WORCESTER POLYTECHNIC INSTITUTE

FINDING OPTIMAL YOUTUBE BUFFER SIZE FOR MOBILE DEVICES ON A CAMPUS NETWORK

David Eufemia, Xia Li, Will Parker, Hnin Pwint Phyu

3/17/2014

Advisor: Mark Claypool

TABLE OF CONTENTS

Wireless video streaming is becoming more prevalent in this day and age. When streaming video, network signal fluctuations can cause interrupts in the video playback. These fluctuations are even more likely to occur when the network is wireless and the device is mobile. To counteract this, web video players use buffers to store the video data ahead of time. However, if the buffer is too small or the network is too unstable, interruptions may still occur. One popular streaming service that implements client buffers is YouTube. In this paper, YouTube video playback when streaming to a mobile wireless device on the WPI campus is analyzed. The experiment starts out by streaming video while walking along predetermined paths. Throughput traces are captured using a network sniffer and run through a video playback simulator with varying buffer sizes. A cost function is developed by weighing different variables that attribute to the quality of the viewing experience such as initial buffer size, interrupted time, interrupt frequency and duration of playback before the first interrupt. Using simulator results, the initial buffer size that results in the lowest cost is determined.

1 INTRODUCTION

With the turn of the century, video streaming has become a popular method to deliver entertainment and informative media across various demographics. On a global scale, there are about 50 billion online video views per month [1]. Online video now accounts for 50 percent of all mobile traffic and up to 69 percent of traffic on certain networks [2]. Cisco has forecasted that online video will amount to 55% of all consumer Internet traffic by 2016 [3].

The term video streaming comes from the fact that the traveling information is a stream of data from a server. The decoder is a stand-alone player or a plugin that works as part of a Web browser. The server, information stream and decoder work together to deliver a requested video to the viewer over the Internet. The video file is compressed using an encoding algorithm in order to reduce file size when being sent over the internet. When the file gets to its destination, the player decodes and displays the frame on the screen, and usually retrieves information a little faster than it is being played. This extra information stays in a buffer in case the stream falls behind, which is the case when the available bandwidth (the rate of data to conduct a file transfer from the server to a computer) drops lower than the bitrate or the playback rate of the video.

With the popularity of online streaming, many video hosting websites emerged in the 2000s. Some of the sites are for commercial purposes where the user can pay to watch professionally produced clips such as movies, TV shows or even educational videos. Such sites are known as subscription video on demand services. On the other hand, there are sites that offer free user generated video sharing. Among these different video hosting sites, YouTube is the best-known, free video-hosting site on the web. As of 2013, over 6 billion hours of video are watched per month on YouTube [4], and it has grown in web traffic referral by over 50% [5]. YouTube is now available on 400 million devices. Video streaming started out with desktop computers but with the immense popularity of mobile computing and the rise of tablets and smart phones, it has now been integrated into mobile devices and is available over wireless networks. YouTube mobile, an application optimized for mobile devices, takes up 40% of total YouTube viewing time and has an average of one billion views per day [4].

Much of the mobile video streaming, through YouTube and other video hosting sites, takes place over wireless networks. The growth in popularity of wireless networks for data transfer came around the same time as smart phones. Data consumption of Wi-Fi devices is predicted to surpass that of wired IP traffic by 2015 [3]. However, one big drawback of Wi-Fi networks versus wired networks is the network variability. Unlike wired networks, Wi-Fi has no physical connections between sender and receiver and transmits data by radio wave propagations instead. With the air being the main transmission media for wireless data, the transmitted signal is exposed to many obstacles and distortion during transmission. Factors that contribute to the Wi-Fi signal's deterioration include the network infrastructure, geographical conditions, network traffic, and even weather. When the mobility component is taken into account, there are even more aspects that affect the wireless network such as movement speed, human traffic, locations of access points and access point handoffs. All of these can affect the wireless networking and thus, the playback of YouTube videos. Issues such as video start-up delays, interruptions, and losing connection with the YouTube server can occur.

The client buffer system is a counteractive measure that YouTube and other video streaming systems utilize to prevent these issues. When a user requests a video to be played, a buffer of a predetermined size is filled with video data before playback begins. This is intended to make up for the variations in bandwidth and prevent potential playback interruptions. However, this buffering mechanism still does not guarantee interrupt-free video playback since the behavior of the wireless network can be unpredictable at times. One solution to this problem would be to have a buffer size large enough that no interrupts will ever occur during playback, but this may result in the user having to wait a long time before the video has started playing.

Our study approaches video buffering with the goal of finding the smallest optimal buffer size for playback with few or no interrupts, while also keeping the initial delay time to a minimum and deliver quality a viewing experience for the user. Our study gathers data on the Worcester Polytechnic Institute (WPI) campus network. We conduct pilot tests to determine among the external factors affecting the video playback (such as path, network traffic, video quality, etc.) which parameters to vary and which to keep fixed. We gather wireless network data from YouTube while walking along set paths and playing a selected video on YouTube using Wireshark, a network analysis tool that records information about packets being sent over a network. We also develop a video playback simulator that allows us to simulate playback of the video with alternate initial buffer sizes. The simulator takes the bandwidth trace and the bitrate of the video as inputs, as well as the initial buffer size. Using these arguments, it simulates playback to determine how many interrupts occur with the given buffer size. By varying the buffer size, we determine how large a buffer is needed to avoid any interrupts for any of the paths as well as explore quality tradeoffs between buffer sizes.

The gathered data indicates that lower quality videos (360p and 480p) are able to stream smoothly without any interrupts while one is moving across a wireless network, and are also able to buffer far ahead of the video playback. However, as the desired video quality gets higher, the gap between the buffered video and the played video gets smaller, and often, the playback catches up to the buffer and causes interrupts. It is also observed that for 1080p video, a linear relationship exists between the average interrupt length and the number of interrupts per one playback session. Using these assessments, we make a cost metric algorithm that combines the initial wait time/delay, total interrupted time, number of interrupts and initial interrupt-free period, and find the buffer size that provides the best perceived video quality for the user.

The rest of this paper proceeds as follows. Work relating to YouTube, wireless networks, and video streaming in general is discussed in Section 2. In Section 3 we explain our methodology for our experiments and the parameters we use. In Section 4 we present our results. In Section 5 we analyze the results and look for the optimal buffer size. In Section 6 we the conclusion to our study. In Section 7 we discuss possible future work.

2 RELATED WORK

Much of the research done on video streaming has focused on characterization of YouTube network traffic, video quality and user engagement. This section discusses various other studies done on YouTube, and video streaming related topics.

Gill, et al. [6], designed a study to monitor Web 2.0 sites; sites which allow users to upload and share their own content. They decided to focus on YouTube which is classified as a Web 2.0 site. Their work characterized YouTube traffic by monitoring both local and global video requests/response transactions and found that YouTube video quality is strongly correlated with traffic volumes. Our study used these findings as an aid to develop our methodology in order to capture the network traces with precision. We also considered this study when determining how buffering is affected by varying amounts of network traffic.

Zink, et al. [7] recorded the duration and data rate of streaming sessions, the popularity of videos, and access patterns for video clips from the clients in the network. Their study provided a detailed usage of YouTube by clients. Not only did many users repeatedly watch the same video, but they also found that there is no strong correlation between global and local popularity of videos. This study also looked at finding alternative distribution infrastructures in order to save bandwidth. They discovered that client based local caching, P2P based distribution, and proxy caching can allow for much faster access to videos, especially for users who repeatedly watch the same video. Our study also involved looking at YouTube traces on a large campus network but with a focus on how buffer size affects playback, rather than how to expedite video load times.

A viewer streaming video on a wireless network while mobile may often experience severe and frequent bandwidth fluctuations and outages. A Web media player has a built-in tool called the quality scheduler, designed to avoid video interrupts, and rapid quality switches while utilizing as much of the potential bandwidth as possible. Proper configuration of a media player's quality scheduler on mobile devices is challenging yet crucial in terms of user enjoyment. Riiser, et al. [8] studied various players, finding that the quality of the video was dramatically different based on the chosen player due to the fact that the players used different quality control techniques. This allows for other studies to investigate what the ideal tradeoff between factors like high average quality, stable quality, or fewest interrupts in order to maximize user enjoyment. Our study considered these findings when analyzing our data, particularly with the development of our cost metric.

User enjoyment of videos hinges on several factors including video resolution and ability for smooth playback. Dobrian, et al. [9] studied what variables contribute most to the ultimate viewing experience. The variables studied includes quality metrics such as initial wait time before video playback, average bit rate, and buffering ratio, content type, and quantitative measures of user engagement in terms of number of views per user and total play time. This study found that the percentage of time spent on buffering, also called the buffering ratio, is the most critical metric for users. Our study did not directly focus on video quality specifically but rather an aspect of the quality: buffering. Rather than finding to what extent buffering affects quality, we use similar metrics such as initial wait time and buffering ratio to evaluate the impact they have on smooth video playback.

Wang, et al. [10] performed a similar study in order to find the impact of resolution, frame rate, and quantization parameter on perceived video quality and bitrate. This study found that quantization distortion had a large impact on video quality for users and therefore large values for the quantization distortion should be avoided. However, by adjusting quantization distortion, frame rate, and resolution together they found that bit rate could be reduced over 50% for similar quality when compared to only adjusting quantization distortion. While this study investigates video specifications and their relations to each other, our study uses these as fixed and variable parameters while gathering YouTube traces.

Hoßfeld, et al. [11] modeled the impact of variable bitrate encoding on interrupts and derived an approximation for the initial delay in playback so that there would be no interruptions while playing a video. Using their data they produced a scatter plot of the optimal initial delay vs. the approximated delay, where approximately 25% of videos still had issues with interrupts. In our study, instead of modelling variable bitrate encoding, we evaluated cost metrics (initial delay, interrupt frequency, interrupted duration, etc.) and optimized the initial buffer so the viewer could get fewer interrupts at minimum initial delay.

Wireless Local Area Networks are commonly found on large campuses and increasingly in Wi-Fi hotspots. Henderson, et al. [12] performed the largest WLAN trace to date using data found at Dartmouth College. Their data were particularly useful for determining some of our testing parameters since our test environment is also a college campus. In particular, the data regarding network traffic vs. time of day revealed the busiest hours for the network.

YouTube uses a form of flow control called block sending, with the blocks typically being 64 KB in size. The amount of data sent during the initial buffering period is equivalent to 32 seconds of block sending. Alcock and Nelson [13] found that this technique interacts poorly with underlying TCP mechanisms which leads to an increase in packet loss. Their study found that over 40% of the packet loss observed by YouTube clients could be attributed to block sending. Our study observed how by varying the initial buffer a user can overcome these issues.

While streaming video is sensitive to bandwidth fluctuations, using a receiver buffer can reduce the effects on the video playback. Mastoureshgh [14] investigated the nature of the change in buffer size with respect to variations in bandwidth by modelling a video streaming system over TCP using simulation to develop a buffering algorithm. This study proposed using a dynamic client buffer size based on measured bandwidth in order to achieve better video playback performance. Using our simulator, we investigated how effective a dynamic buffer would be at reducing interruptions as well as producing a quality metric which could be used to compare videos based on aspects including: number of interrupts, total time interrupted, and initial buffer time.

Unlike earlier analysis and characterization of YouTube, our study attempts to further explore the relationships between the variables that dictate the rate videos are streamed and their effects on buffering in a wireless environment. Wireless network traces have never been taken for WPI's campus network and therefore this study could be of particular relevance if the infrastructure of the wireless network were to be redesigned in the future. Similarly our methodology in totality has never been done before; the concept of capturing network traces using a network sniffer for use in a simulator is inspired by the Mastoureshgh thesis [14], but our focus is on the wireless network and traces are instead captured moving along paths in multiple locations while using multiple video qualities.

3 METHODOLOGY

The effect of buffering on observed quality for the user compelled the design of an experiment in which wireless network traces over set paths could be captured and simulated in order to determine optimal buffer size. The traces are gathered on a Dell Latitude e6430 laptop running Windows 7 using the software Wireshark [15]. The data collected from Wireshark is used to calculate bitrate of the video used in the experiment. The parameters used for the experiment include video quality and type of path. The network data captured with Wireshark is put through a video playback simulator which is developed to take initial buffer size as an input argument in addition to the throughput of each trace and the bitrate, and outputs the number of interrupts, when they occur and how long they last. By simulating video playback with different buffer sizes using the collected network traces, we can find the optimal initial buffer size for streaming video with the fewest interruptions, and the least amount of initial waiting time.

