ENSC 894 Special Topics II: Communication Networks Final Project

Performance Analysis of YouTube streaming with WiFi

Project website

http://www.sfu.ca/~hdhondea/ENSC894Group2.html

Group 2

Prepared by:

Amandeep Kaur (Aman)

301394838

aka148@sfu.ca

Haotian Ye (Tian)

301226346

haotiany@sfu.ca

Ashiv Rao Dhondea (Hans)

301400489

hdhondea@sfu.ca

Spring 2020

Project prepared for:

Prof. Ljiljana Trajković

&

Mr. Zhida Li

April 17, 2020

Abstract

Video streaming is quickly becoming the most common use case for Internet traffic globally. The dominant real-time entertainment service supplier, YouTube, with 23.4% of the daily traffic in North America [1], employs HTTP adaptive streaming, DASH. This is made possible with the ever-increasing Quality of Service (QoS) and bandwidth capabilities of today's Internet. Technologies used by hosts for video streaming include Ethernet, WiFi and LTE (Long Term Evolution). The popularity of Ethernet is waning as more and more people make use of mobile devices and laptops which do not possess Ethernet ports. Video streaming over LTE is gaining traction in North America as people opt to do their YouTube or Netflix streaming while commuting to work or traveling. While LTE poses interesting challenges to video streaming, it was not investigated in this report because our version of *Riverbed Modeler*, the *Academic Edition version 17.5*, does not allow the use of LTE technology. WiFi is now available on university campuses, schools, coffee shops, shopping malls, restaurants and even on public transit in some countries. It has become ubiquitous and it is therefore a good choice of technology to investigate in this report.

We make use of *Riverbed Modeler* to simulate various scenarios and record useful statistics such as throughput and packet delay to see how WiFi performs for video streaming. The platform chosen is YouTube because it is the dominant entertainment service supplier. We show results of simulations in which the video display resolutions was varied from 7200p (720 pixels, progressive scan) to 1080p.

Acknowledgments

We would like to thank Professor Ljiljana Trajković for her expert lectures on Communication Networks.
We would like to thank the TA, Mr. Zhida Li, for giving us prompt guidance and valuable feedback whenever we requested it.

Contents

List of Figures

List of Abbreviations

ADSL Asymmetric Digital Subscriber Line. 1

DASH Dynamic Adaptive Streaming over HTTP.
An adaptive bitrate streaming technique that permits users to stream media content in high quality. i, 2, 3, 5, 37, 38

FTTH Fiber-To-The-Home. 1

HTTP Hypertext Transfer Protocol. i, vii, 3, 9, 10, 13, 15, 16, 19–21

IEC International Electrotechnical Commission. 3

IEEE 802.11 IEEE 802.11 standards concern protocols regulating WLAN communications. These standards are developed by the IEEE 802.11 Working Group. viii, 4, 6, 10–12, 24–34, 36, 37

IPv6 Internet Protocol version 6. 8

IP Internet Protocol. vii

ISO International Organization for Standardization. 3

LTE Long Term Evolution. i, 5, 38

MIMO Multiple Input Multiple Output. 4

MPEG Moving Picture Experts Group. 3

ns-3 Network Simulator 3.
A discrete-event network simulator. 5, 38

OPNET Optimized Network Engineering Tools.
OPNET is the predecessor to Riverbed Modeler. 5, 6

QoS Quality of Service. i, 2, 4, 5, 10, 31, 33, 36, 37

RTT Round Trip Time. 8, 9

TCP Transmission Control Protocol. 2, 9, 10, 19

URL Universal Resource Locator. 3

VoIP Voice over Internet Protocol. 6

WLAN Wireless Local Area Network. vii, 4, 6, 10–13, 15, 17, 24, 29–31, 38

WiFi Wireless Fidelity, a wireless networking technology encompassing IEEE 802.11 standards. i, 1, 2, 4–6, 9–11, 19, 23–25, 27–29, 31–33, 35–38

Chapter 1

Introduction

With 23.4% of the daily traffic in North America according to the 2018 *Sandvine* Internet Phenomena Report [1], YouTube is the dominant real-time entertainment service supplier. According to [2], YouTube's popularity has greatly increased over the years. Over a period of four years, YouTube's daily usage by its subscribers has grown from more than 25% in July 2013 to more than 45% in July 2017, as can be seen in the following figure.

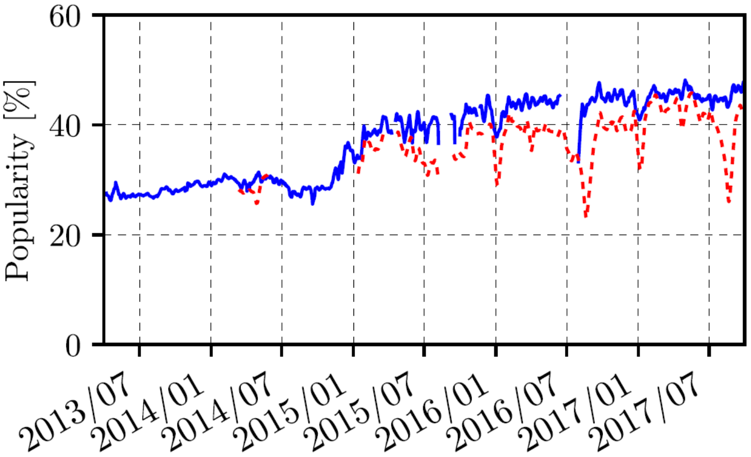


Figure 1-1: Popularity of YouTube video streaming from 2013/07 to 2017/07. The blue graph indicates ADSL traffic while the red graph represents FTTH traffic. Source: [2]

In Figure 1-1, ADSL means Asymmetric Digital Subscriber Line and FTTH means Fiber To The Home.

Given YouTube's growing popularity, it is the ideal platform to investigate in this project.

WiFi is now available on university campuses, schools, coffee shops, shopping malls, restaurants and even on public transit in some countries. It has become ubiquitous in Canada and it is therefore a good choice of technology to investigate in this report.

This project aims to investigate YouTube video streaming over WiFi under QoS parameters. The discrete network simulator *Riverbed Modeler* will be used to simulate various scenarios and record useful statistics such as throughput and packet delay to see how WiFi performs for video streaming.

1.1 Project objectives, scope and limitations

The main goal is to simulate YouTube video streaming over WiFi with a discrete network simulator. The scope of the project is to simulate network topologies featuring WiFi links and applications for YouTube video streaming using Riverbed Modeler. Various scenarios with different types of video streaming at distinct resolutions will be simulated. Phenomena relating to the wireless nature of the technology used will also be investigated, e.g. a client moving with respect to the WiFi access point.

The main limitations in this project are imposed by the network simulator used. If a desired feature or module is not available in the chosen network simulator, simulations requiring this feature or module will necessarily be absent in this report.

Before delving into the network topology and simulation scenarios used in this project in Chapter 2, we discuss the background to this project in the following section.

