

The average throughput, on the left of the figure, shows how sensitive 802.11n is to mobility during the FTP. The furthest node (red in the figure) achieved the highest average throughput at about 1.05 Mb/s and was still rising at the end of the simulation. The closest fixed node was also rising at the end of the simulation, but only reached 950 kb/s. Lastly, the mobile node only reached as high as 800 kb/s, and seems to begin to settle around this value.

The average delay furthers the claim of 802.11n's sensitivity to movement during the FTP. The mobile workstation reached up to 300ms of average delay, whereas the fixed workstations settled around 60ms and 30ms for the far and close fixed nodes respectively. This difference isn't as significant for File Transfer, and the end user would likely not notice much improvement between each of the workstations.

4. Comparison & Discussion
4.1 YouTube

Figure 30 below shows the comparison of average throughput for YouTube for 802.11b (left) and 802.11n (right). Both technologies have their fixed nodes reaching a steady-state value of 100 kb/s, but the mobile node on the 802.11b simulation shows four (4) times this performance. This is an interesting effect of 802.11b, as mobility seems to have more of an effect on throughput.

Although the steady-state values are the same, it is clear that the total throughput (area under the curve) is greater for 802.11n as expected. Lastly, the maximum throughput of the mobile and fixed nodes for 802.11n was upwards of 160% and 370% larger respectively than 802.11b.
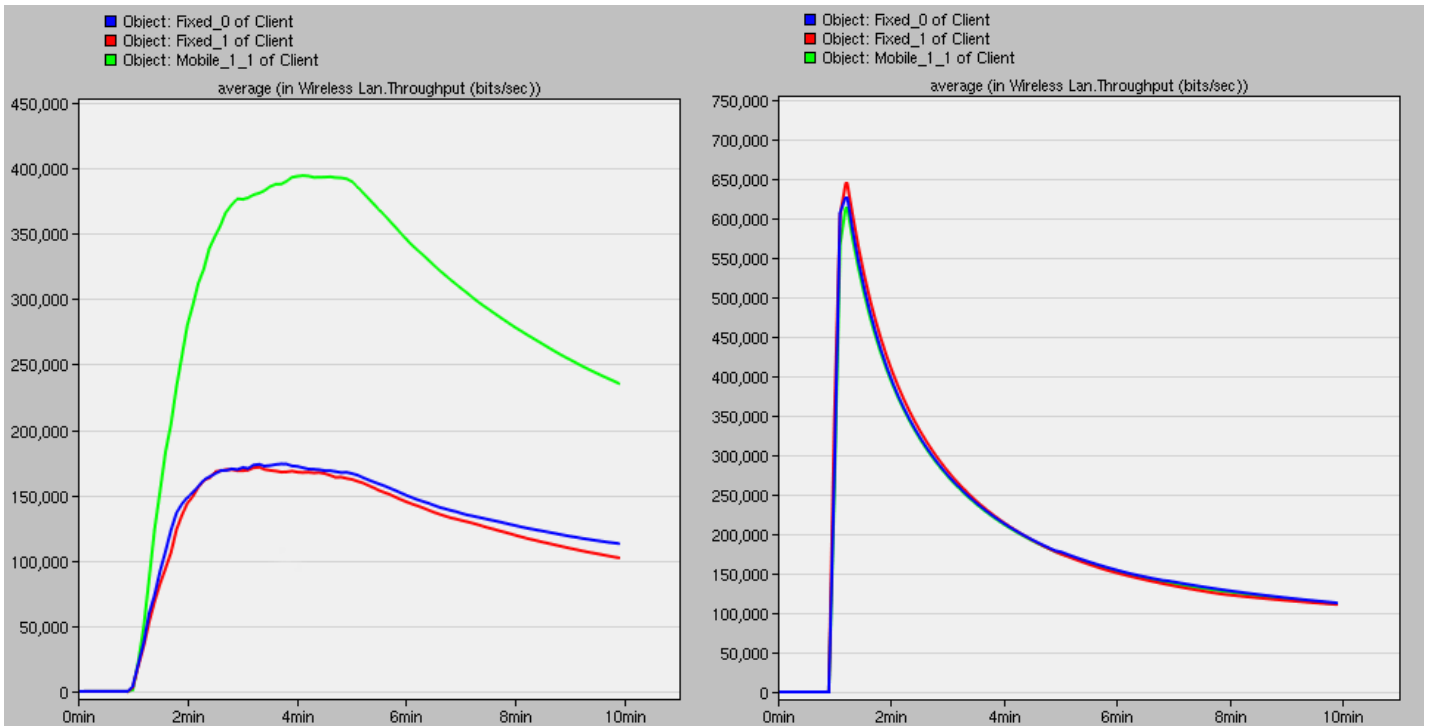
Figure 30: YouTube average throughput (bits/sec), 802.11b on left, 802.11n on right.