3.1 DATA COLLECTION

Our study focuses on YouTube video streaming, while moving across a wireless network. In order to accommodate mobility in our experiment, we set up 4 different paths on the WPI campus, 2 indoors, and 2 outdoors, and gather the network traces on a laptop while walking along these paths at a constant pace.

3.1.1 Bandwidth Traces

We gather bandwidth traces with Wireshark [15]. Wireshark is an open-source network analyzer and a packet sniffer that captures messages being sent or received between a computer and a network. It can be used to capture live packets and filter them to get the exact information the user desires.

Our experiment begins by first choosing one of our paths. For now we will call these simply: Founders, Atwater-Kent (AK), Campus Center (CC) to Library, and Morgan to CC. Each of these paths is distinct and unique while descriptions of each path appear under the parameters subsection. After choosing a path the following steps lead to the capturing of a bandwidth trace which describes the network conditions observed on the path:

1. Open Wireshark
2. Close all other windows

3. Begin live capture with Wireshark
4. Open the YouTube video and begin playing at 360p. Note the time when the video began
5. Begin walking along the path while making sure connection to the network is not lost (If connection is lost at any point during live capture, the trial is restarted from the begining)
6. Record the time and duration of interrupts for the video while it is playing
7. Finish path
8. End live capture with Wireshark. Note the duration of the video that has been played and the amount of video which has been buffered
9. In Wireshark set filters for source IP address with the address of the laptop used and for destination IP with the address of the YouTube server
10. Save the Wireshark file to use for analysis later

Once the buffer packets are separated from the rest of the network traffic, the data can be exported for use in the simulator. This can be done by saving the file as a .csv and removing the excess columns of data in Excel. The two relevant pieces of information Wireshark records are the times the packets are received and the packet size. The amount of the video buffered can be found by summing the sizes of all the packets. This value determines how much of the video the simulator will play back for the given trace. The simulator also needs the time the packets arrived so that it can recognize bandwidth as a function of time.

3.1.2 Signal Strength

We gather signal strength data using the program WirelessMon [16]. This software records signal strength over time as well as the mac address and channel of the access point. The procedure to gather this data is similar to how the bandwidth traces were acquired. We have WirelessMon record data while walking along the path. At the end of the path we stop recording, which automatically saves the logs. Unlike the bandwidth traces, we only record one signal strength trial for each path.

3.2 Parameters

In order to observe the effects of environment on video streaming we conduct trials in both indoor and outdoor locations. Each of the chosen paths are common walkways for students and represent varying environments. The outdoor paths can be seen in Figure 1. All paths are described below:

1. *Atwater-Kent:* This path is indoors. The building is used for academic purposes. The building is rectangular and the path incorporates a staircase which lies in the corner. The path begins at the top of the staircase. It then continues in a rectangular fashion around the second floor of the building. When the path reaches the beginning point, the staircase is then taken to the first floor. The path then continues along the rectangular hallway around the first floor (directly underneath the second floor's path) and ends at the bottom of the staircase.
2. *Founders Hall:* This path is indoors. The building is residential. It is also the farthest from the center of campus. The path begins on the ground floor. The hall has two staircases on different sides of the building. The path starts at the bottom of the on staircase. It continues through the hall to the other staircase and up to the second floor. The path goes back and forth between the two staircases, going up a floor each time. It comes to an end after reaching and crossing the hall on the fourth floor.
3. *From Morgan Hall to the Campus Center (Figure 1, Building H to 7, yellow path)*: This path is outdoors. It begins at the entrance of Morgan, a residence hall. The path then continues out cutting diagonally across the quad. After the quad the path continues along next to a parking lot on the right and a Gymnasium on the left. The path takes a left and goes down an alley between the Gym and an academic building, heading towards the Campus Center. This path then ends outside the front of the CC. Due to the quad, this path ventures the farthest from adjacent buildings.
4. *From the campus center to the library (Figure 1, Building 7 to 14, green path)*: This path is outdoors. It begins in front of the campus center. The path is a straight line from the CC to the Library. It passes by multiple academic buildings close to either side of it. This path represents our most crowded outdoor path in terms of foot traffic.

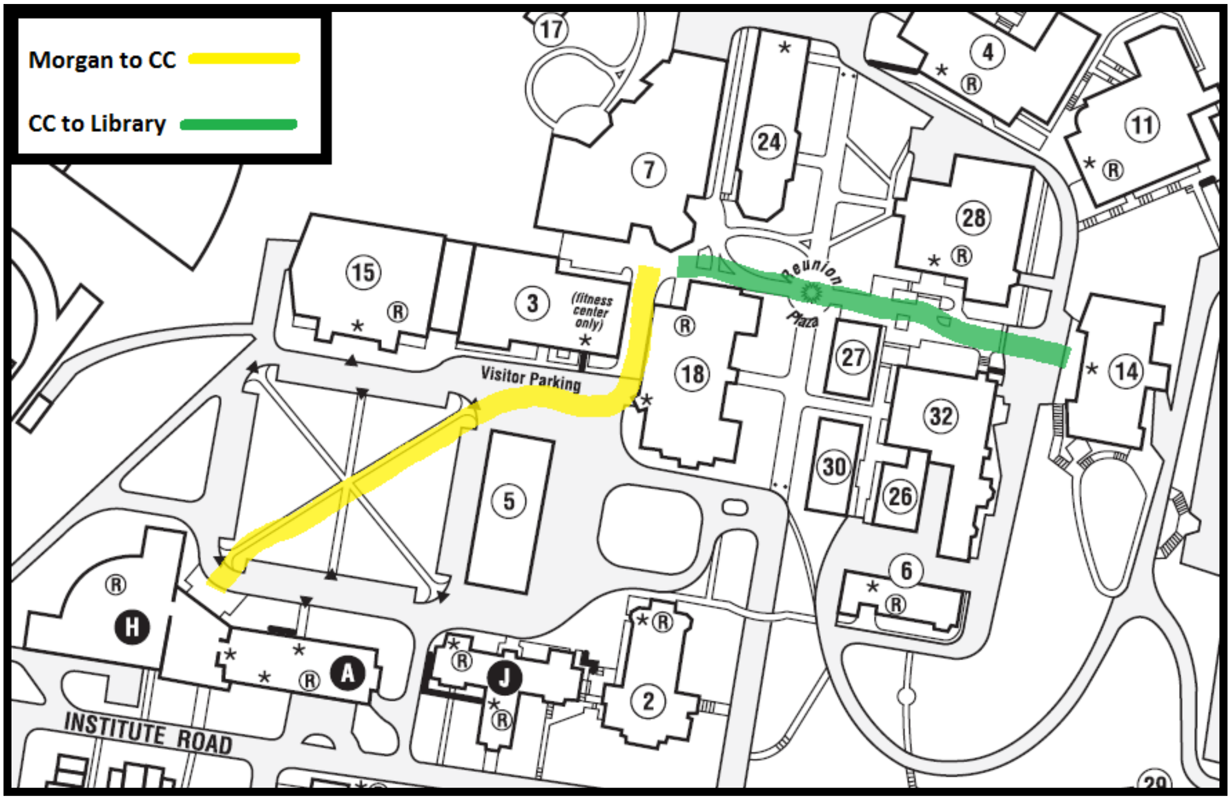

Figure 1: Campus map showing the two outdoor paths

Other factors that were taken into account for collecting mobile network traces were the walking speed and the time of day. In order to properly distinguish the effects of walking speed on the network traces, we performed pilot tests with two extreme speeds (running and walking very slowly). We found that the wireless connection would sometimes drop when running. Walking slowly had no different impact on the video playback than walking at a normal pace. Therefore we take our traces while walking at an average walking speed (about 3mph).

We also examined the network conditions at different times of day such as:

- Midday when there is the most network traffic [12] and most pedestrian traffic (the ten-minute interval between end of class and the start of next class).
- Midday when there is the most network traffic and least pedestrian traffic (when classes are in session).
- Early morning when there is the least network traffic [12].

We found that time of day had a significant influence on our wireless bandwidth traces. The average throughput was less in the afternoon than it was in the morning. During the time between classes, the throughput was even smaller. Thus, we chose to only capture traces between 1:00pm – 4:00pm. We skip the ten-minute interval between end of class and start of next class (the last ten minutes of every hour) because this would give us very little time to collect data. The afternoon is the time of day when network traffic is at its peak, which slows down the data transmission from the server and thus makes streaming video with few interrupts difficult.

3.3 Simulator

To determine the ideal buffer size, we utilize a video playback simulator that we have developed. The simulator uses bandwidth traces from Wireshark and the bitrate of the videos to find how many interrupts occur based on a given buffer size. The simulator is programmed in Matlab [17]. Wireshark can export captured data as Character Separated Values (.csv) which can be imported into Matlab. Figure 2 shows the code for a simple version of the simulator. When bytes are added to the buffer, the number of bytes added is based on the bandwidth trace. The trace is a function of time, so whenever bytes are added to the buffer, the time interval is incremented. We assume the bitrate of the video is constant since we have no means to determine the bitrate as a function of time. This version of the simulator returns three variables: the number of interrupts (`interrupts`), a matrix of when the interrupts occurred (`intTimes`), and a matrix of how long they were (`delay`).

```
function [interrupts, intTimes, delay] = bufSim(bitrate, trace, bufTime)

traceLength = size(trace, 1); % how long the trace is in seconds
time = 0;    % number of elapsed seconds
buffer = 0; % number of bits buffered
playedVideo = 0; % number of bits played
interrupts = 0; % number of interrupts
% number of bits needed to buffer after interrupt
rebufferSize = bufTime * bitrate;
intTimes = []; % times when interrupts occur
delay = []; % length of every interrupt including initial buffer time

% fill the initial buffer
while ((bufTime > time) && (time < traceLength))
      time = time + 1;
      buffer = buffer + trace(time);
end

delay(1) = time; % length of initial delay

while (time < traceLength) % while the trace isn't over
      if playedVideo > buffer % if the played video exceeded the buffer
            interrupts = interrupts + 1;
            intTimes(interrupts) = time;
            while ((buffer <= rebufferSize + playedVideo) % rebuffer
                   && (time < traceLength))
                  time = time + 1;
                  buffer = buffer + trace(time);
            end
            % calculate length of interrupt
            delay(interrupts + 1) = time - intTimes(interrupts);
      end

      % play video and buffer
      while ((playedVideo <= buffer) && (time < traceLength))
            time = time + 1;
            playedVideo = playedVideo + bitrate;
            buffer = buffer + trace(time);
      end
end
```

Figure 2: Simulator Code

4 ANALYSIS

4.1 RESULTS

Figure 3 is a snapshot of the Wireshark interface that we use in gathering the network traces for streaming video. Wireshark produces several columns of incoming network signal information. The data that is relevant to our study are the length and time. The length is the size of the packet in bytes. The time column is the time at which each packet is received relative to when data began being recorded. The source and destination columns show their respective IP addresses. They serve as the criteria for filtering the video data from all the other packets being sent over the network.

	A	B	C	D	E	F	G	H	I	J
1	No.	Source	Time	Destination	Protocol	Length	Info			
2	1559	74.125.9.4	2.624585	130.215.168.50	TCP	66	http > af [SYN, ACK] Seq=0 Ack=1 Win=29200			
3	1579	74.125.9.4	2.653003	130.215.168.50	TCP	56	http > af [ACK] Seq=1 Ack=372 Win=30272 Le			
4	1580	74.125.9.4	2.654043	130.215.168.50	HTTP	605	HTTP/1.1 200 OK (text/x-cross-domain-polic			
5	1690	74.125.9.4	2.689407	130.215.168.50	HTTP	194	HTTP/1.1 204 No Content			
6	1802	74.125.9.4	2.906798	130.215.168.50	HTTP	194	[TCP Retransmission] HTTP/1.1 204 No Conte			
7	1845	74.125.9.4	3.009757	130.215.168.50	TCP	66	http > innosys-acl [SYN, ACK] Seq=0 Ack=1 W			
8	1848	74.125.9.4	3.009909	130.215.168.50	TCP	546	[TCP segment of a reassembled PDU]			
9	1850	74.125.9.4	3.010217	130.215.168.50	TCP	1514	[TCP segment of a reassembled PDU]			
10	1852	74.125.9.4	3.010525	130.215.168.50	TCP	1514	[TCP segment of a reassembled PDU]			
11	1853	74.125.9.4	3.010788	130.215.168.50	TCP	1514	[TCP segment of a reassembled PDU]			
12	1855	74.125.9.4	3.011822	130.215.168.50	TCP	1514	[TCP segment of a reassembled PDU]			
13	1856	74.125.9.4	3.013173	130.215.168.50	TCP	1514	[TCP segment of a reassembled PDU]			

Figure 3: Example Wireshark Data

Figure 4 is an excerpt from a connected-node data log produced by the wireless signal monitoring application, Wirelessmon. It contains seven columns, three of which are relevant to our study: the MAC address, signal strength, and the channel. The MAC address is the hardware address of the access point the laptop is communicating through. Whenever there is a change in MAC address, e.g, between row six and seven, it is an indication that a handoff has occurred and the mobile device is now connected to a different access point. This is relevant because there is typically a drop in throughput when a handoff occurs. The channel represents the frequency range in which the device is operating, out of the available Wi-Fi frequency range of 2.4 GHz. The strength is the signal strength of the network at the connected node. Lower absolute value of the signal strength indicates a stronger signal.