1.2 Background

This section provides the background to the project. Video streaming is introduced and discussed in Section 1.2.1. In particular, the specific protocol which YouTube uses, DASH is described in this section. This is followed by an overview of WiFi technologies employed in this project in Subsection 1.2.2. Subsection 1.2.3 briefly discusses how Quality of Service is measured in this project. The network simulator employed in this work is described in Subsection 1.2.4. A survey of the literature on video streaming simulations using network simulators is presented in Subsection 1.2.5. Finally, Subsection 1.2.6 provides a review of related projects done previously at SFU.

1.2.1 Video streaming

Video streaming allows end-users to play the video while the file contents are being downloaded. YouTube employs TCP (Transmission Control Protocol) in the transport layer for video streaming. [3] In the application layer, YouTube makes use of DASH, Dynamic Adaptive Streaming over HTTP [4]. Introduced in the open ISO/IEC 23009-1:2014 Standard [5], DASH is a technique which enables adaptive bitrate streaming over HTTP. The latest (revised) version of the Standard is ISO/IEC 23009-1:2019 [6]. YouTube uses a variant of DASH known as MPEG-DASH [4] (MPEG-DASH stands for Motion Picture Experts Group - Dynamic Adaptive Streaming over HTTP [7]). DASH entails encoding chunked media content (e.g. video or audio files) at multiple bit rates and storing them along with a *manifest file* on the server side. The manifest file contains a URL for each file chunk encoded at each bit rate, as can be seen in Figure 1-2. The chunks have a duration of 2 to 10 seconds [8].

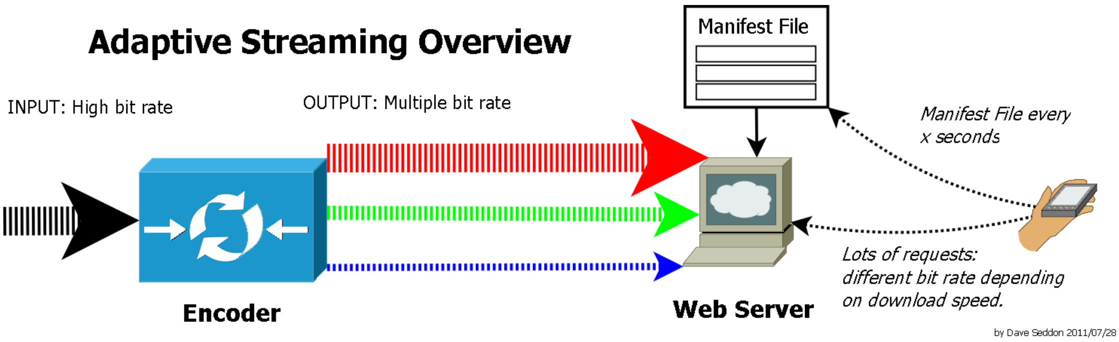


Figure 1-2: Adaptive Streaming Overview [9]. The encoder outputs streams at multiple bit rates which contain chunks of variable length. The client requests chunks from different bit streams according to the currently available bandwidth.

On the client side, the server-to-client bandwidth and the client's CPU capacity are measured regularly and the quality of the media stream is fine-tuned to the maximum coding rate which can suit these conditions. This implies that the client can choose different coding rates at various points in time according to the available bandwidth at that time, as illustrated in Figure 1-3 from [10].

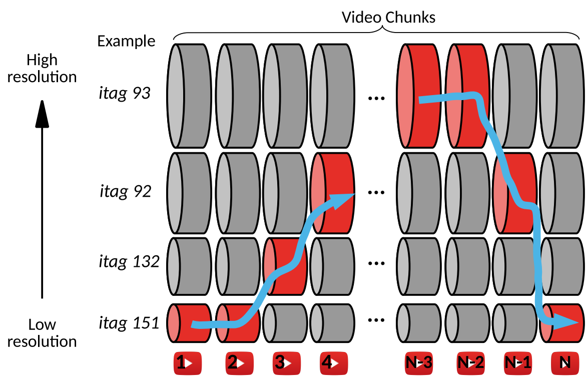


Figure 1-3: YouTube's streaming strategy [10]: over time, the streaming client receives chunks encoded at varying resolutions.

1.2.2 IEEE 802.11 technologies

IEEE 802.11 protocols are WLAN standards created and curated by the IEEE[1] 802.11 Working Group[2].

The first IEEE 802.11 standard considered in this work is 802.11a which has a maximum theoretical data rate of 54 Mbit/s. The frequency range is around 5 GHz [11] which is an unlicensed frequency band. Since the associated wavelength is of the order of 6 cm, it is readily absorbed by objects and walls in buildings. On the other hand, 802.11g WiFi which operates at 2.4 GHz is less readily absorbed. WiFi at 2.4 GHz is also in an unlicensed frequency band. Due to the fact that several devices also operate at 2.4 GHz such as microwave ovens, Bluetooth-enabled devices, cordless phones, baby monitors, car alarms and garage door openers [12], electromagnetic interference may occur if a WLAN operates in the 2.4 GHz frequency band. 802.11g also has a maximum theoretical data rate of 54 Mbit/s. [11] These technologies are effectively combined in 802.11n: it operates in both frequency ranges, 2.4 GHz and 5 GHz. 802.11n differs in the sense that it may be implemented in MIMO (Multiple Input, Multiple Output) systems with multiple transmitting antennas and different modulation schemes. [11] Furthermore, it has increased the maximum theoretical data rate to 72 Mbit/s. Modern devices such as smartphones are now being developed to be dual-band compatible so that they may operate in both frequency bands of 802.11n. Since fewer devices exploit the higher 5 GHz range, this means that a larger number of hosts may be accommodated by a 5 GHz WLAN.

1.2.3 Quality of Service

The term 'Quality of Service' (QoS) is used to quantify the overall performance of a network. It comprises of the following parameters [13]:

- Throughput
- Packet loss
- Latency
- Jitter

In this project, we mainly investigated throughput and packet delay (latency).

[1] Institute of Electrical and Electronic Engineers. https://www.ieee.org/
[2] http://www.ieee802.org/11/

1.2.4 Riverbed Modeler

Riverbed Modeler is a modeling and simulation environment which can be used to simulate computer networking scenarios. [14]. It is a proprietary software whose *Academic Edition* used in this course provides a restricted set of functionalities. *Riverbed Modeler* provides a Graphical User Interface which permits the user to create simulations quickly, without having to develop simulation scripts or coding libraries, unlike the open-source discrete event simulator ns-3 [15]. Due to the time-constrained nature of this project, we opted for *Riverbed Modeler Academic Edition version 17.5* over ns-3.

However, the Academic Edition has some limitations [16] which restrict the simulations envisaged in this project:

- The Academic Edition software does not allow the user to import capture files/trace files discrete network simulations.
  This is an issue since we envisaged collecting trace files of video streaming over WiFi and LTE and passing these into our Riverbed Modeler simulations.
- The Academic Edition software does not support System-In-The-Loop simulations.
  We envisaged using our computers connected to WiFi networks as systems-in-the-loop in simulations.