Average delay on the other hand, as shown below in figure 31, shows a much more significant difference in the two technologies. YouTube as shown to the left hand side of the figure, cripples 802.11b, reaching almost one (1) second and rising at the end of the simulation. 802.11n performs as expected, keeping average delays to less than a millisecond throughout the course of the simulation.

This is where we expected the largest difference during this comparison. YouTube HD is the most data intensive application and we were pleased to have the results to demonstrate this.

Figure 31: YouTube average delay (s), 802.11b on left, 802.11n on right.

4.2 VoIP

Figure 32 below shows the comparison of average throughput for VoIP for 802.11b (left) and 802.11n (right). Where 802.11b seems to reach steady-state values ranging from 70 to 105 kb/s, 802.11n continued to increase past the length of the simulation after reach values from 80 to 150 kb/s. Besides the higher throughput for 802.11n, another notable is that the factor of distance and mobility of each workstation played different roles in each WiFi.

For 802.11b, close proximity seemed to of helped overall throughput. 802.11n was the opposite, the furthest node saw the best performance as both mobility and small distance to the router decreased throughput almost 50%.
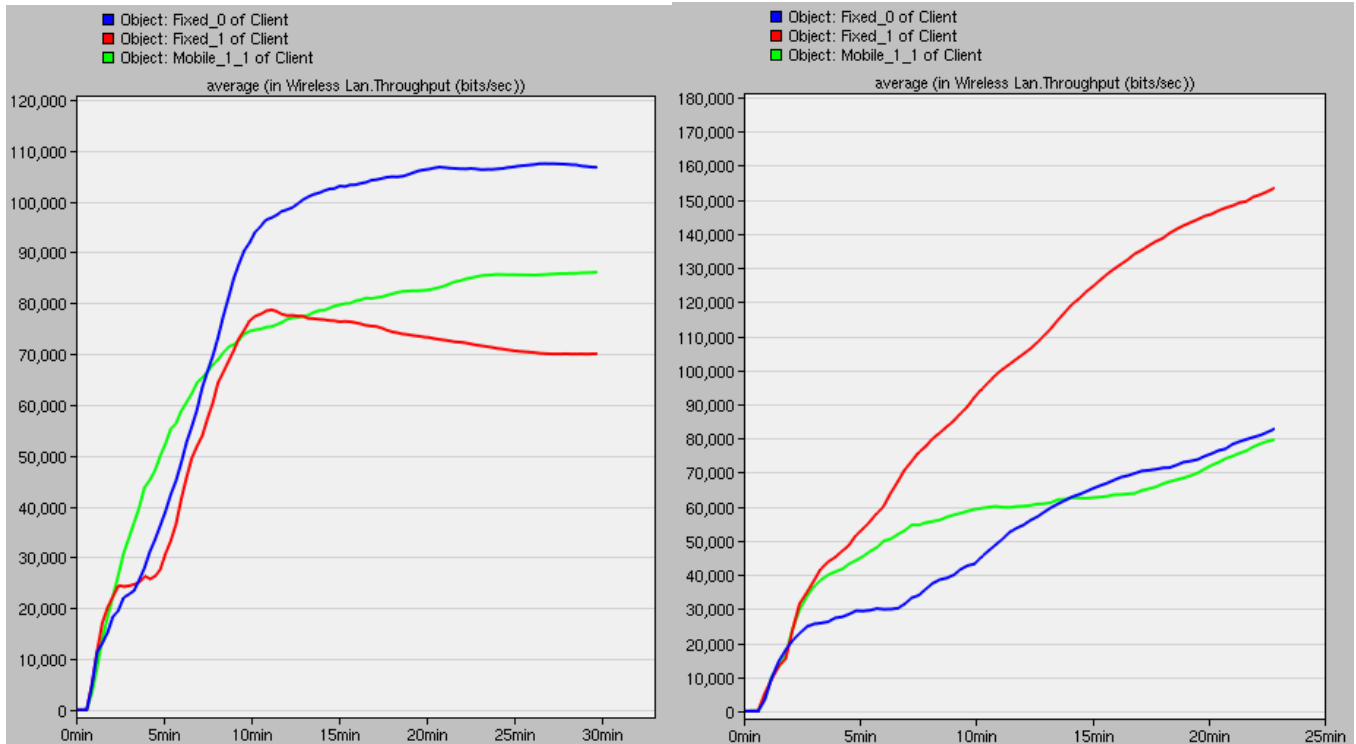
Figure 32 : VoIP average throughput (bits/sec), 802.11b on left, 802.11n on right.