Index	Time	SSID	MAC	Channel	Percentage(%)	Strength(dBm)
0	16:28:31:730 29-Jan-2014	WPI-Wireless	4C-96-14-20-07-80	1	31	-65
1	16:28:33:889 29-Jan-2014	WPI-Wireless	4C-96-14-20-07-80	1	33	-63
2	16:28:36:258 29-Jan-2014	WPI-Wireless	4C-96-14-20-07-80	1	32	-64
3	16:28:38:417 29-Jan-2014	WPI-Wireless	4C-96-14-20-07-80	1	30	-66
4	16:28:40:506 29-Jan-2014	WPI-Wireless	4C-96-14-20-07-80	157	30	-66
5	16:28:42:665 29-Jan-2014	WPI-Wireless	4C-96-14-20-07-80	1	25	-70
6	16:28:45:163 29-Jan-2014	WPI-Wireless	4C-96-14-20-07-80	157	22	-72
7	16:28:47:322 29-Jan-2014	WPI-Wireless	78-19-F7-78-E6-42	1	53	-47
8	16:28:50:901 29-Jan-2014	WPI-Wireless	78-19-F7-78-E6-42	1	42	-56
9	16:28:54:089 29-Jan-2014	WPI-Wireless	78-19-F7-78-E6-42	1	40	-58
10	16:28:56:468 29-Jan-2014	WPI-Wireless	78-19-F7-78-E6-42	1	37	-60
11	16:28:58:607 29-Jan-2014	WPI-Wireless	78-19-F7-78-E6-42	1	37	-60

Figure 4: Example Wirelessmon Data

Tables 1, 2, 3, and 4 below, show data about the interrupts that occurred during the traces. Both the number of interrupts and the average length of those interrupts are recorded on the tables. Playback is the number of seconds of video that played during the trial. Buffer is the number of seconds of video that was buffered, including the video that was played. This value is approximate and was found just by hovering the mouse cursor over the end of the grey bar that indicates the buffer in the video player.

AK					
Quality	Trial	# of Interrupts	Average Interrupt Length (s)	Playback (s)	Buffer (s)
360p	1	4	11	140	200
	2	0	0	163	421
	3	0	0	165	372
480p	1	3	9.67	132	187
	2	0	0	165	431
	3	0	0	174	409
720p	1	1	1	162	199
	2	0	0	174	301
	3	1	21	145	222
1080p	1	8	6	130	130
	2	8	6.75	110	110
	3	6	7.83	116	118

Table 1: AK Interrupts

Founders					
Quality	Trial	# of Interrupts	Average Interrupt Length (s)	Playback (s)	Buffer (s)
360p	1	0	0	131	285
	2	0	0	122	390
	3	0	0	115	180
480p	1	1	2	108	273
	2	0	0	122	160
	3	0	0	105	133
720p	1	0	0	140	140
	2	2	7.5	98	98
	3	2	5.5	111	133
1080p	1	7	6	62	62
	2	6	7	79	79
	3	5	14.6	56	56

Table 2: Founders Interrupts

Morgan to CC					
Quality	Trial	# of Interrupts	Average Interrupt Length (s)	Playback (s)	Buffer (s)
360p	1	0	0	175	452
	2	0	0	178	447
	3	0	0	173	452
480p	1	1	NA	144	144
	2	6	7.6	131	131
	3	0	0	177	200
720p	1	2	1	128	128
	2	2	5	174	330
	3	2	21.5	120	152
1080p	1	5	2	110	110
	2	2	14	94	94
	3	4	6.33	69	69

Table 3: Morgan to CC Interrupts

CC to Library					
Quality	Trial	# of Interrupts	Average Interrupt Length (s)	Playback (s)	Buffer (s)
360p	1	0	0	137	410
	2	0	0	130	340
	3	0	0	129	340
480p	1	0	0	135	320
	2	0	0	137	430
	3	0	0	137	335
720p	1	2	20.5	110	110
	2	0	0	135	165
	3	3	19	75	75
1080p	1	1	32	85	85
	2	8	2.75	62	62
	3	6	10.67	45	45

Table 4: CC to Library Interrupts

The interrupt scatter plots in Figures 5-8 display interrupt lengths versus time for all three trials of each path at 1080p. The plot shows when interrupts occur during playback and how long each interrupt lasts. The y-axis of the graph is the length of interrupt and the x-axis shows the time when the interrupt occurs relative to the playback. These data points were recorded by hand while gathering the traces and are only approximate.

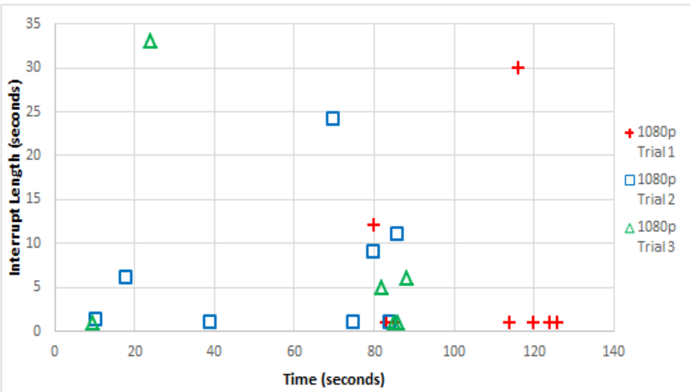

Figure 5: AK Interrupt Occurrences

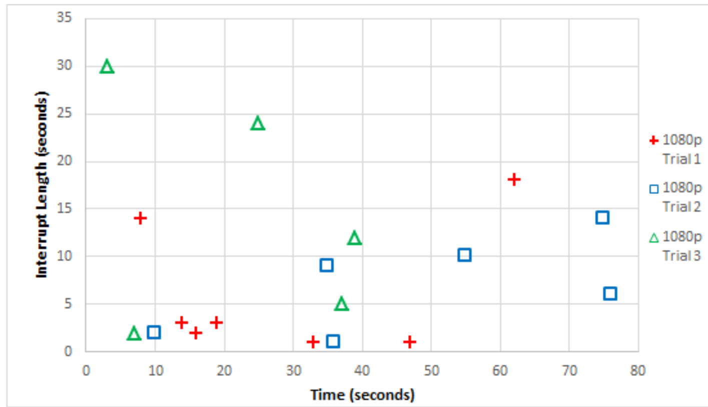

Figure 6: Founders Interrupt Occurrences

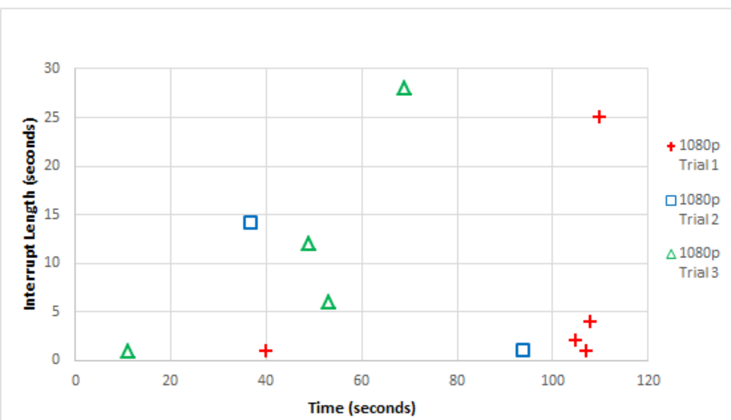

Figure 7: Morgan to CC Interrupt Occurrences

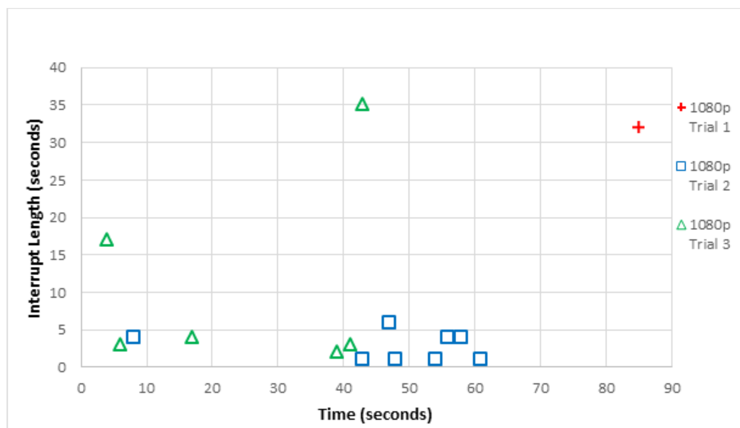

Figure 8: CC to Library Interrupt Occurrences

4.2 Trace Graphs

Figure 9 compares the throughput in kilobits per second across the three trials taken along the path in AK at 1080p. At 1080p, the traces are observed to be following similar patterns of fluctuation. However, some traces have zero throughput for certain time intervals while the other trials are still downloading video. For example, between 20-60 seconds trial 3 is not downloading, but trials 1 and 2 are still getting data. For the 1080p trials, both the maximum throughput and average throughput are higher than that of the lower qualities.

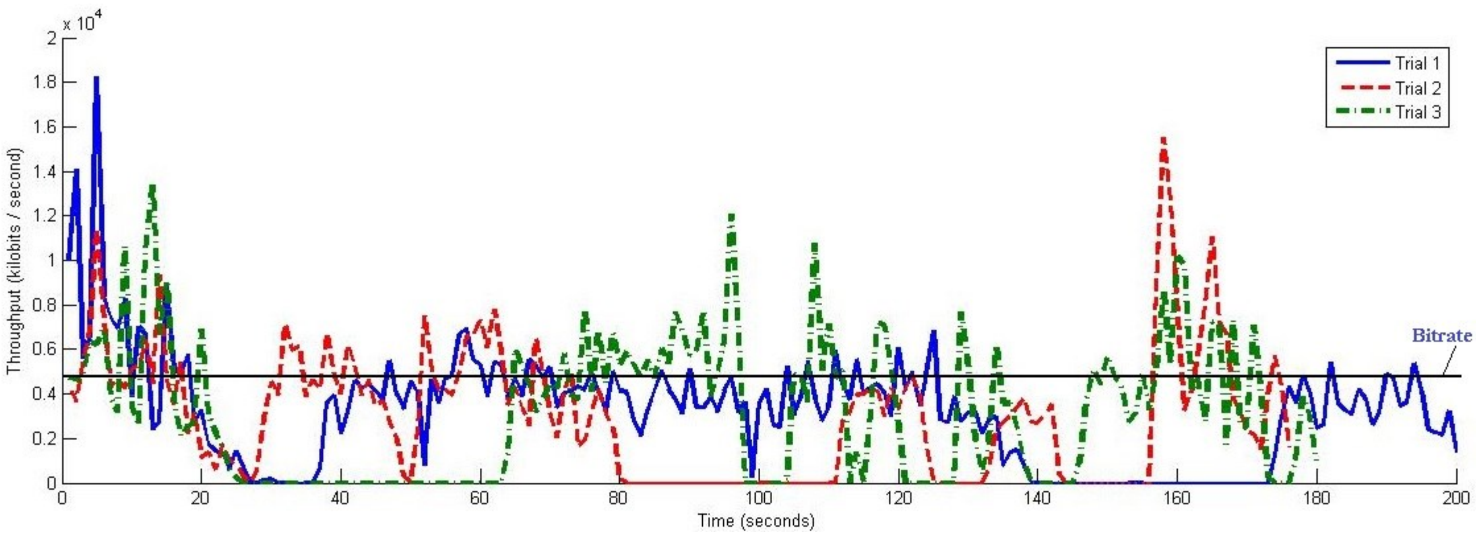

Figure 9: AK Throughput at 1080p

Figure 10 is the comparison of three trials at 360p. The average throughput is lower than that of higher qualities. Throughput fluctuations still exists at 360p, which is the lowest video quality that we used for our study, and the traces are not at all consistent between the three different trials. Most of the traces at 360p had no interrupts; however, the exception is trace 1 for AK which had 4 interrupts in total with average interrupt duration of 11 seconds. This behavior explains the low throughput of trace 1 on the graph throughout the entire duration of the trial. From observing trace 1, it can be perceived that even at the lowest quality, the video playback with Wi-Fi tends to have unpredictable disruptions from outside impacts like surrounding environment and human traffic which cannot be regulated or controlled.