1.2.5 Video streaming simulations with network simulators

In 2014, Hassan *et al.* reported on real-time video streaming experiments done with the System-In-The-Loop (SITL) module of OPNET (the predecessor to Riverbed Modeler) using WiMax in [17]. Employing DASH, they used video segments of varying duration to investigate the influence on bandwidth utilization and CPU resource utilization at the streaming client.

In 2017, using OPNET, Mohamed and Ibrahim simulated video streaming over LTE featuring • co-channel interference between adjacent cells • jamming by a jammer node to cancel co-channel interference in [18] QoS parameters such as throughput, jitter and end-to-end delay were recorded.

1.2.6 Related work at SFU

A number of previous projects in this course also focused on video streaming over WiFi. They were informative in our investigations.

- *Video Streaming over the 802.11g WLAN Technologies*, Spring 2011. [19]
  Xue made use of OPNET (the original incarnation of *Riverbed Modeler*) to investigate three cases in [19]:

  – a IEEE 802.11g WLAN simulation with various data rates.

  – a IEEE 802.11g WLAN simulation with a faster server.

  – a IEEE 802.11g WLAN simulation with a server with more powerful transmit power.

  Metrics such as packet end-to-end delay and throughput were collected and analyzed [19].

- *Performance Analysis of a Wireless Home Network* [20]
  Calzada *et al.* used the videoconferencing application of OPNET to simulate video streaming, among several other scenarios, such as web browsing and VoIP (Voice over Internet Protocol). The delay resulting from these applications were compared in [20].

- *Performance Analysis of Video Streaming over WiFi and Ethernet*, Spring 2015. [21]
  Singh and Labayo in [21] used *Riverbed Modeler* to run the following scenarios:

  – Ethernet with a single host (workstation).

  – WLAN with a single host.

  – Both Ethernet and WLAN with a single host.

  – Both Ethernet and WLAN with two hosts.

  The video conferencing module of *Riverbed Modeler* was used to simulate video streaming [21].

- *Video Streaming over WiFi*, Spring 2015. [22]
  Kim *et al.* simulated a home network and ran the following scenarios in *Riverbed Modeler*: light browsing, heavy browsing, VoIP (Voice over Internet Protocol), and video conferencing of movies' traces to simulate video streaming. [22] The simulations were adapted to investigate changing data rate, changing the protocol from IEEE 802.11a to IEEE 802.11g to IEEE 802.11n and finally, varying the distance between the WiFi access point and the host. [22]

- *Video Streaming over WiFi using Riverbed Modeler*, Spring 2016. [23]
  Ng and Weng also simulated a home network and ran two types of scenarios:

  – Light browsing, heavy browsing and video streaming.

  – Streaming two movies and varying attributes, such as data rate, to the server.

  Metrics such as throughput, delay and jitter were collected and analyzed. [23]

1.3 Report overview

This report is structured as follows:

- Chapter 2 describes in detail the design and execution of the simulation experiments conducted in this project. Based on the foregoing discussion in the literature review in Section 1.2.6, experiments were designed to best accomplish the objectives of this project using the software tool chosen. Results from the simulations are shown and discussed in depth.
- Chapter 3 reviews the experiments in the previous chapter. It outlines how the goals of this project were achieved. Furthermore, Chapter 3 puts forth our recommendations for future work.
- Appendix A provides important information which will help the reader to replicate our experiments and to re-create our report from the source files.

Chapter 2

Simulation design & Results

2.1 Introduction

This chapter is concerned with the overall design and execution of the experiments conducted in this project. It begins with basic investigations to gain an understanding of YouTube video streaming. Section 2.2 shows results when pinging YouTube. These results are informative in running a data collection experiment with *Wireshark* in Section 2.3. In light of the information collected in the foregoing sections, Section 2.4 describes the Riverbed Modeler scenarios implemented in this project and presents the simulation results. Finally, Section 2.5 concludes this chapter.

2.2 Pinging YouTube

The URL youtube.com was pinged to have a realistic idea of the Round Trip Times (RTTs). The IPv6 address for the YouTube server is `2607:f8b0:400a:801::200e` as can be seen in the screenshot in Figure 2-1.

```
C:\Users\haoti>ping -n 10 www.youtube.com

Pinging youtube-ui.l.google.com [2607:f8b0:400a:801::200e] with 32 bytes of data:
Reply from 2607:f8b0:400a:801::200e: time=7ms
Reply from 2607:f8b0:400a:801::200e: time=14ms
Reply from 2607:f8b0:400a:801::200e: time=32ms
Reply from 2607:f8b0:400a:801::200e: time=8ms
Reply from 2607:f8b0:400a:801::200e: time=27ms
Reply from 2607:f8b0:400a:801::200e: time=23ms
Reply from 2607:f8b0:400a:801::200e: time=8ms
Reply from 2607:f8b0:400a:801::200e: time=21ms
Reply from 2607:f8b0:400a:801::200e: time=26ms
Reply from 2607:f8b0:400a:801::200e: time=7ms

Ping statistics for 2607:f8b0:400a:801::200e:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 32ms, Average = 17ms
```

Figure 2-1: Windows command window: pinging YouTube

The RTT values ranged from 7 ms to 32 ms with an average of 17 ms. A *whois* domain lookup done on www.arin.net/whois shows that the owner of this server is *Google LLC*. This is expected since Google is the owner of YouTube.

2.3 Wireshark data collection

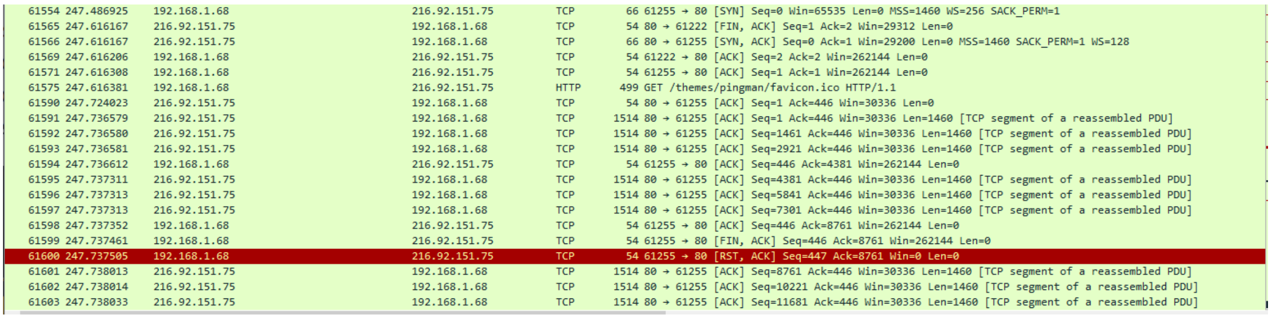
Wireshark was used to sniff the packets while streaming a 1080p YouTube video using a WiFi connection. This was instructive in understanding how YouTube implements video streaming. Since YouTube is known to be an HTTP application, port 80 was used to filter the data traffic during packet sniffing.