Looking at the average delay of 802.11b in the left hand side of figure 33 below, you'll notice that once again, like YouTube, the average delay continues to increase well past the end of the simulation. Times reach almost 0.7s at the end of the 30-minute simulation, decreasing user experience with delay during real-time transmission of voice.

802.11n on the right side of the figure, kept average delays well under 0.2ms the length of the simulation, where the three (3) nodes experiences differences seen only in the hundredths of milliseconds.

Once again, the newer protocol proved to be challenging for the dated 802.11b protocol as shown by the lagging average throughput and increasing average delay graphs. We did expect this to be the case, however, we did not predict that the performance comparison for VoIP would be as similar as HD YouTube as the required bit rate differs by over 4500% (3000+ kb/s vs. 64 kb/s)
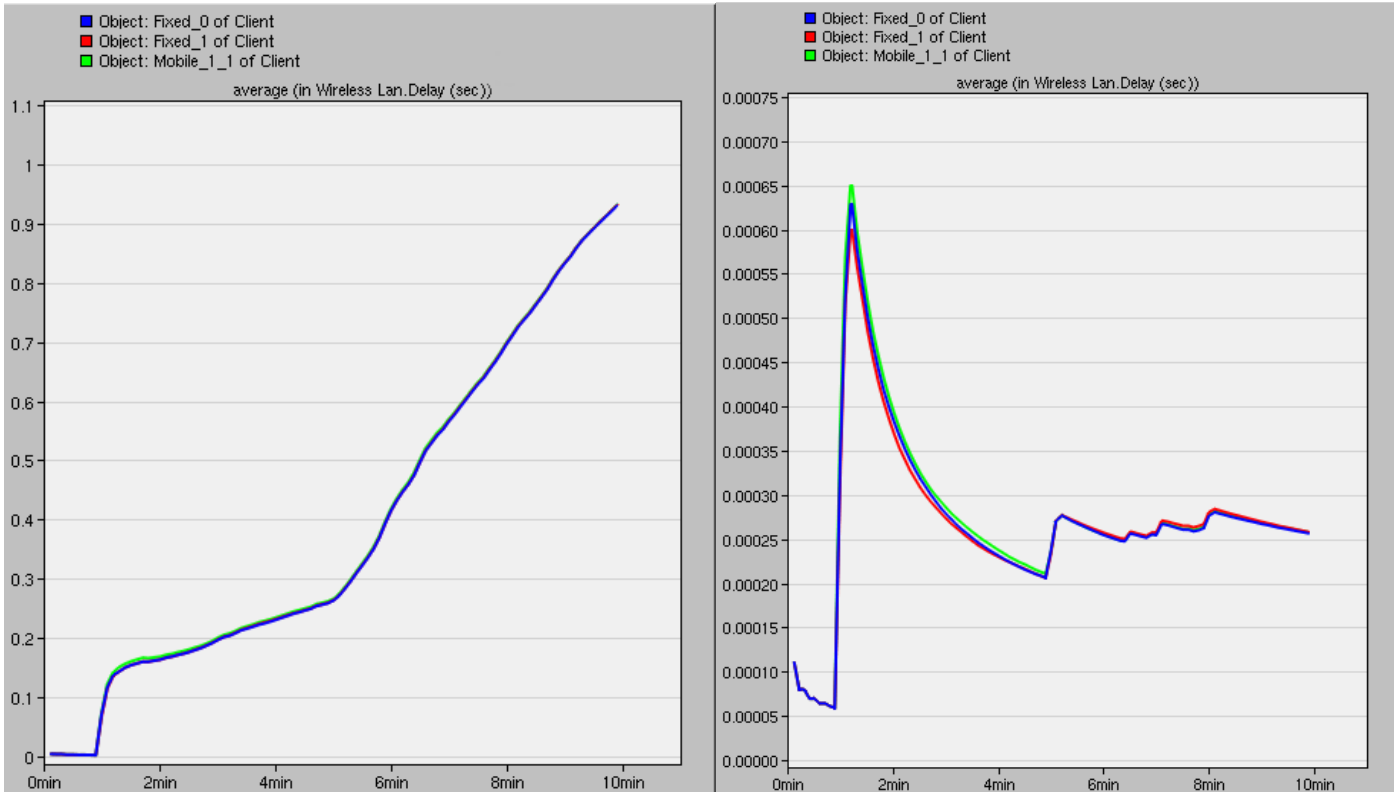
Figure 33: VoIP average delay (s), 802.11b on left, 802.11n on right.