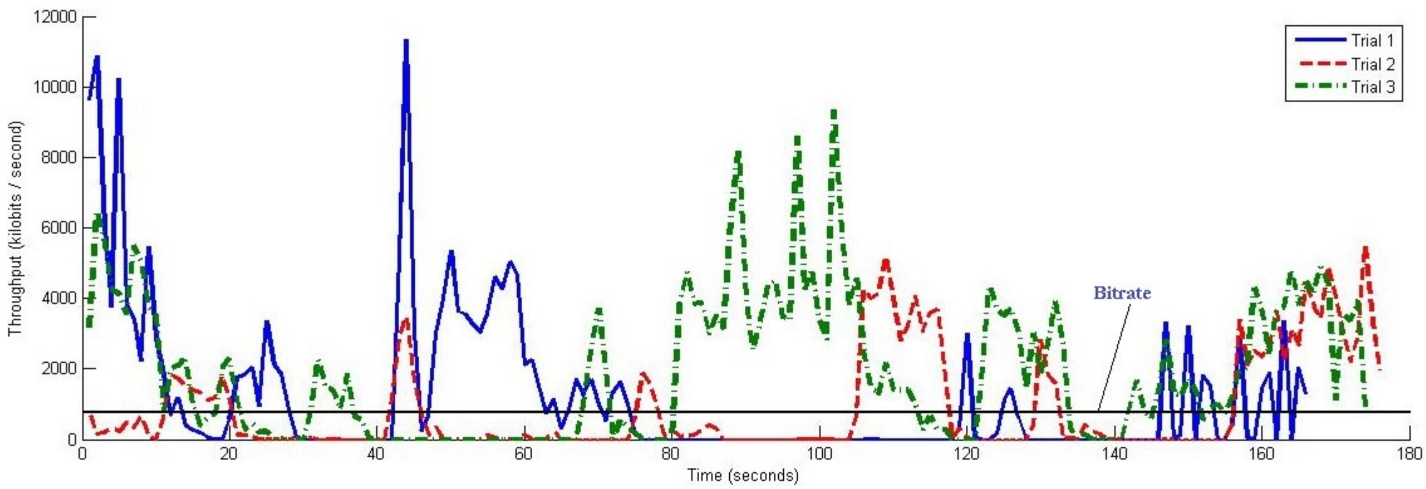

Figure 10: AK Throughput at 360p

4.3 Path comparisons

4.3.1 Throughput

The graphs below in Figures 11-14 compare the throughput in one second intervals for all four paths, and for each quality. The dashed lines are outdoor paths and the solid ones indoors. Atwater Kent (AK) has the longest path length out of all the paths and therefore has the longest session time, which explains why the throughput line for AK is longer than the rest of the paths.

The respective bitrate of the video for each quality is represented by the solid, straight line and serves as a reference point to observe how well each trace is doing. It can be seen that the throughput starts out at a point much higher than the bitrate except for 1080p. Then, as the wireless device starts to move along the path, the incoming signals fluctuate and the throughput moves down closer to the bitrate line. As the quality goes higher, the average throughput gets closer to the bitrate, and at times, below the bitrate.

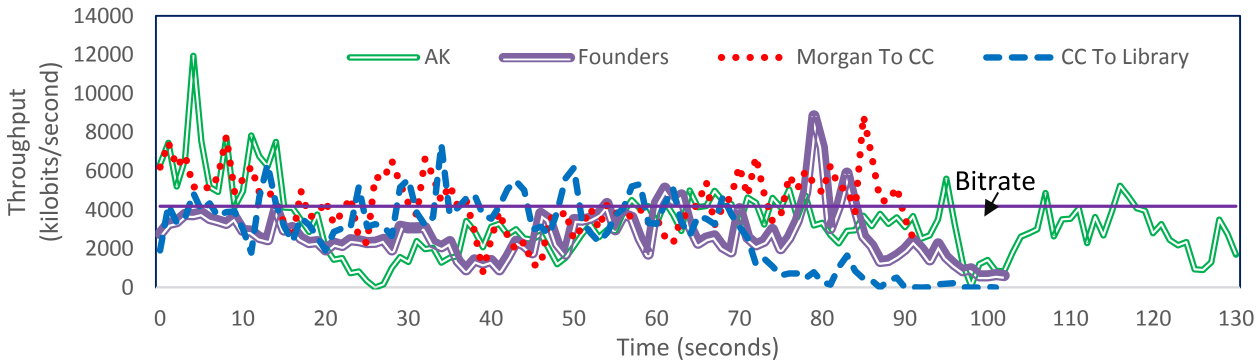

Figure 11: Average Throughput at 1080p

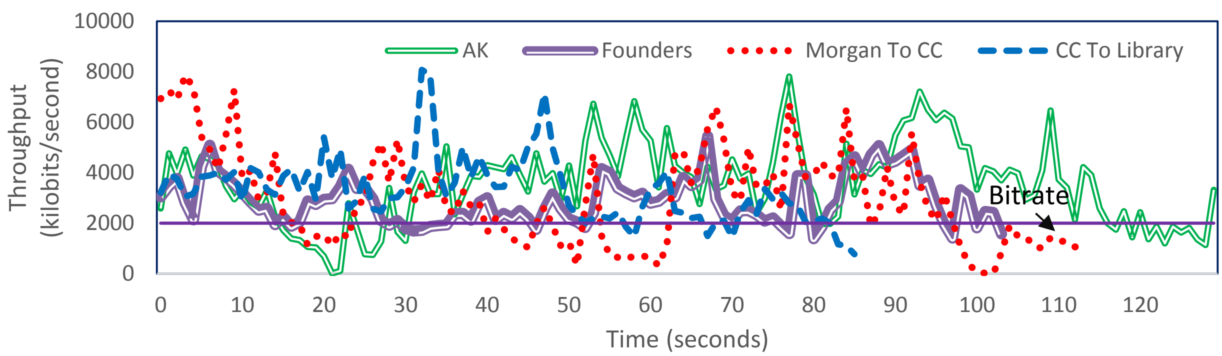

Figure 12: Average Throughput at 720p

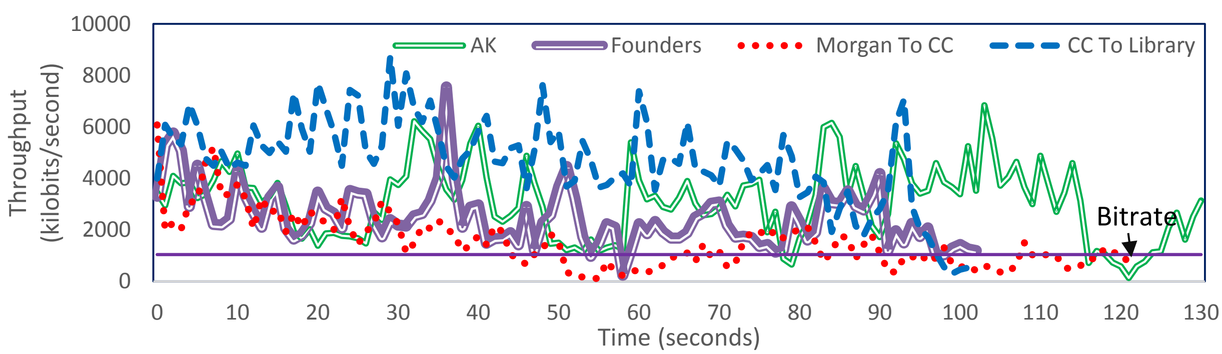

Figure 13: Average Throughput at 480p

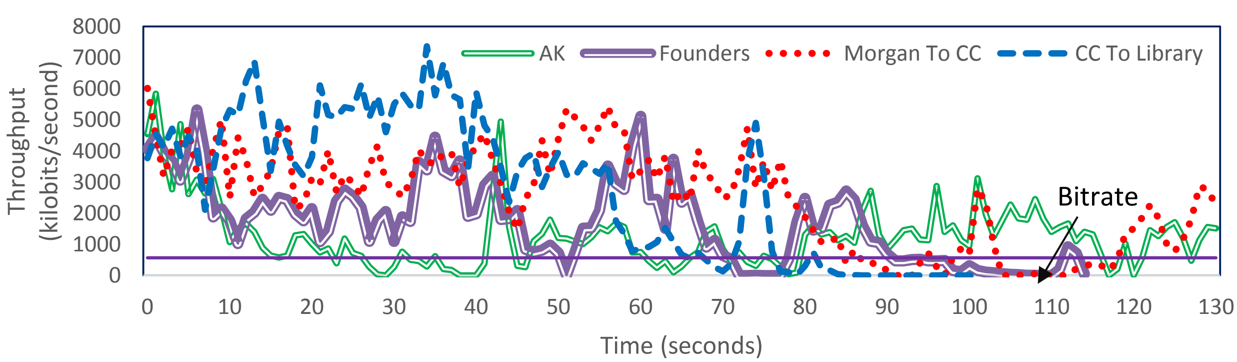

Figure 14: Average Throughput at 360p

The cumulative distribution graphs in Figures 15-22 show how the packets were delivered along each path, where each line represents an average of three trials. There are eight of these graphs, four of which compare qualities while keeping path constant and four which compare paths while keeping quality constant. Although it appears as if there is a large amount of variance between the graphs there are several patterns. In terms of distribution with regards to quality, there is a distinct trend that the lowest quality, 360p, always has the highest percentage of small packets. This phenomenon is represented in Figures 13-16 where the proportions of small bitrates for 360p rise quickly above those of the other qualities.

Conversely on these same four graphs it is evident that the 1080p trials generally downloaded the largest amount of data per second. Both of these observations make intuitive sense as the 360p trials would buffer so fast that they would wait to receive more packets, leading to a higher percentage of 0 of periods with little throughput. On the other hand, the 1080p trials should consistently have a high throughput since these trials have the largest bit rate. Thus it is unlikely for a 1080p trace to buffer so much that it can afford to stop downloading.

These graphs also lend themselves to comparing indoor versus outdoor paths. With the exception of 360p and 480p, all the paths show very similar behavior. In the 480p graph (Figure 21), every path varies significantly from one another. In particular, the two outdoor paths vary the most as they are the two extremes. This further supports the notion that classifying a path as either outdoor or indoor tells nothing significant about the expected behavior of streaming video along that path.