Figure 2-2 shows the Wireshark trace recorded when a YouTube video was streamed.

55267	153.897340	192.168.1.68	216.92.151.75	TCP	66	61194 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
55361	153.991186	216.92.151.75	192.168.1.68	TCP	66	80 → 61194 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
55366	153.991304	192.168.1.68	216.92.151.75	HTTP	499	GET /themes/pingman/favicon.ico HTTP/1.1
55452	154.092162	216.92.151.75	192.168.1.68	TCP	54	80 → 61194 [ACK] Seq=1 Ack=446 Win=30336 Len=0
55454	154.094045	216.92.151.75	192.168.1.68	TCP	1514	80 → 61194 [ACK] Seq=1 Ack=446 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
55455	154.094059	192.168.1.68	216.92.151.75	TCP	54	61194 → 80 [ACK] Seq=446 Ack=1461 Win=262144 Len=0
55457	154.095004	192.168.1.68	216.92.151.75	TCP	54	61194 → 80 [FIN, ACK] Seq=446 Ack=1461 Win=262144 Len=0
55458	154.095056	192.168.1.68	216.92.151.75	TCP	54	61194 → 80 [RST, ACK] Seq=447 Ack=1461 Win=0 Len=0
55459	154.105758	216.92.151.75	192.168.1.68	TCP	1514	80 → 61194 [ACK] Seq=1461 Ack=446 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
55460	154.105760	216.92.151.75	192.168.1.68	TCP	1514	80 → 61194 [ACK] Seq=2921 Ack=446 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
55461	154.107029	216.92.151.75	192.168.1.68	TCP	1514	80 → 61194 [ACK] Seq=4381 Ack=446 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
55462	154.107031	216.92.151.75	192.168.1.68	TCP	1514	80 → 61194 [ACK] Seq=5841 Ack=446 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
55463	154.108320	216.92.151.75	192.168.1.68	TCP	1514	80 → 61194 [ACK] Seq=7301 Ack=446 Win=30336 Len=1460 [TCP segment of a reassembled PDU]

Figure 2-2: Wireshark trace of TCP connection with server

Once the user starts playing the video, the socket from their computer tries to make a TCP connection with the socket port number 80 (which belongs to YouTube's server side socket). After the connection is established, the client's computer sends an HTTP Get message to the YouTube server. This is followed by the server transferring streaming packets to the client's computer.



61554	247.486925	192.168.1.68	216.92.151.75	TCP	66	61255 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
61565	247.616167	216.92.151.75	192.168.1.68	TCP	54	80 → 61222 [FIN, ACK] Seq=1 Ack=2 Win=29312 Len=0
61566	247.616167	216.92.151.75	192.168.1.68	TCP	66	80 → 61255 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
61569	247.616206	192.168.1.68	216.92.151.75	TCP	54	61222 → 80 [ACK] Seq=2 Ack=2 Win=262144 Len=0
61571	247.616308	192.168.1.68	216.92.151.75	TCP	54	61255 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
61575	247.616381	192.168.1.68	216.92.151.75	HTTP	499	GET /themes/pingman/favicon.ico HTTP/1.1
61590	247.724023	216.92.151.75	192.168.1.68	TCP	54	80 → 61255 [ACK] Seq=1 Ack=446 Win=30336 Len=0
61591	247.736579	216.92.151.75	192.168.1.68	TCP	1514	80 → 61255 [ACK] Seq=1 Ack=446 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
61592	247.736580	216.92.151.75	192.168.1.68	TCP	1514	80 → 61255 [ACK] Seq=1461 Ack=446 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
61593	247.736581	216.92.151.75	192.168.1.68	TCP	1514	80 → 61255 [ACK] Seq=2921 Ack=446 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
61594	247.736612	192.168.1.68	216.92.151.75	TCP	54	61255 → 80 [ACK] Seq=446 Ack=4381 Win=262144 Len=0
61595	247.737311	216.92.151.75	192.168.1.68	TCP	1514	80 → 61255 [ACK] Seq=4381 Ack=446 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
61596	247.737313	216.92.151.75	192.168.1.68	TCP	1514	80 → 61255 [ACK] Seq=5841 Ack=446 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
61597	247.737313	216.92.151.75	192.168.1.68	TCP	1514	80 → 61255 [ACK] Seq=7301 Ack=446 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
61598	247.737352	192.168.1.68	216.92.151.75	TCP	54	61255 → 80 [ACK] Seq=446 Ack=8761 Win=262144 Len=0
61599	247.737461	192.168.1.68	216.92.151.75	TCP	54	61255 → 80 [FIN, ACK] Seq=446 Ack=8761 Win=262144 Len=0
61600	247.737505	192.168.1.68	216.92.151.75	TCP	54	61255 → 80 [RST, ACK] Seq=447 Ack=8761 Win=0 Len=0
61601	247.738013	216.92.151.75	192.168.1.68	TCP	1514	80 → 61255 [ACK] Seq=8761 Ack=446 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
61602	247.738014	216.92.151.75	192.168.1.68	TCP	1514	80 → 61255 [ACK] Seq=10221 Ack=446 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
61603	247.738033	216.92.151.75	192.168.1.68	TCP	1514	80 → 61255 [ACK] Seq=11681 Ack=446 Win=30336 Len=1460 [TCP segment of a reassembled PDU]

Figure 2-3: Wireshark trace of TCP connection with server

As can be seen from Figure 2-3, YouTube, as an HTTP application, provides reliable video streaming through using TCP as transport protocol because of its reliable data transfer characteristic. The TCP protocol allows the receiver to buffer the data packets sent from the streaming server.

2.4 Riverbed Modeler simulations

YouTube streaming sessions usually last for one hour. Therefore we set the duration of our Riverbed Modeler simulations to one hour. However, as a consequence of the long simulation duration, we were not allowed on include additional work stations in the WLANs we simulated. We opted for a longer simulation duration over supporting additional users to ensure that our results are more comparable to a real-life user streaming YouTube videos on WiFi in a residence.

This section presents simulations done in Riverbed Modeler. Four scenarios are developed in this work:

- Scenario 1 in Subsection 2.4.1 investigates how three different types of browsing profiles (namely light browsing, heavy browsing and video streaming) affect the QoS enjoyed by the WiFi client.
- Scenario 2 uses two video streaming resolutions to find how they affect the throughput and average delay experienced by a WiFi client.
- Scenario 4 in Subsection 2.4.3 investigates three aspects:
  - the impact of the WiFi technology (either of IEEE 802.11g and n) employed.
  - the effect of the chosen frequency band employed.
  - the effect of varying the data rate.
- Scenario 4 in Subsection 2.4.4 considers the effect of varying the range from the WiFi client to the access point.

These four scenarios were selected to explore the breadth of the experience of a single client browsing on the Internet and streaming YouTube videos in their residence using a WiFi network.