4.3 File Transfer

Figure 34 below shows the comparison of average throughput for 802.11b (left) and 802.11n (right). File Transfer was the application most interesting, as it showed how a new technology handled an older protocol, FTP, compared to a technology designed in its time. 802.11b had a respectively large range for each of the workstations; the fixed node furthest from the router had the largest average throughput, reaching a steady state value just over 1 Mb/s. The mobile node achieved a steady-state average throughput of 700 kb/s while the remaining fixed node was just shy of 600 kb/s.

The maximum average throughput for 802.11n was essentially consistent for the three nodes until the last quarter of the simulation. Again, the highest average throughput was for the furthest node, reaching a value of over 1 Mb/s and still rising. Next was the nearby fixed node at over 950 kb/s and rising, and lastly was the mobile node which appeared to settle around 800 kb/s.

Although the maximum average throughput was similar at simulation's end for both technologies, 802.11b reached a steady-state whereas 802.11n was still on the rise. We see here that even though an older protocol is used, the advancements of 802.11n are still applicable.
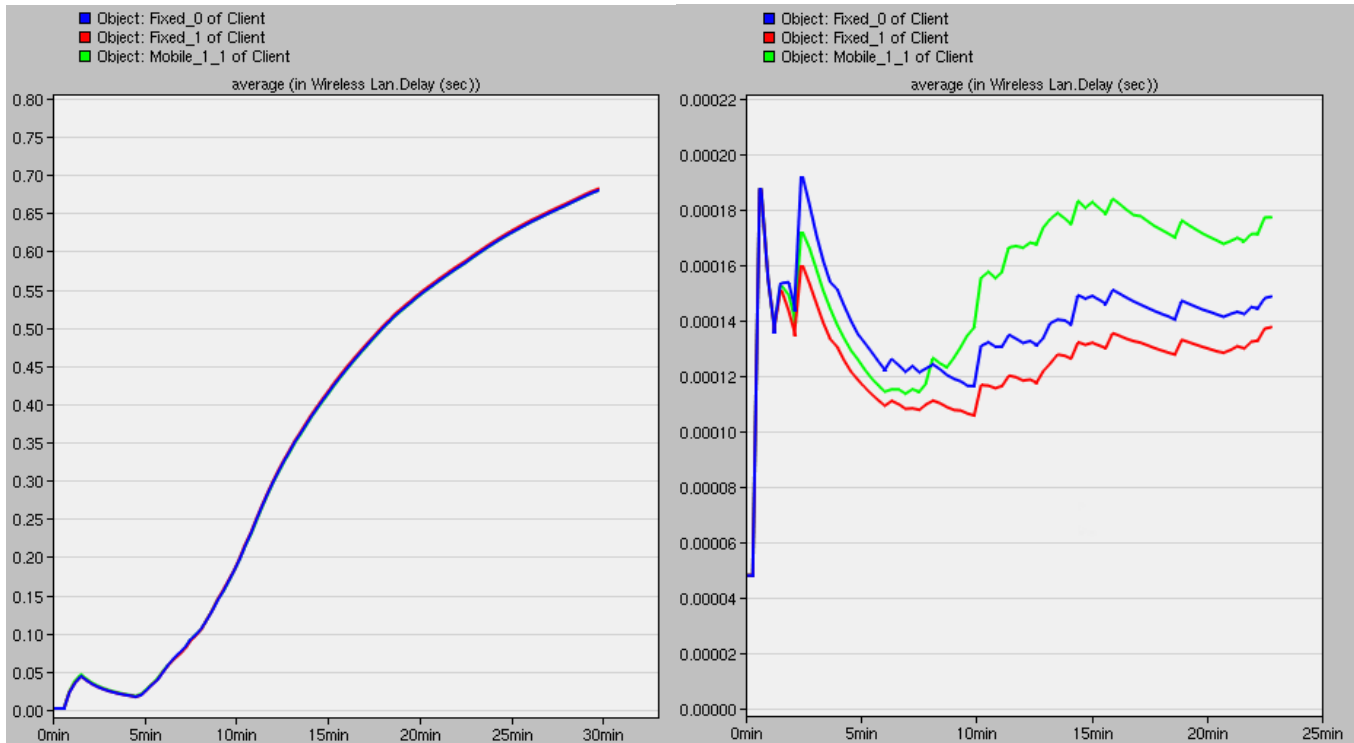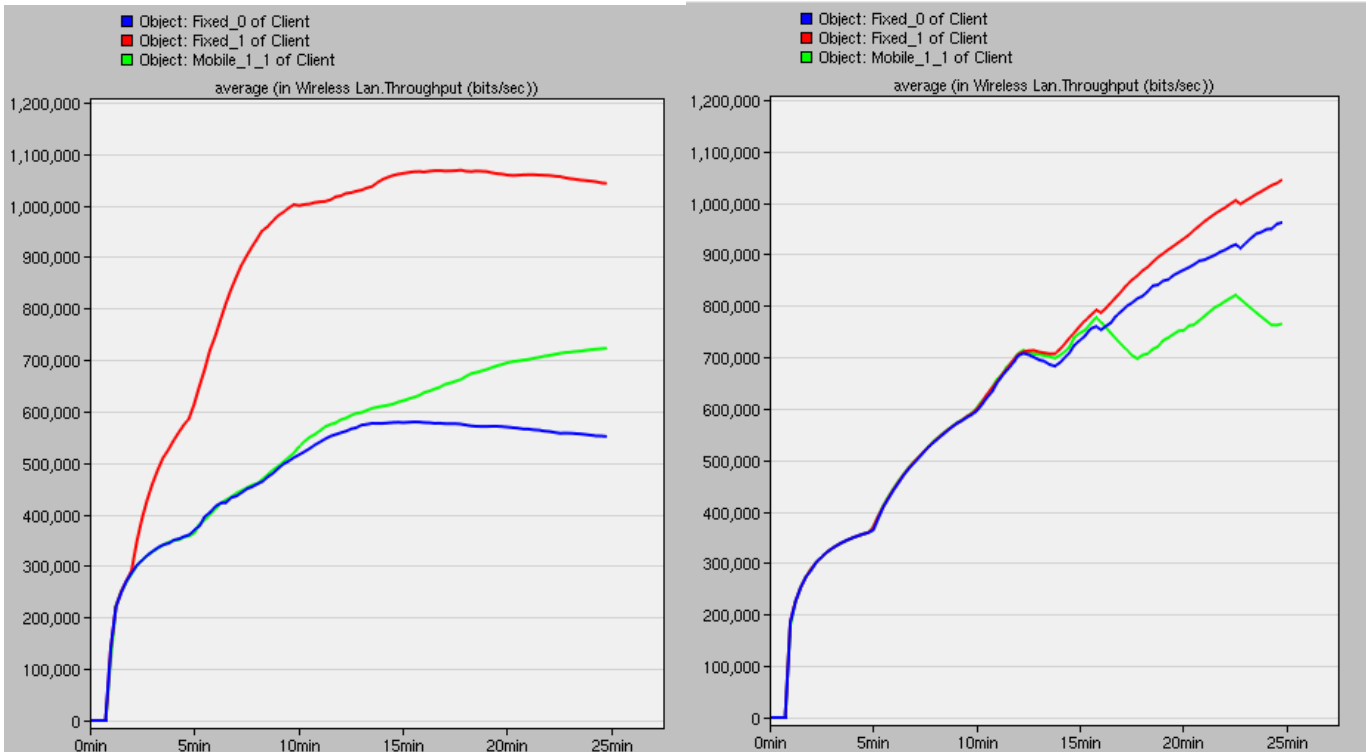
Figure 34: File Transfer average throughput (bits/sec), 802.11b on left, 802.11n on right.

The average delay for 802.11b (left) and 802.11n (right) can be seen below in figure 35. 802.11b's average delay approaches 80ms at the end of the simulation and is still rising. Neither distance nor mobility made an effect as all three nodes demonstrated similar performance.