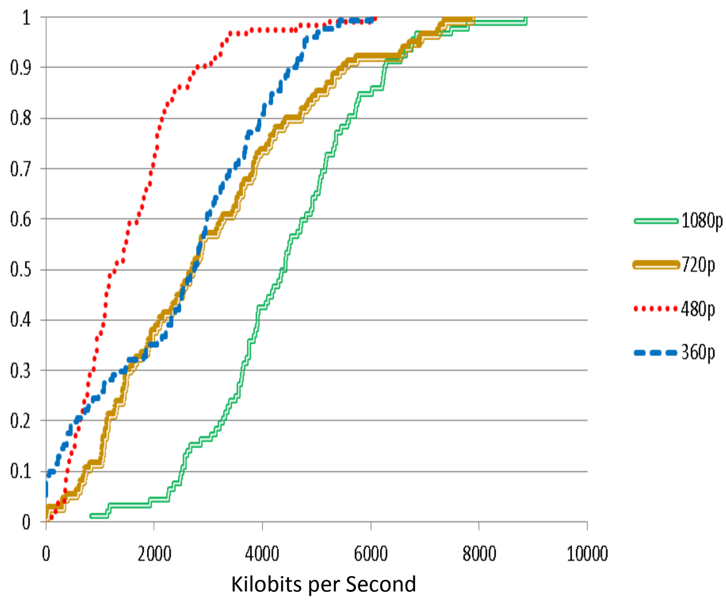

Figure 15: Morgan to CC Bitrate CDF

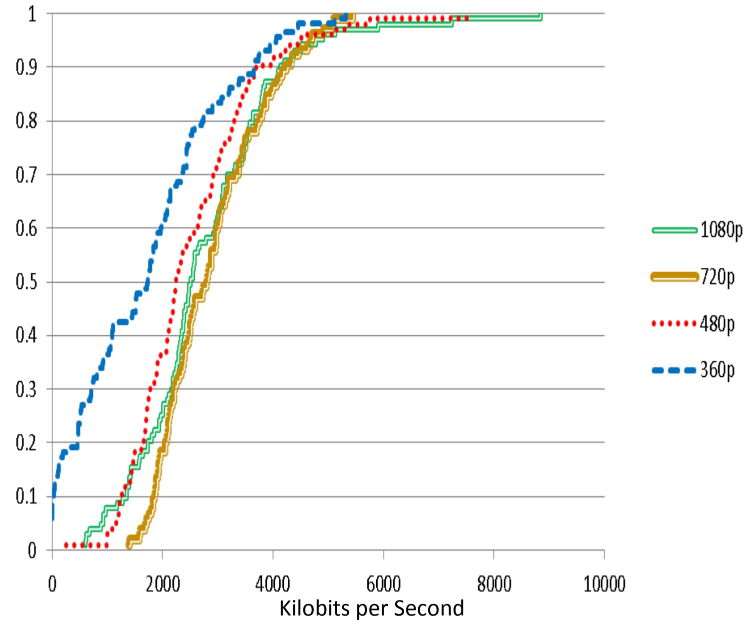

Figure 16 Founders Bitrate CDF

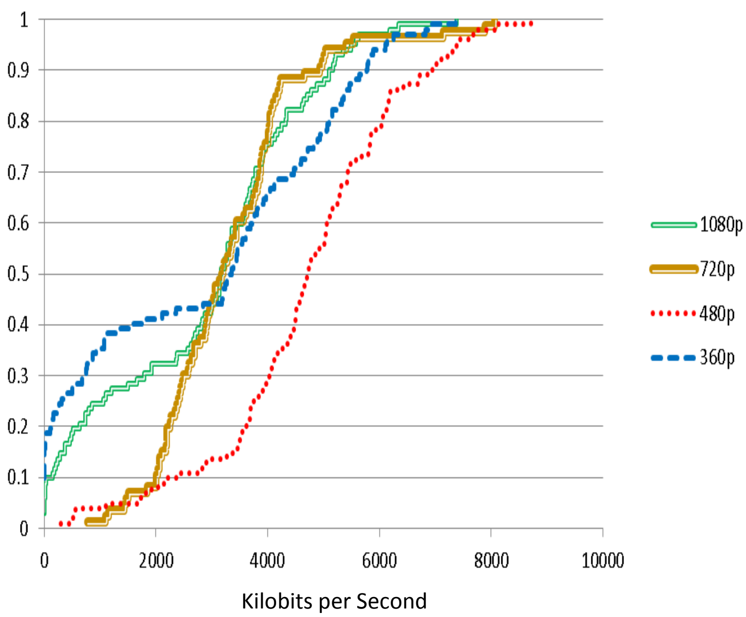

Figure 17 CC to Library Bitrate CDF

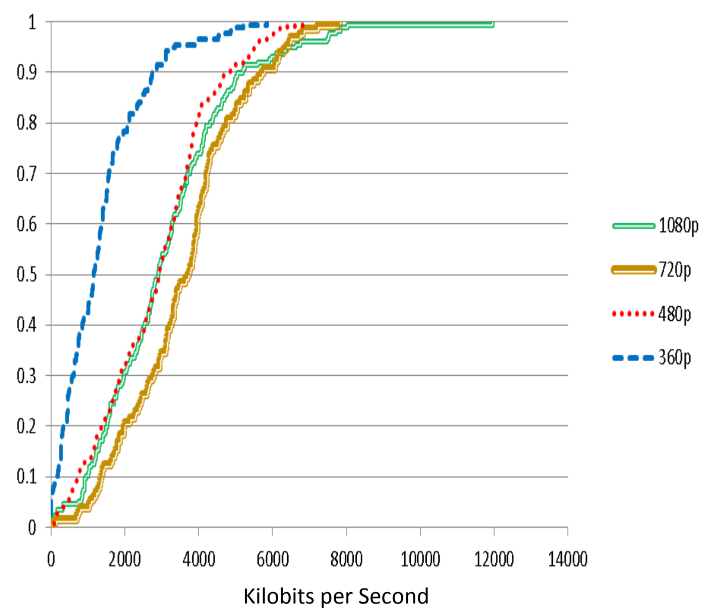

Figure 18 AK Bitrate CDF

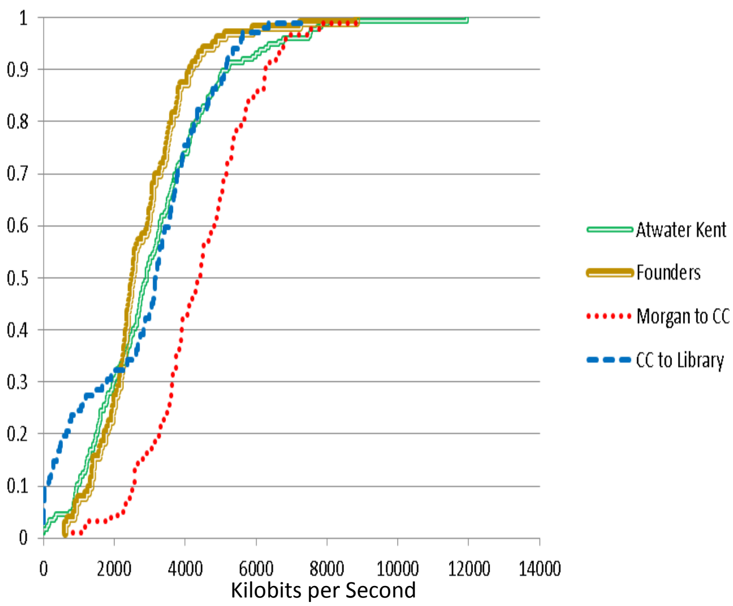

Figure 19: 1080p Bitrate CDF

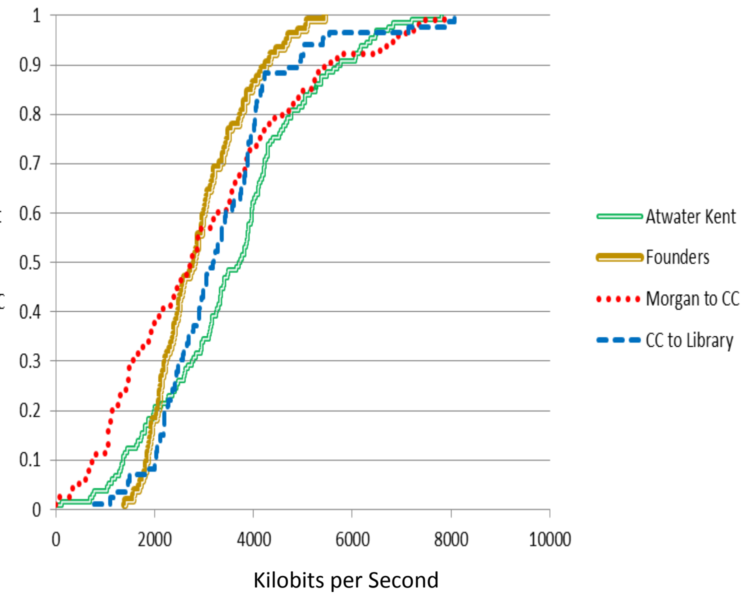

Figure 20: 720p Bitrate CDF

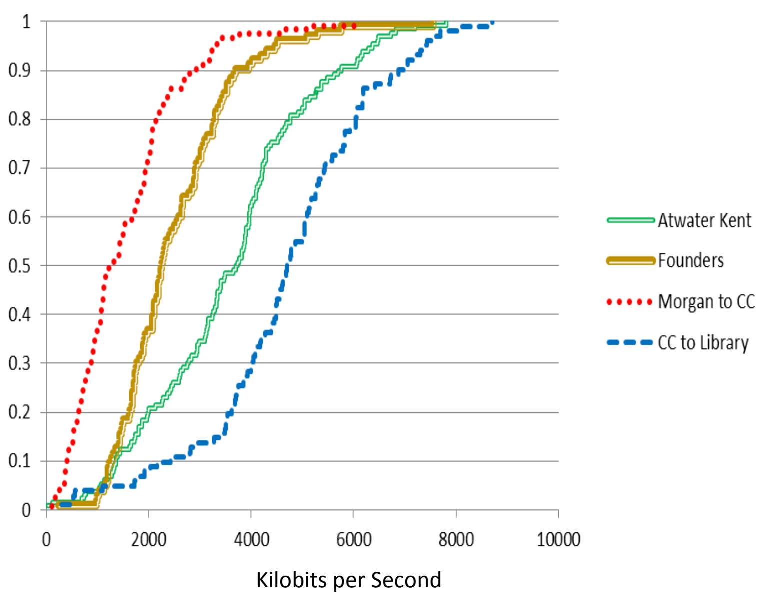

Figure 21: 480p Bitrate CDF

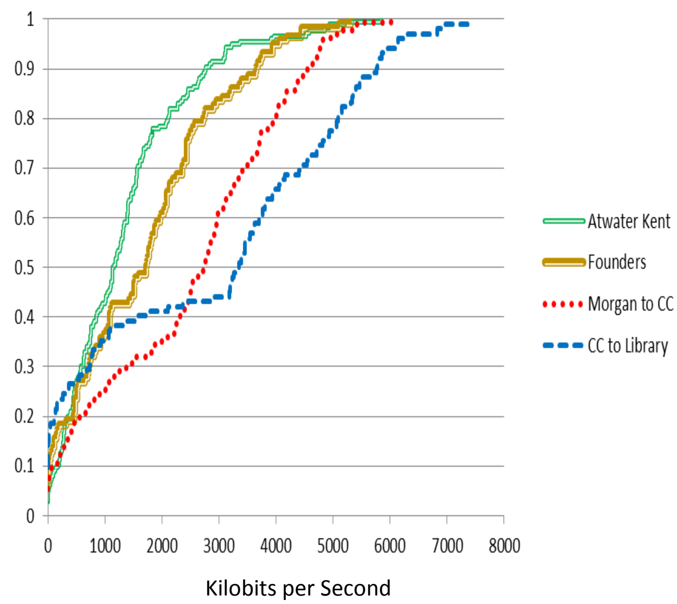

Figure 22: 360p Bitrate CDF

The average throughput for each quality on each path shows the effect of quality on average throughput. The graph in Figure 23 shows the average throughput for each path at each quality. The throughput is represented as a percentage of the bitrate at the given quality. As the quality is increased, the throughput percentage decreases. The average throughput generally is at least 100% for all paths when the quality is below 1080p. Even though the average throughput at 720p was above 100% for three paths, each of those paths had traces with interrupts at 720p. This shows that even if the average throughput is above the bitrate, that does not necessarily mean that no interrupts occurred.