2.4.1 Scenario 1: Light browsing, heavy browsing & video streaming

2.4.1.1 Topology

The topology in scenario 1 makes use of wireless LAN (WLAN) workstation fixed nodes. The router used is a WLAN Ethernet router which is connected to the server by an 10000BaseX Ethernet link. The distance between the router and application nodes are around ten to fifteen meters. The three application nodes are equidistant from the router. The effect of the range between the router and server is not considered during this simulation. Figure 2-4 shows the topology implemented in scenario 1.

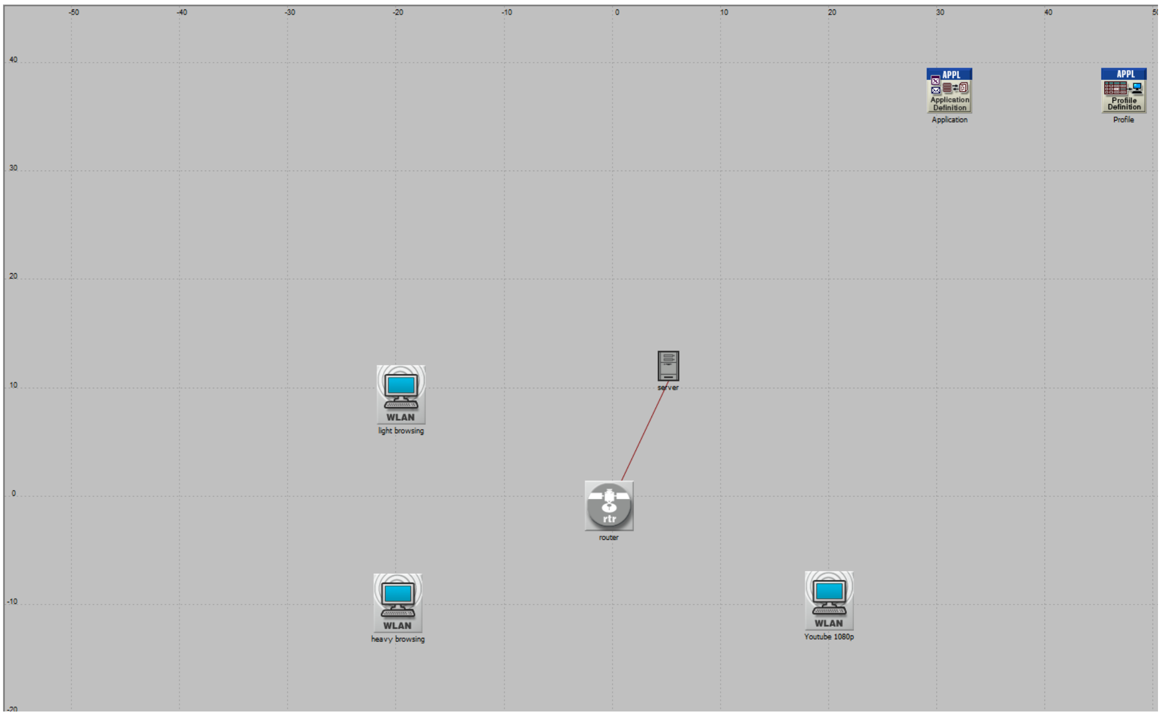


Figure 2-4: Scenario 1: topology

The BSS Identifier is set to one for both router and wireless LAN workstations so that these wireless LAN workstations can recognize that the router and workstations are in the same local area network. The physical characteristics and data rate are set to IEEE 802.11n 2.4 GHz and 65 Mbps (base)/600 Mbps for both router and wireless workstations in this scenario. We will implement the effect of physical characteristics and data rate in the later scenario.

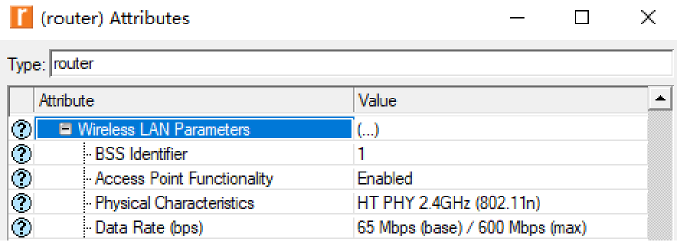


Figure 2-5: WLAN parameters of router

Figure 2-5 shows the wireless LAN parameters that we set for the router.

Setting the number of spatial streams to more than 1 and shorter guard intervals will result in higher physical data rate. To get a high physical data rate in this scenario, we set the number of spatial streams to 2 and the Guard Interval (GI) to a short time period of 400 ns under WLAN high throughput parameters, as can be seen in Figure 2-6. These two configurations lead to a physical data rate of 57.8 Mbps according to Table 20-30 in IEEE 802.11n-2009 [24]. We also enabled the Greenfield Operation attribute. This implies that WLAN nodes are allowed to a shorter physical layer header format and therefore resulting in higher throughput.

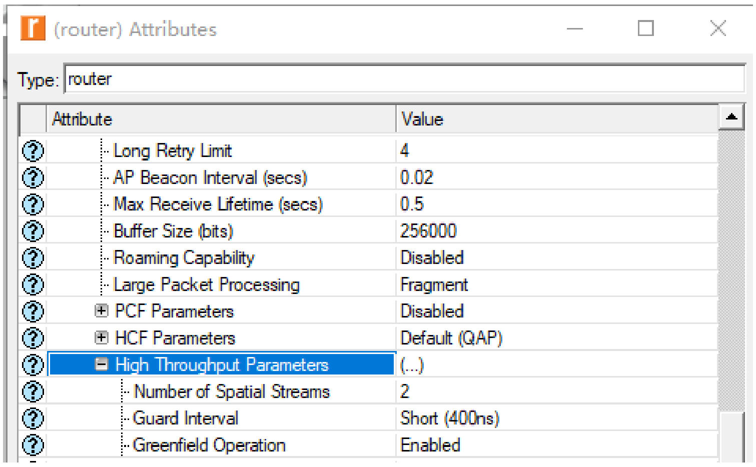


Figure 2-6: High throughput parameters of Local Area Network

2.4.1.2 Light browsing

The WLAN user is first simulated for light browsing, which consumes less system resources. We can see how the profile was configured in Figure 2-7, where we chose the object size as 10,000 bytes with uniform interval between 100 bytes and 4000 bytes. Also we set only one object to be contained in a page with normal distribution having mean of 10 objects and variance 5 objects.