802.11n on the other hand, saw its fixed workstations reach steady-state values of approximately 60ms and 30ms for the far and nearby nodes respectively. Unlike 802.11b though, the mobile node saw a dramatic increase in average delay. At the end of the simulation, it reached a value of 300ms, which is 375% greater than the mobile node for 802.11b.

Figure 35: File Transfer average delay (s), 802.11b on left, 802.11n on right.

5. Conclusions

As we expected, 802.11n outperformed 802.11b in both average throughput and average delay from the server. However, the results weren't as obvious as we had expected. Originally, we assume that the throughput for 802.11n would be significantly greater in comparison, but as seen below in table 2, that wasn't the case. Table 2 was c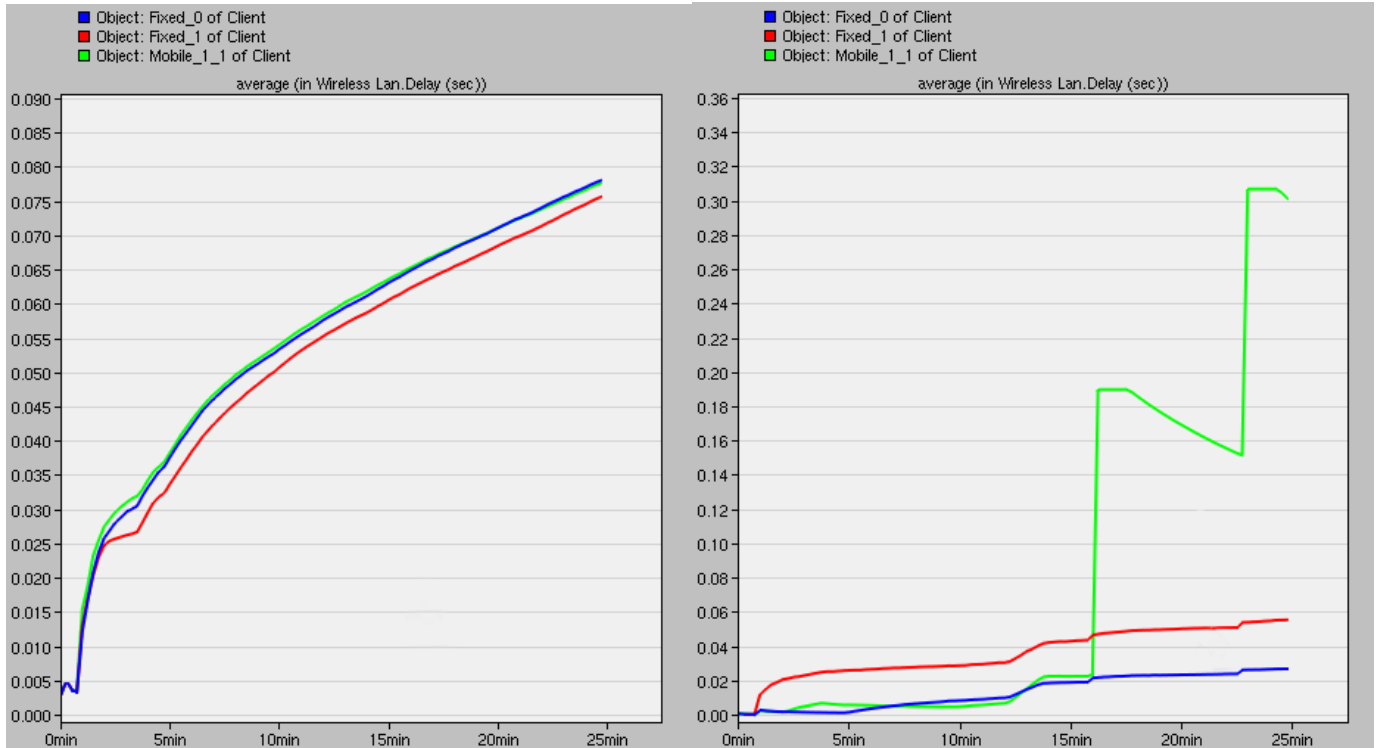onstructed by averaging the three workstation's performance for both maximum average throughput and average delay. The "+" designator, shows that the value was still increasing at the end of the simulation.

The largest difference in maximum and steady-state throughput was 250% and 120% respectively. This, compared to the bit rates used in our simulation (5.5 Mb/s for 802.11b and 480 Mb/s for 802.11n) was not a direct correlation.