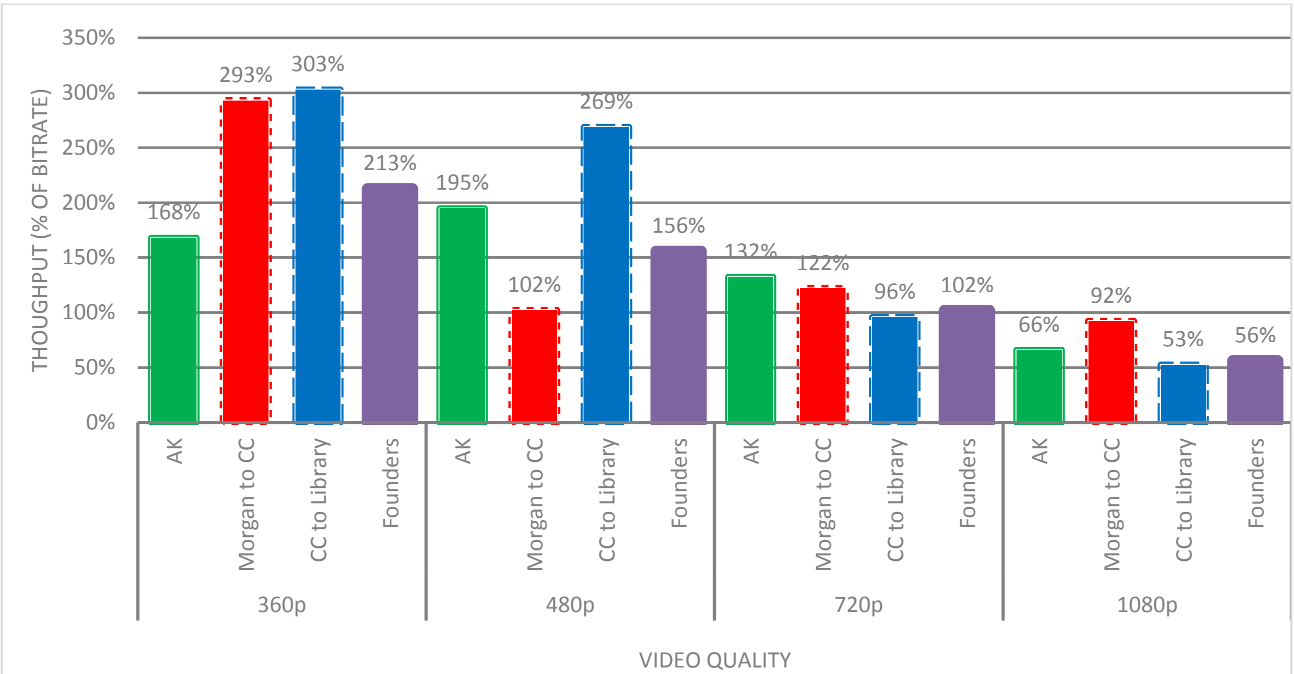

Figure 23: Average Throughput Bar Graph

4.3.2 Total Buffer

Figure 24 is a graph that shows how many seconds of video was buffered as a percentage of the video played. The video buffers far more than the video played when the quality is 360p and 480p in all the four paths, and thus resulting in buffers above 200% of the video played; even the 720p video buffered beyond what it played despite those trials often having interrupts. As the video quality increases, the total buffer decreases. In 1080p, all four paths have 100% buffers which mean the video only buffered as far as it played. This is an indication of frequent interrupts.

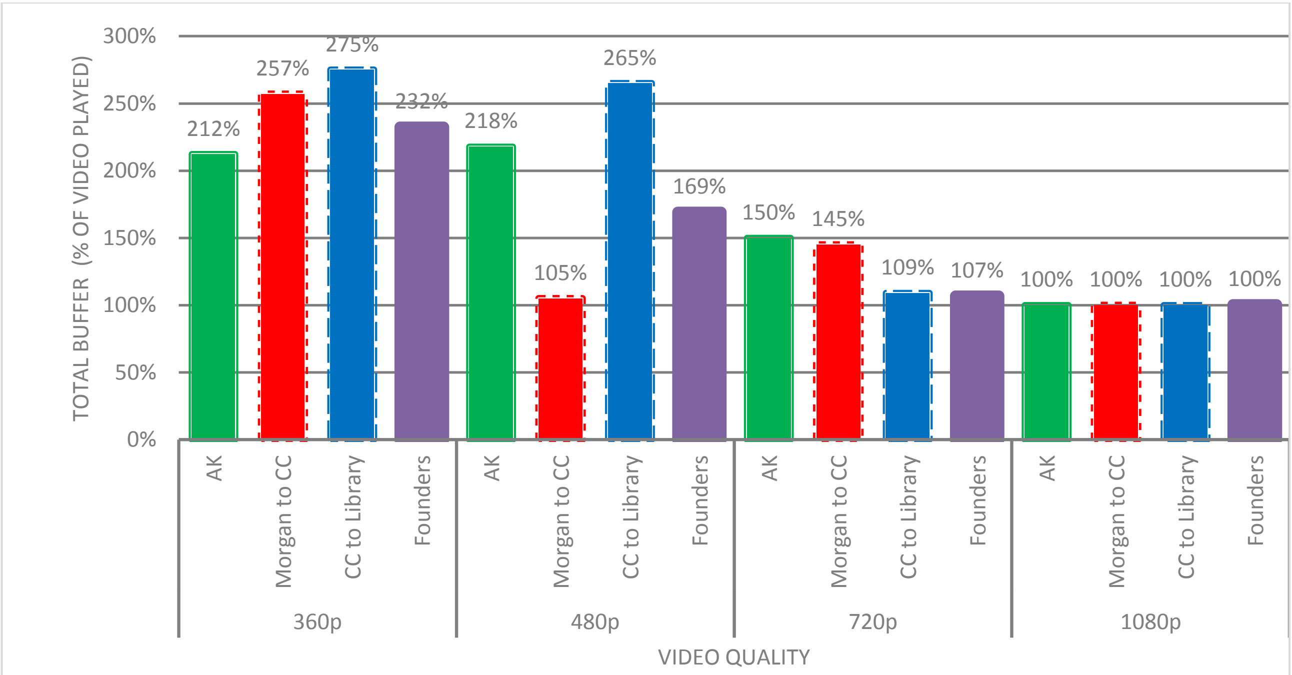

Figure 24: Total Buffer Bar Graph

4.3.3 Interrupts

The graph in Figure 25 is a scatter plot of the average length of an interrupt for each path against the average total number of interrupts of each trial for the 1080p video. Each data point represents one trial. A downward linear relationship can be seen, especially for AK and CC to Library. The interrupts tend to get shorter as they increase in number. Since the interrupted time is the number of interrupts multiplied by the average interrupt length, this linear trend indicates that the interrupted time for a given path remains fairly constant between trials. I.e. there are either few but long interrupts, many but short interrupts, or something in between. This means that the area under the trend line for a path should be that path's interrupted time. This appears to hold true when comparing this scatter plot to Figure 26.

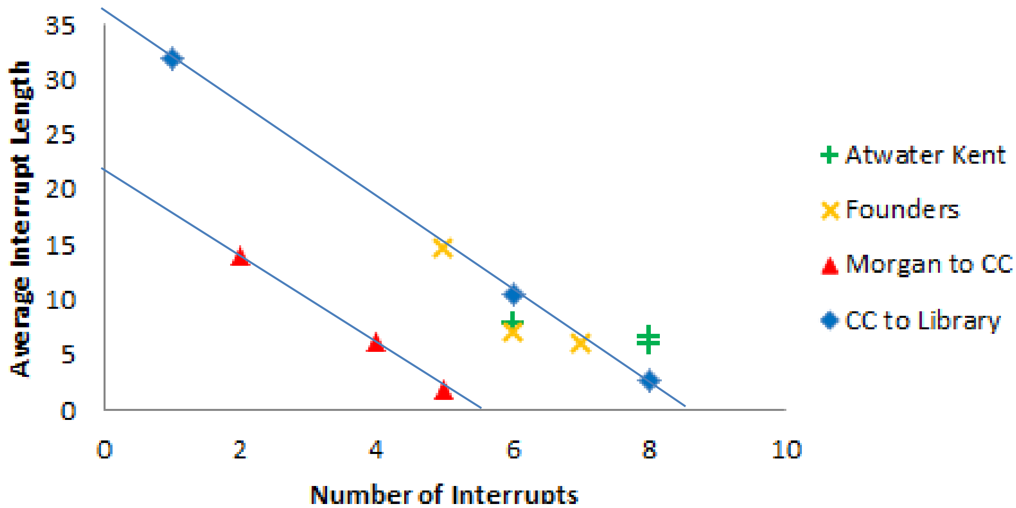

Figure 25: Interrupt Comparison by Path at 1080p

Figure 26 shows the average interrupted time as a percentage of the session time. The total interrupted time is generally observed to be increasing as the video quality gets higher. The interrupted times for outdoor paths (Morgan to CC and CC to Library) are higher than that of indoor paths for 720p but the exact opposite was observed for 1080p. This supports the idea that a path being outdoor or indoor does not significantly impact performance since there is no clear pattern.

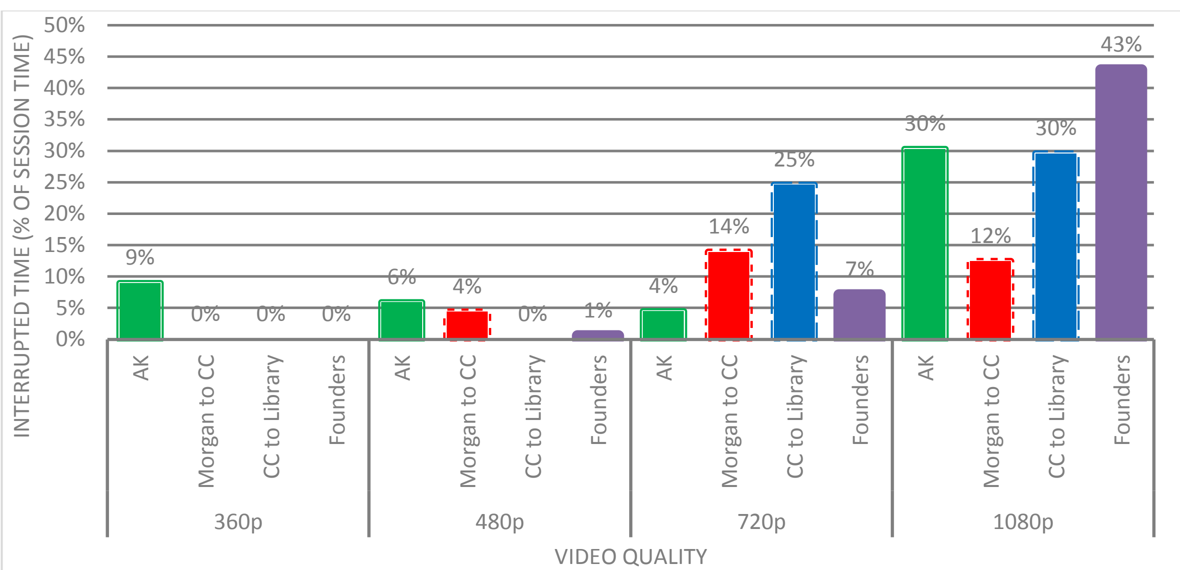

Figure 26: Interrupted Time Bar Graph

4.4 SIMULATOR

Playback graphs describe the different states in time of the video playback of each trial. These states are the initial buffering period, playback, and interruption. An example playback graph is shown below in Figure 27. Playback graphs are produced using results from the simulator to show how the video playback changes as the initial buffer size increases. The x-axis is the initial buffer time. The graph is plotted until there are no more interrupts.

A typical video playback behavior observed from these simulated playbacks is that the interrupts are shorter and more common when the initial buffer size is small. The interrupt length gets longer as the initial buffer size increases but the frequency of interrupts decreases. This is partially a consequence of how the simulator re-buffers when an interrupt occurs. When there is an interrupt, the simulator will not begin playback until there is a buffer the size of the initial buffer. This means that as the initial buffer increases, the time it takes for playback to resume after an interrupt increases. We use the data from the playback graphs in a cost function to determine the optimal buffer size.

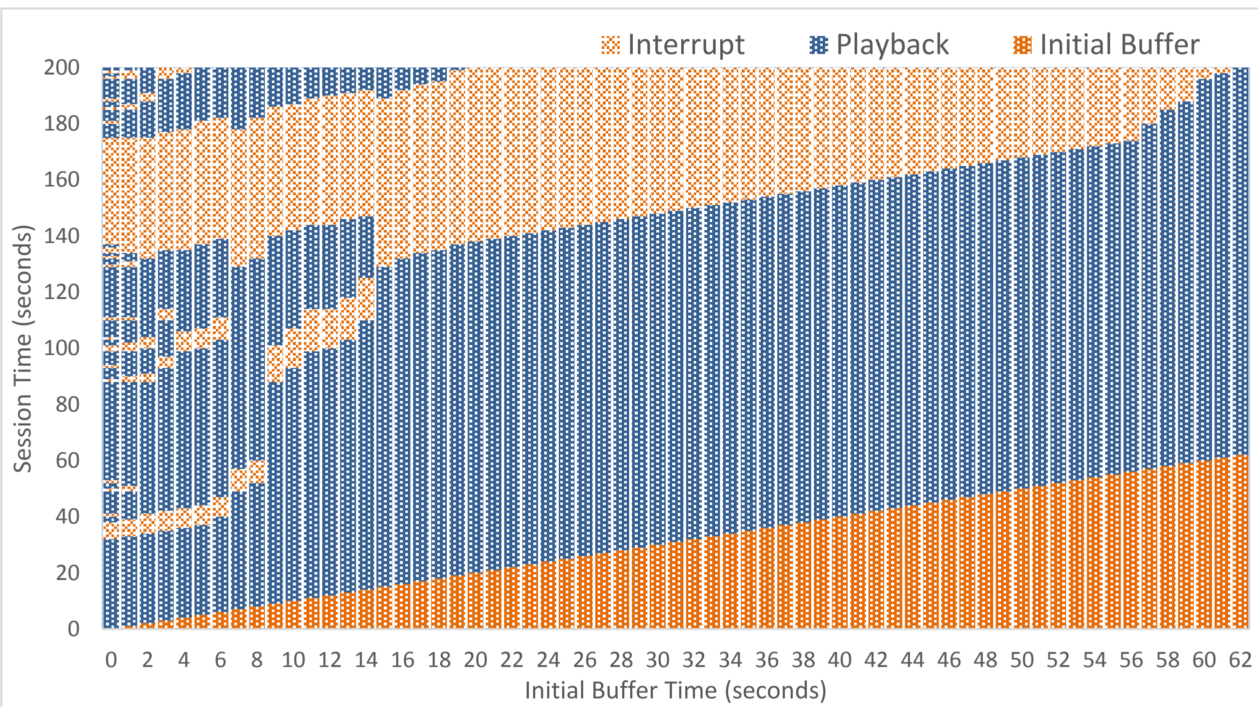

Figure 27: Playback Graph for AK Trial 1 at 1080p

The cost function is a means to determine the quality of the viewing experience. It uses four different cost metrics: initial delay, interrupt frequency, total interrupted time and the time at which the first interrupt occurs. The variables are weighted according to the intensity of the impact that each of them has on the viewing experience of the user and then added up to produce the total cost for the user. Figure 28 shows an example output of the cost function. The data used was from the playback graph above in Figure 27. The x-axis is the initial buffer time as a percentage of the session time. Since this is a plot of the cost, the lower values equate to a better viewing experience.