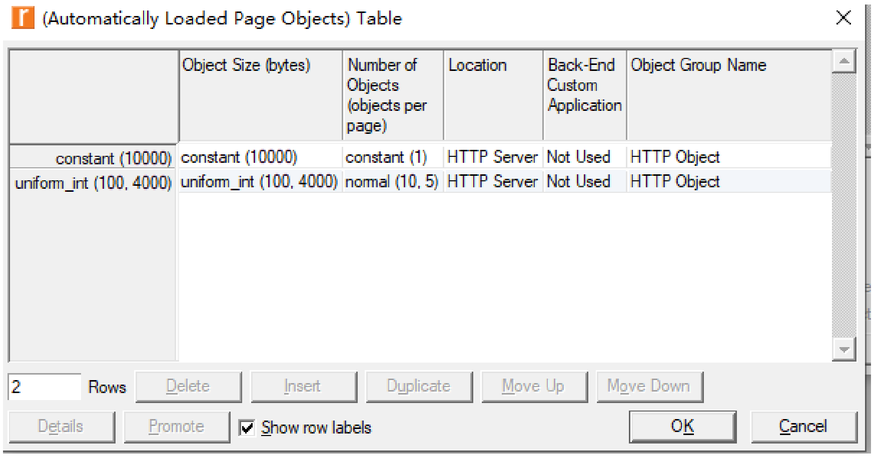


Figure 2-7: Light browsing scenario: setting object size

Figure 2-8 shows the HTTP table for light browsing, which shows that the page inter-arrival time is exponentially distributed with a rate parameter of 720 s by default.

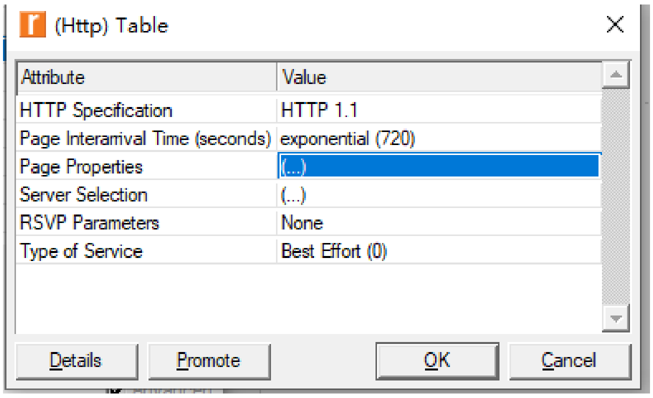


Figure 2-8: Light browsing scenario: HTTP application table

Figure 2-9 shows the profile setup employed for light browsing.

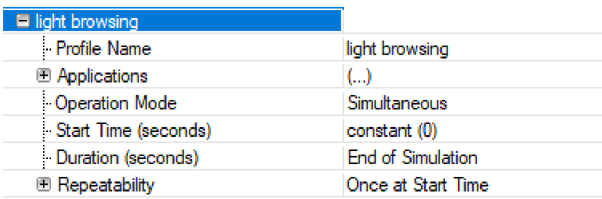


Figure 2-9: Light browsing scenario: profile setup

Figure 2-10 shows a plot of the WLAN throughput for this scenario.

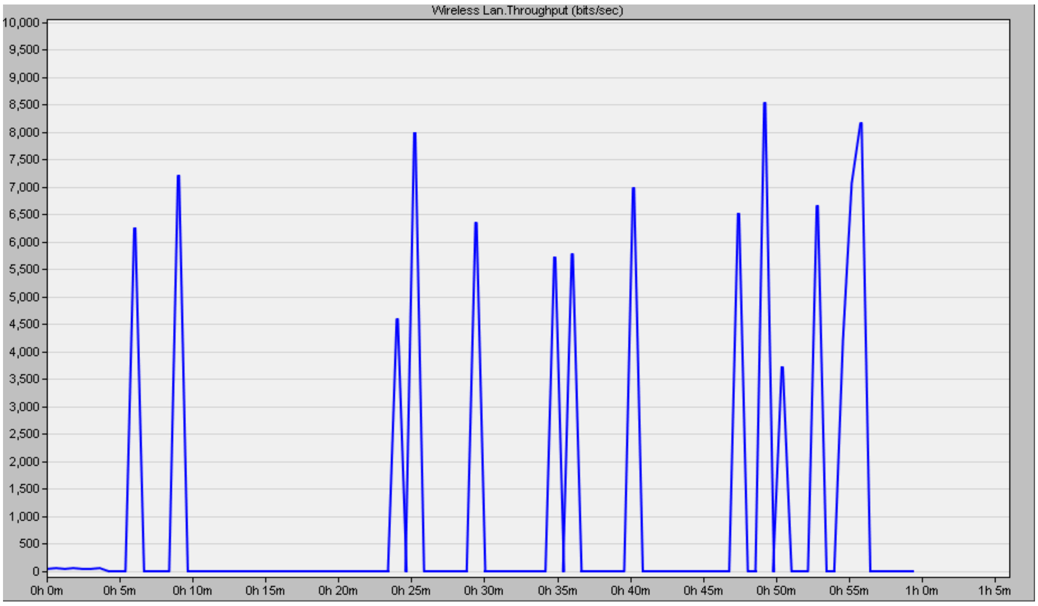
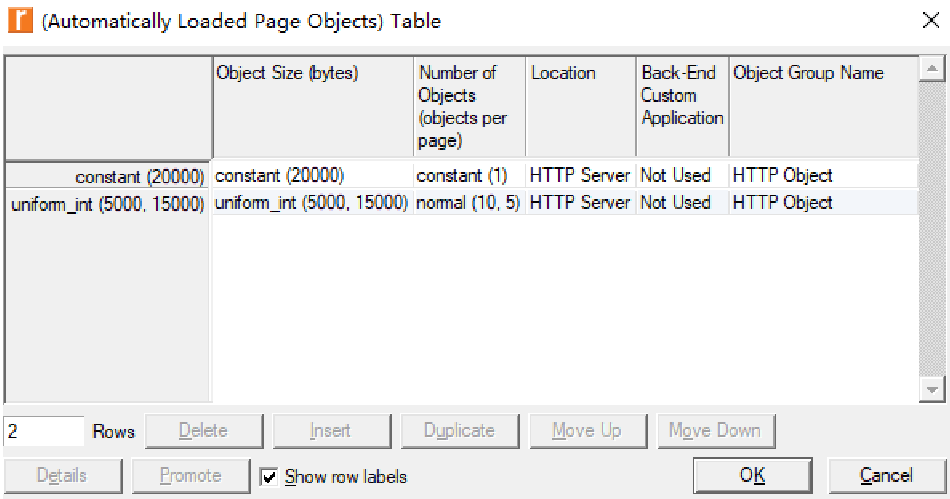


Figure 2-10: Light browsing scenario: throughput plot

2.4.1.3 Heavy browsing

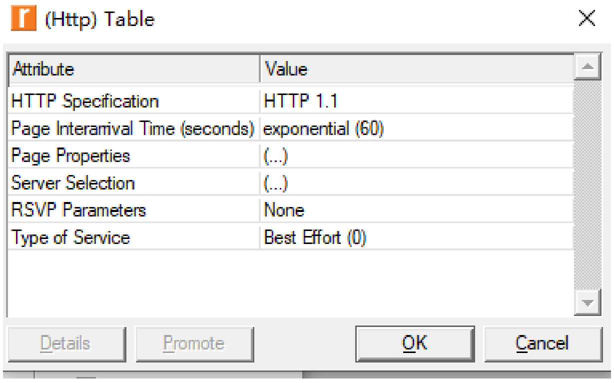
In the heavy browsing scenario, the object size was chosen to be larger (20,000 bytes versus 10,000 bytes in the light browsing scenario in Subsection 2.4.1.2) and the page inter-arrival time was decreased (more pages are browsed per unit time). As can be seen in Figure 2-11, the intervals are uniformly distributed between 5,000 bytes and 15,000 bytes. In a similar vein to the light browsing scenario, we set only one object to be contained in a page, according to a normal distribution with mean 10 objects and variance 5 objects.