Average delay on the other hand, was much more representative of the technology differences and end user experience. Unlike throughput, the differences in table 2 were huge in comparison. Maximum average delay differed by up to 1500000%, while steady-state average delay was even higher at 3500000%. This poor average delay for 802.11b would obviously affect a user's video stream or voice call making the technology handicapped in today's large bandwidth applications.

Furthermore, larger differences were experienced as the age of the application reduced. YouTube HD saw the largest difference between 802.11n and 802.11b, and the margin was reduced when

moving to VoIP and finally File Transfer. We hoped to prove this result, as it was intuitive that the two WiFis would handle applications in their era more appropriately.

Table 2: Summary of approximate results averaged of all three nodes.

	Average Throughput (kb/s) (max/steady-state)		Average Delay (ms) (max/steady-state)	
	802.11b	802.11n	802.11b	802.11n
YouTube	250/160	625/110	950+/950+	0.6/0.25
VoIP	90/90	100+/100+	680+/680+	0.16/0.16
File Transfer	750/750	900+/900+	85/60	3.5/1.5

By using three (3) different nodes, we also hoped to show a correlation between distance and movement. There was no immediate clear sign of superiority between distance and movement, but we did manage to find some general trends.

Delay appeared to have little dependence on mobility or distance from the router as five of six results showed the same value for each workstation. The one exception surprisingly, was 802.11n during the File Transfer application. The delay shot up 500% compared to the next nearest value, which must be an indication of the newer technology's legacy mode running the old application.

Average throughput on the other hand, was all over the map. Most of the time, the *furthest* fixed node had the best performance (mostly for 802.11n). The other two workstations also had simulations where their average throughput was the highest as well. As a result, we had little conclusions to draw between the three (3) nodes and their effect on average throughput; it seems as if being 5m from an 802.11n router has reduced throughput than if you're working 15m away.

5.1 Future Work

There are three (3) primary areas where we would like to expand our research.

1. As Riverbed Modeler only supported up to 802.11n, we weren't able to use the latest version of WiFi. As this becomes available, we'd like to analyze the average throughput and average delay of 802.11ac, which was introduced in 2013. With this update, we would cover up to fourteen (14) years of improvement in 802.11 LAN standards, which represents a very significant segment of technological improvement in the industry.

2. As this project focused on the advancement of LAN standards, we are also interested in the same period of time for Wide Area Networks (WANs). In particular, EDGE networks compared to modern Long-Term Evolution (LTE) WANs. A comparison would be done to see if the improvements lead, lag, or maintain the same progress in LANs.

3. Lastly, we would like to explore our original project proposal: application performance using 802.11ac WiFi LAN compared to a LTE WAN. Two things prevented us from exploring this originally. First, no licenses were available for LTE and secondly, 802.11ac was not available in Riverbed Modeler 18

6. References

[1] G. R. Hiertz, D. Denteneer et al., "The IEEE 802.11 Universe," Communications Magazine, IEEE, Volume: 48, Issue: 1, January 2010. Accessed: February 15, 2015. Available: http://ieeexplore.ieee.org.proxy.lib.sfu.ca/xpl/articleDetails.jsp?arnumber=5394032&tag=1

[2] "Wireless Networking," Vicomsoft. Accessed: March 22, 2015. Available: http://www.vicomsoft.com/learning-center/wireless-networking/#3

[3] L. Phifer, "Differences between WLANs, Wi-Fi and WiMax". Accessed: February 15, 2015. Available: http://searchnetworking.techtarget.com/answer/Differences-between-WLANs-Wi-Fi-and-WiMax

[4] C. Chen, S. Sheng, and J. Yoo, "High Resolution Video Streaming over Wi-Fi, WiMAX and LTE," Accessed: March 22, 2015. Available: http://www.sfu.ca/~cyc19/report.pdf

[5] R. Gill, T. Farah, and L. Trajkovic, "Comparison of WiMAX and ADSL Performance when Streaming Audio and Video Content," Accessed: March 22, 2015. Available: http://www2.ensc.sfu.ca/~ljilja/papers/Opnetwork2011_farah_gill_final.pdf

[6] Riverbed Modeler's Guide to Applications. Accessed: March 22, 2015. Available: https://www.sfu.ca/~jaridw/Reference6.pdf

[7] MCR Connect, "Where Voice, Data, and Communication Converge," Accessed: April 12, 2015. Available: http://mcrconnect.com/where-voice-data-and-communication-converge/