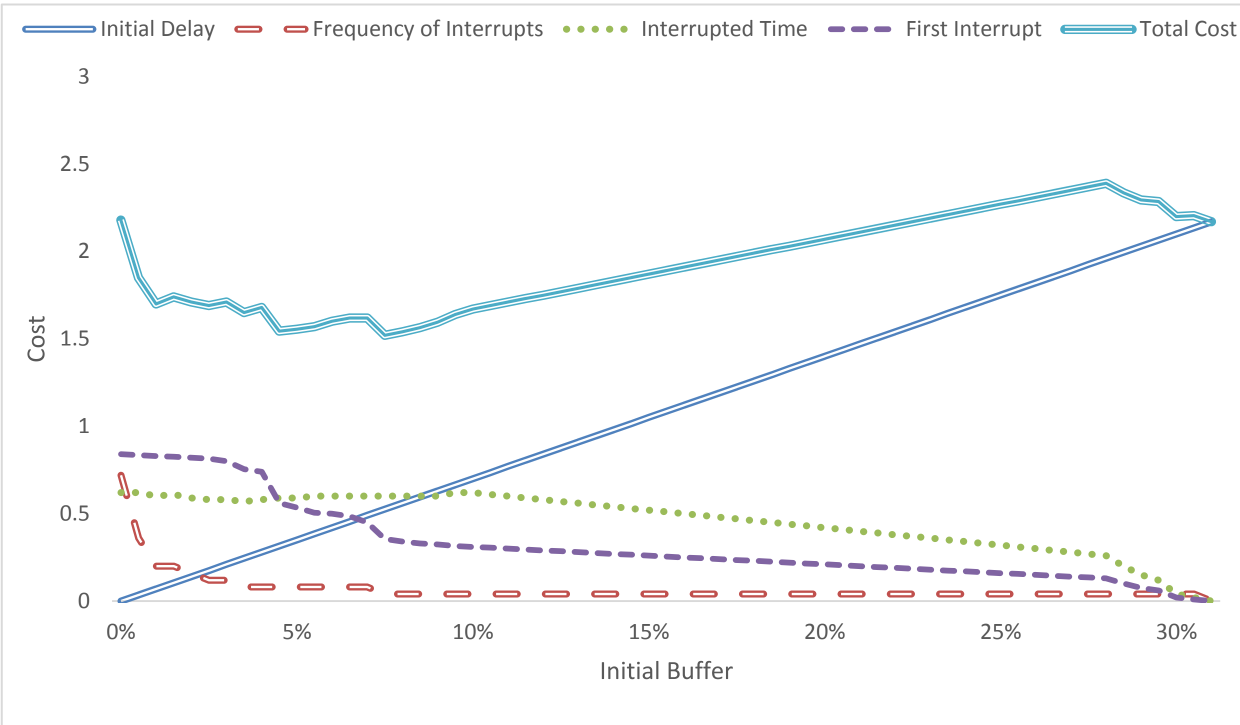

Figure 28: Cost Graph for AK Trial 1 at 1080p

The formula for the total cost is the following:

$$C = \frac{k_i i + k_n n + k_l l + k_f(f - s)}{s}$$

Table 5 below describes each variable. The "k" values are constants that determine the impact of each metric on the overall cost.

Variable Name	Description	Value
i	The initial delay in seconds	Input
n	The number of interrupts that occur	Input
l	The length of interrupted time in seconds	Input
f	The time at which the first interrupt occurs in seconds	Input
s	The session time in seconds	Input
k_i	The initial buffer constant	7
k_n	The interrupt constant	8
k_l	The interrupted time constant	2
k_f	The first-interrupt constant	1

Table 5: Cost Function Variables

The quality rating comes directly from the cost. Its purpose is to be more intuitive than the cost function. The quality has a value from zero to one, with one being the highest quality in terms of video viewing experience and being zero the lowest. As the cost goes higher, the quality goes down. A quality of one would only occur if there were no interrupts with no initial buffer. A quality of zero would be if there was no initial buffer, and an interrupt occurred every other second.

The peak value in the quality graph represents the best possible viewing experience. Thus, the optimal initial buffer size is the buffer size at a peak in quality. The following graphs in Figures 29-32 show the quality rating against the initial buffer size for every trace that had interrupts. The formula for the quality is as follows:

$$Q = 1 - \frac{C}{7}$$

The cost is "C". It is divided by seven, because seven is the highest posible cost. The value seven comes from the constants in Table 5. Different values for these constants would result in a different maximum cost.