	Object Size (bytes)	Number of Objects (objects per page)	Location	Back-End Custom Application	Object Group Name
constant (20000)	constant (20000)	constant (1)	HTTP Server	Not Used	HTTP Object
uniform_int (5000, 15000)	uniform_int (5000, 15000)	normal (10, 5)	HTTP Server	Not Used	HTTP Object

Figure 2-11: Heavy browsing scenario: setting object size

Figure 2-12 shows the application table for heavy browsing, which shows that the page inter-arrival time is exponentially distributed with a rate parameter of 60 seconds.

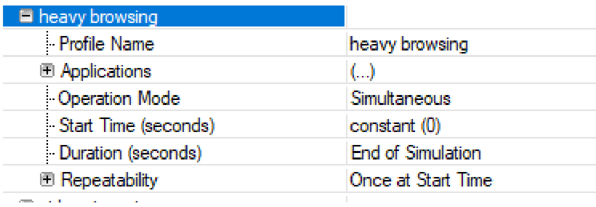

(Http) Table

Attribute	Value
HTTP Specification	HTTP 1.1
Page Interarrival Time (seconds)	exponential (60)
Page Properties	(...)
Server Selection	(...)
RSVP Parameters	None
Type of Service	Best Effort (0)

Details Promote OK Cancel

Figure 2-12: Heavy browsing scenario: HTTP application table

Figure 2-13 shows the profile setup employed for heavy browsing.


heavy browsing

Profile Name	heavy browsing
Applications	(...)
Operation Mode	Simultaneous
Start Time (seconds)	constant (0)
Duration (seconds)	End of Simulation
Repeatability	Once at Start Time

Figure 2-13: Heavy browsing scenario: profile setup

Figure 2-14 shows a plot of the WLAN throughput for this scenario.

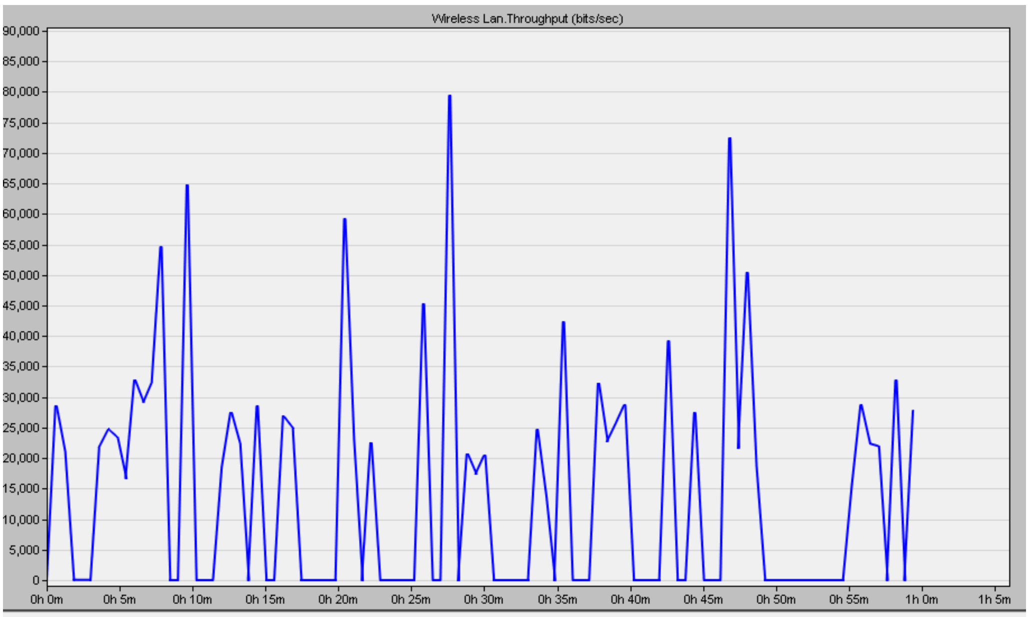


Figure 2-14: Light browsing scenario: throughput plot

2.4.1.4 Video streaming

In the video streaming scenario, the HTTP definition was changed to video browsing and the page inter-arrival time rate parameter was set to 360 s. The latter was chosen such that it is shorter than that (720 s) of the light browsing scenario in Subsection 2.4.1.2 and longer than that (60 s) of the heavy browsing scenario in Subsection 2.4.1.3.

For the short video, we set object size to 1000 bytes as can be seen in Figure 2-15. We can expect that the throughput of video streaming to be higher than light and heavy browsing. This is because video streaming requires higher data usage due to larger file sizes.

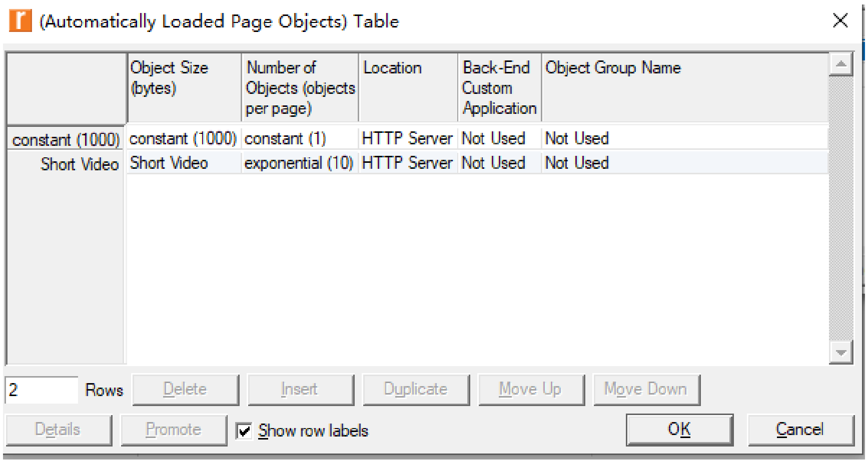


Figure 2-15: Video streaming scenario: setting object size

Figure 2-16 shows the application table for video streaming, which shows that the page inter-arrival time is exponentially distributed with a rate parameter of 360 seconds.