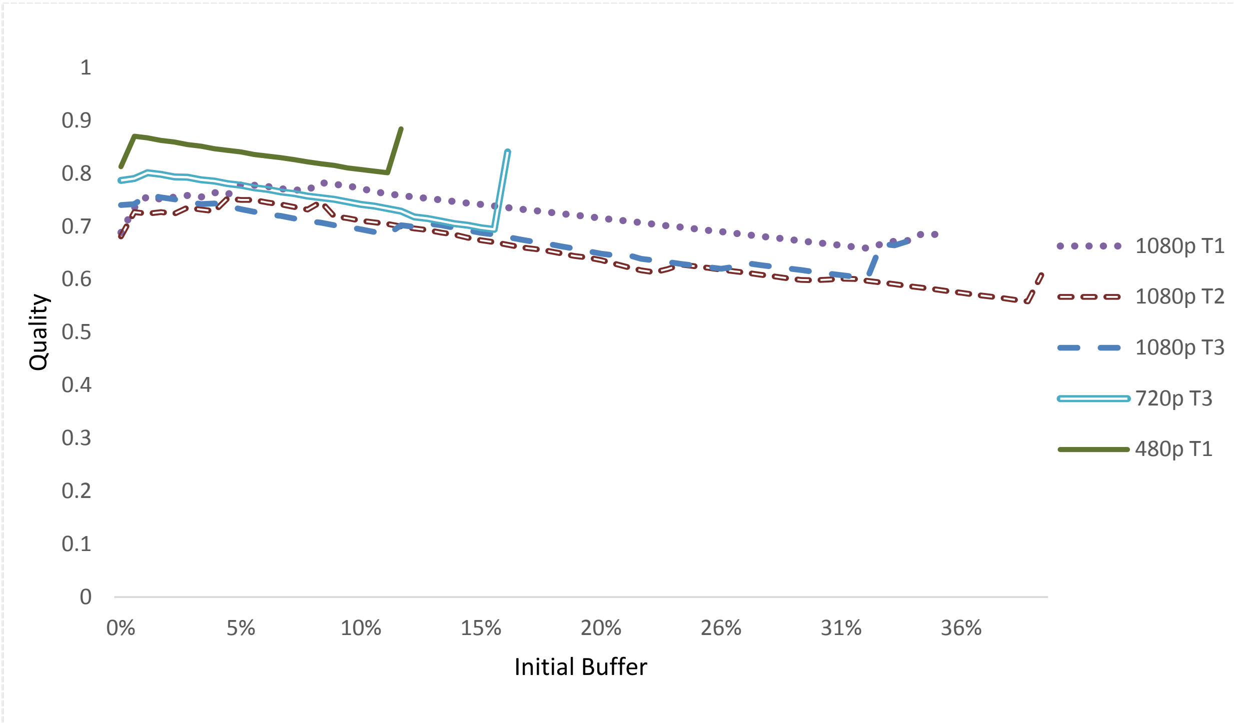

Figure 29: AK Quality vs Initial Buffer

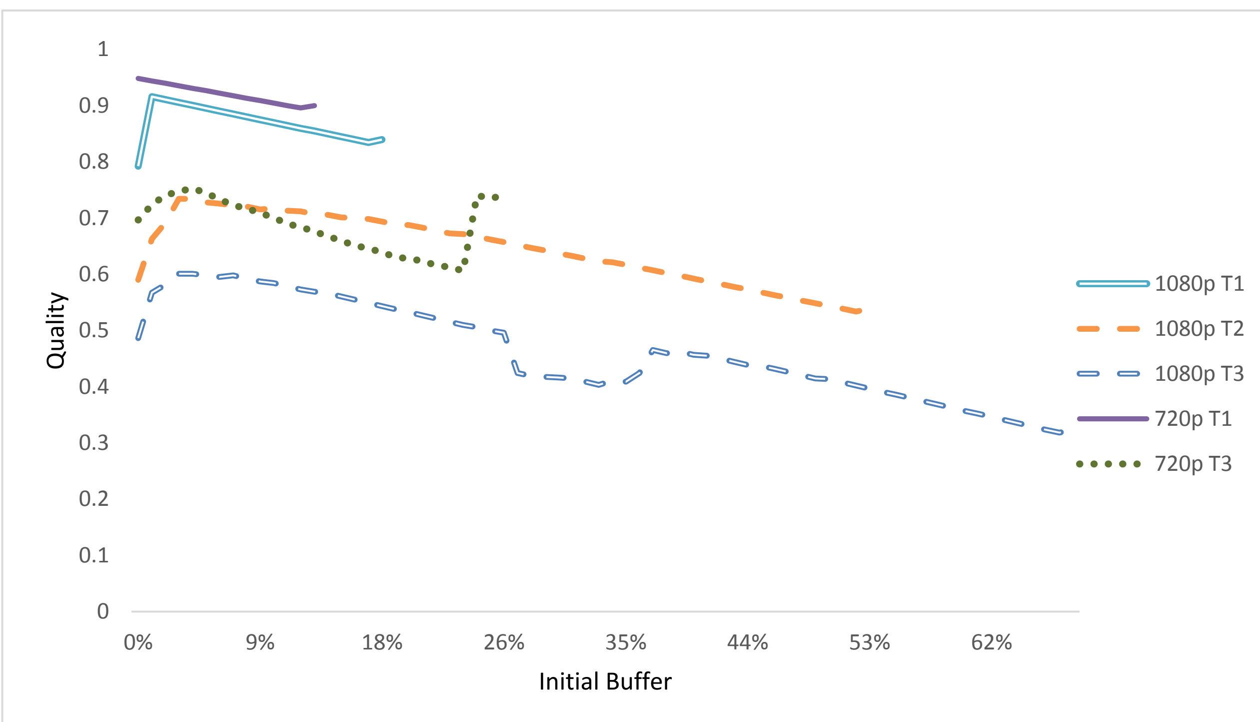

Figure 30: CC to Library Quality vs Initial Buffer

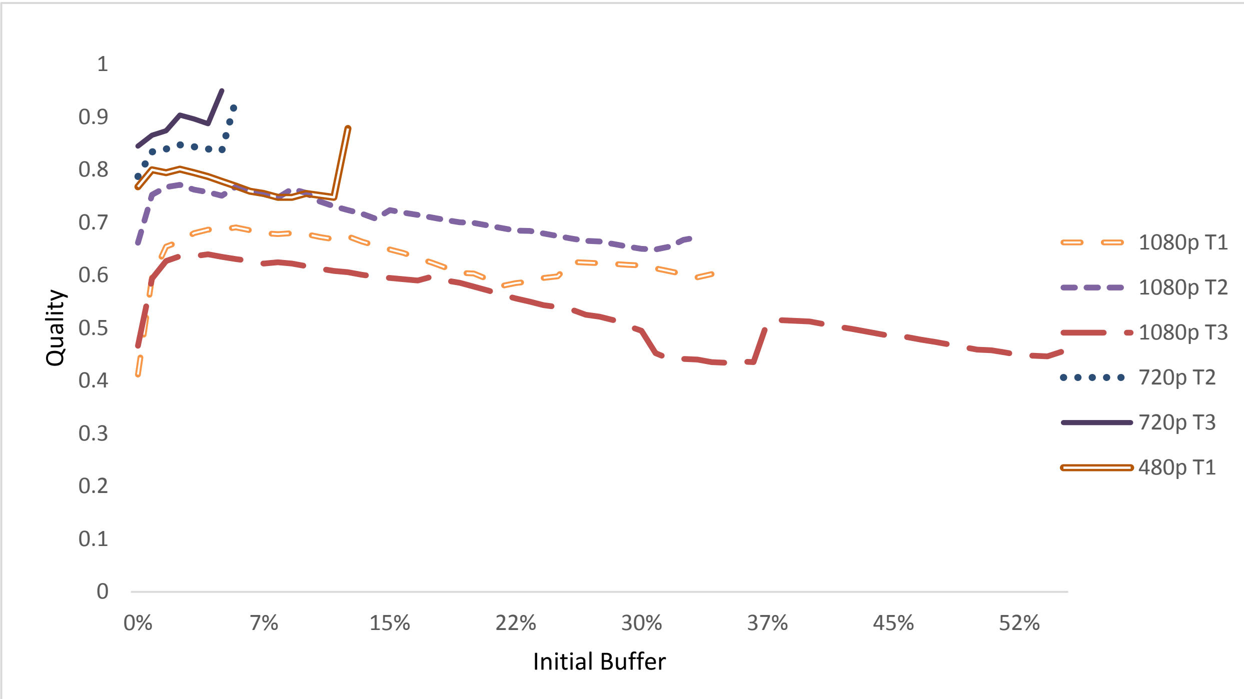

Figure 31: Founders Quality vs Initial Buffer

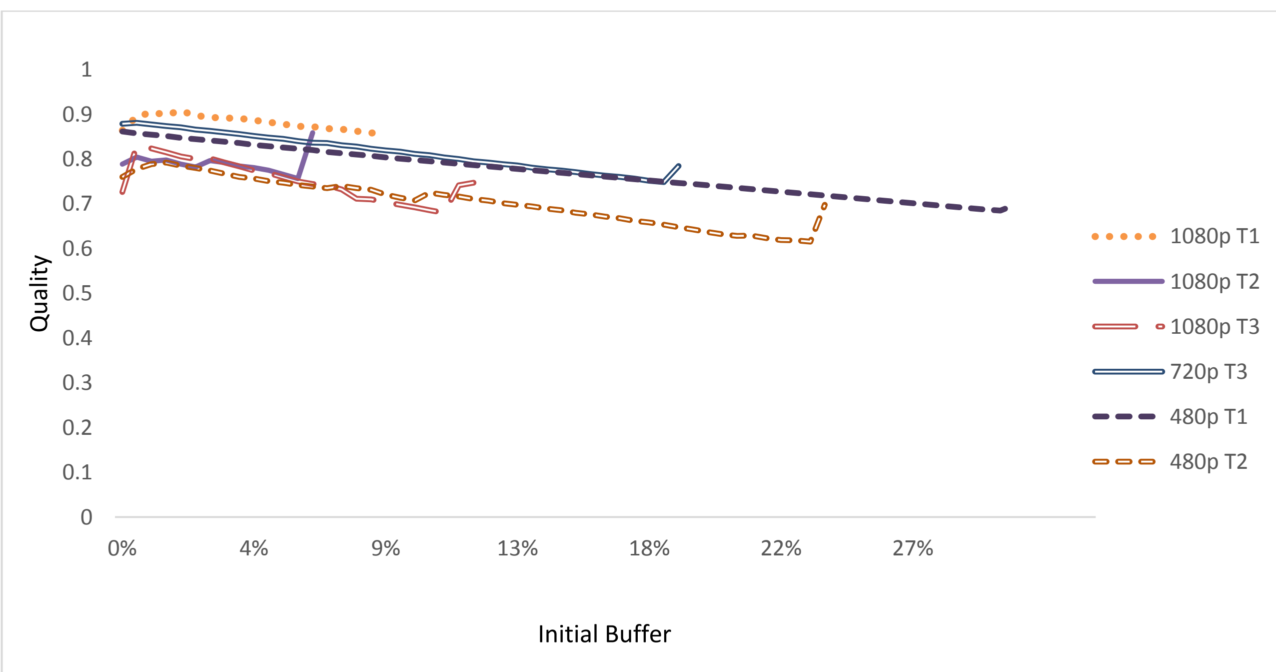

Figure 32: Morgan to CC Quality vs Initial Buffer

Table 6 is the summary of quality ratings obtained from the simulated data. For the 1080p video, average values across the three trials for each path are presented. For lower qualities, average values across all 4 paths are calculated. For the lowest quality, 360p, there were no interrupts and no initial buffer was necessary for the video to play out smoothly, thus achieving the highest quality rating of 1. The relationship between achievable quality rating and video resolution is observed to be inversely proportional, with the quality rating getting higher as resolution decreases. The initial buffer size varies more with resolution, as 720p requires a bigger initial buffer size than most of the 1080p videos, however, with a bigger buffer size, a higher quality rating is also achieved for 720p.

Path	Resolution	Peak Quality Rating	Optimal Initial Buffer Size (%)
AK	1080p	0.77	4.4
Morgan to CC	1080p	0.87	6.1
CC to Lib	1080p	0.71	2.7
Founders	1080p	0.70	4.4
ALL	1080p	0.76	4.4
ALL	720p	0.88	5.4
ALL	480p	0.95	2.1
ALL	360p	0.98	0.33

Table 6: Average Qualities and Initial Buffers

5 Conclusion

The recent surge in popularity of video streaming has encouraged exploration into optimizing user experience. In particular, our study concentrated on the effects of buffering on mobile users' quality of experience with YouTube, one of the largest video-hosting websites. Users connected to a Wi-Fi network often experience large deviations in signal quality based on factors including: network infrastructure, geographical conditions, network traffic, weather, movement speed, human traffic, locations of access points, and access point handoffs.

Our study was able to provide data and analysis towards better understanding video performance with emphasis on comparisons by both path and quality. The quality comparisons showed that the two lower video qualities (360p and 480p) generally can play through video without any interrupts using a college campus Wi-Fi network when using YouTube's current buffer policy. Based on our simulated results, video played at 360p and 480p can be assigned initial buffers of 0% and 2.1% of the total video length respectively in order to achieve maximum quality of experience for the user. Both 360p and 480p had nearly perfect quality values, with 360p having a quality of 1.0 and 480p a quality of .95. The 720p trials were much less consistent than the other qualities as they generally had some interrupts. On the other hand the 1080p trials were consistently the poorest performers in terms of quality of user experience. The 1080p trials had the highest frequency of interrupts for every path. According to our simulated results, 1080p had a much lower peak quality value than 720p (.76 and .88 respectively); however the 1080p trials reached this peak with a smaller initial buffer size (4.4% and 5.4% respectively).

The variation between not only paths but even trials was so great that there were no consistent characteristics which could be used to define indoor vs outdoor paths. However for a given path, the total interrupted time remains relatively constant between trials while the frequency and average length of interrupts varies. This observation only holds true for video played at 1080p. This suggests that interrupted time could be the most revealing characteristic of a path in terms of how well video can be streamed along that path.

6 Future Work

This chapter looks at our study as a foundation for future exploration into the video buffering problem space. There are many variables when it comes to streaming video over wireless networks. Our study kept most variables constant. Future studies in this area could change the streaming service, mobile device, type of wireless network, or any of the other parameters.

Our study looked specifically at YouTube, but there are other streaming sites that could be observed. Two of the biggest are Netflix and Hulu, which primarily stream TV shows and movies. Using the same procedure we used to analyze YouTube, these other video streaming sites could also be analyzed.

Another variable in this study was the type of wireless network. We only used Wi-Fi when streaming video, but there are other ways to transmit data wirelessly. One popular method is 3G and 4G. It would be interesting to compare the differences between streaming video over Wi-Fi and streaming over 3G/4G.

Smartphones and tablets are ubiquitous, and they are often capable of both Wi-Fi and 3G/4G connections. A study could look at video streaming on these devices rather than laptops. We used laptops in our study primarily due to the availability of software such as Wireshark that let us observe network traffic. Research into video streaming on smartphones and tablets would be valuable considering their prevalence in modern-day society.

7 WORKS CITED

[1] A. Lella, "comScore Releases December 2013 U.S. Online Video Rankings," 10 Janurary 2014. [Online]. Available: https://www.comscore.com/Insights/Press_Releases/2014/1/comScore_Releases_December_2013_US_Online_Video_Rankings.

[2] G. Abramovich, "15 Stats Brands Should Know About Online Video," 3 April 2013. [Online]. Available: http://digiday.com/brands/celtra-15-must-know-stats-for-online-video/.

[3] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018," 5 February 2014. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html.

[4] YouTube, "Statistics," 1 March 2014. [Online]. Available: http://www.youtube.com/yt/press/statistics.html.

[5] C. Smith, "By the Numbers: 32 Amazing YouTube Statistics," 24 February 2014. [Online]. Available: http://expandedramblings.com/index.php/youtube-statistics/#.UxI88LSomcw.

[6] P. Gill, M. Arlitt, Z. Li and A. Mahanti, "Youtube traffic characterization: a view from the edge," ACM, New York, 2007.

[7] M. Zink, K. Suh, Y. Gu and J. Kurose, "Watch global, cache local: YouTube network traffic at a campus network: measurements and implications," SPIE, San Jose, 2008.

[8] H. Riiser, H. S. Bergsaker, P. Vigmostad, P. Halvorsen and C. Griwodz, "A comparison of quality scheduling in commercial adaptive HTTP streaming solutions on a 3G network," ACM, New York, 2012.

[9] F. Dobrian, A. Awan, D. Joseph, A. Ganjam, J. Zhan, V. Sekar, I. Stoica and H. Zhang, "Understanding the impact of video quality on user engagement," 2011.

[10] D. Wang, F. Speranza, A. Vincent, T. Martin and P. Blanchfield, "Toward optimal rate control: a study of the impact of spatial resolution, frame rate, and quantization on subjective video quality and bit rate," SPIE, Lugano, 2003.

[11] T. Hoßfeld, F. Liers, T. Volkert and R. Schatz, "FoG and Clouds: Optimizing QoE for YouTube," 2011.

[12] T. Henderson, D. Kotz and I. Abyzov, "The changing usage of a mature campus-wide wireless network," ACM, New York, 2004.

[13] S. Alcock and R. Nelson, "Application flow control in YouTube video streams," ACM, New York, 2011.

[14] S. Mastoureshgh, "Measurement and Method for Receiver Buffer Sizing in Video Streaming," Worcester Polytechnic Institute, Worcester, 2012.

[15] "Wireshark: Go Deep," [Online]. Available: http://www.wireshark.org/. [Accessed 17 October 2013].

[16] "WirelessMon by Passmark Software," Passmark Software, [Online]. Available: http://www.wirelessmon.com/. [Accessed 20 February 2014].

[17] "Matlab: The Language of Technical Computing," The MathWorks, Inc., [Online]. Available: http://www.mathworks.com/products/matlab/. [Accessed 17 October 2013].

[18] X. Cheng, C. Dale and J. Liu, "Statistics and Social Network of YouTube Videos," IEEE, Enschede, 2008.

[19] "Keepvid: Download and save any video from Youtube, Google, Metacafe, Putfile and more," [Online]. Available: http://keepvid.com/. [Accessed 18 October 2013].

[20] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2012–2017," 2013.

[21] E. Protalinski, "Sandvine: Netflix owns one-third of North American traffic at peak, has doubled its mobile share in 12 months," thenextweb.com, May 2013. [Online]. Available: http://thenextweb.com/insider/2013/05/14/sandvine-netflix-owns-one-third-of-north-

american-traffic-at-peak-has-doubled-its-mobile-share-in-12-months/. [Accessed 18 October 2013].