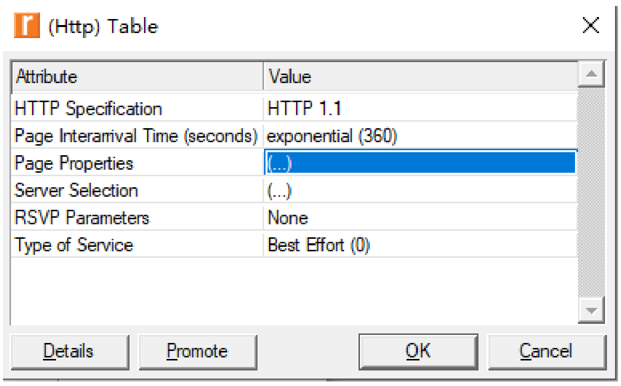
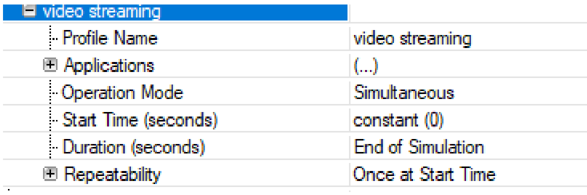


Figure 2-16: Video streaming scenario: HTTP application table

Figure 2-17 shows the profile setup employed for heavy browsing.



⊟ video streaming	
Profile Name	video streaming
⊞ Applications	(...)
Operation Mode	Simultaneous
Start Time (seconds)	constant (0)
Duration (seconds)	End of Simulation
⊞ Repeatability	Once at Start Time

Figure 2-17: Video streaming scenario: profile setup

Figure 2-18 shows a plot of the WLAN throughput for this scenario.

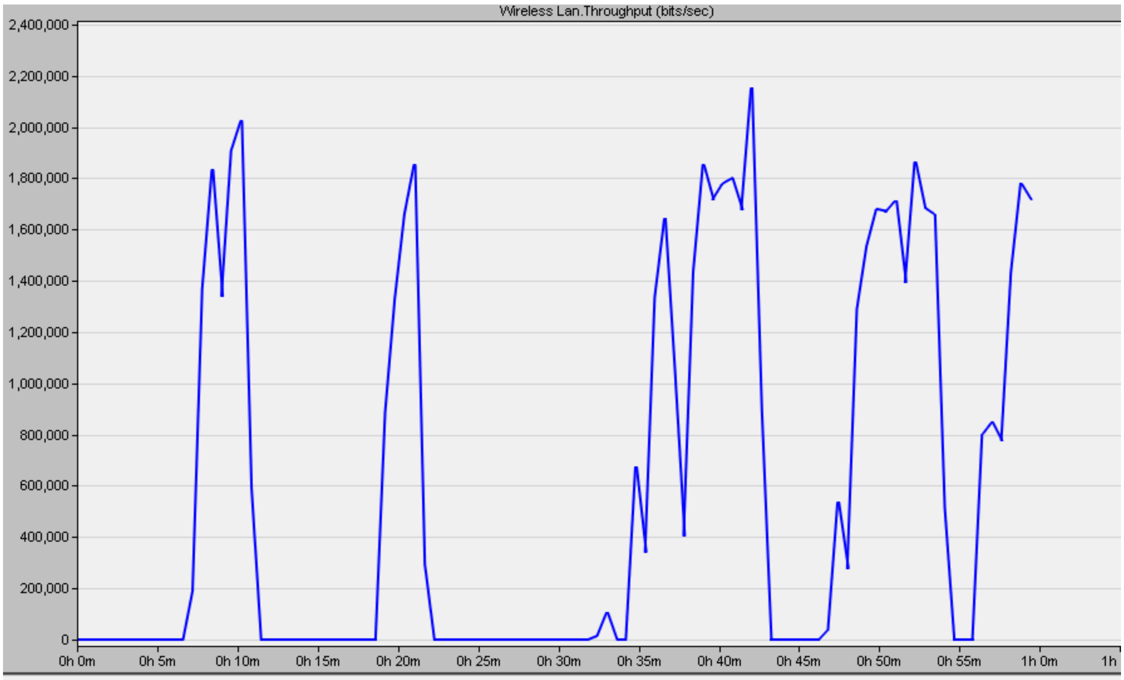


Figure 2-18: Video streaming scenario: throughput plot

2.4.1.5 Summary of Scenario 1

Based on the work done in Subsections 2.4.1.2, 2.4.1.3 and 2.4.1.4, we now summarize the results of scenario 1. As can be seen in Figure 2-19, the average throughput of heavy browsing is more than 8 times higher than that of light browsing. This result may seem intuitive and even though it is logical, its purpose is to serve as a sanity check in our work in this project.

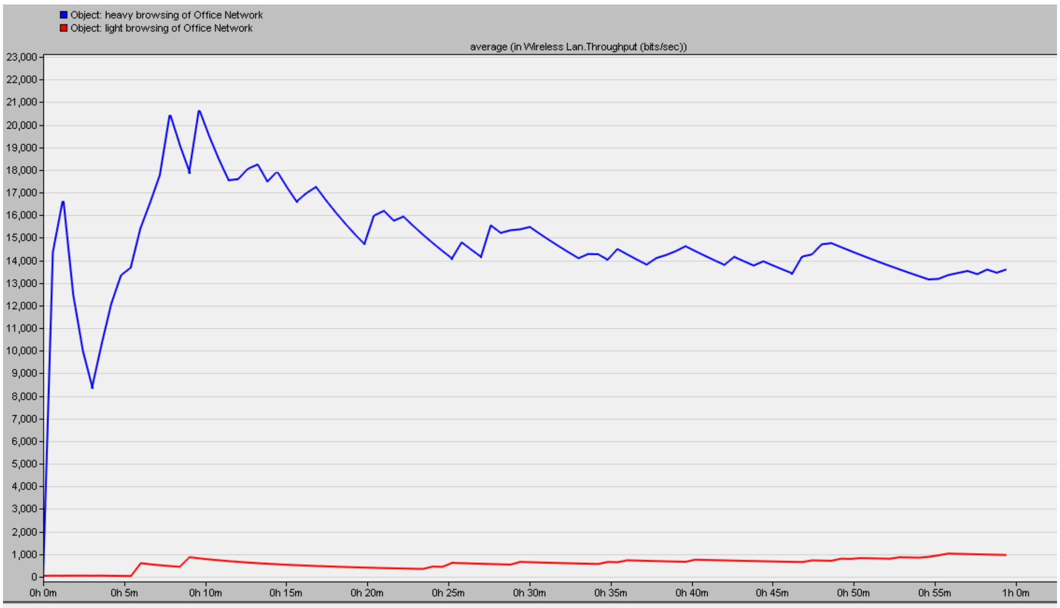


Figure 2-19: Comparison of average throughput of light browsing and heavy browsing. The blue graph represents heavy browsing while the red graph represents light browsing.

The reason for this significant difference is that heavy browsing has larger object size and shorter page inter-arrival time as compared to light browsing.

Furthermore, Figure 2-20 compares the average throughput of video streaming to that of the other two cases. This figure was included separately from Figure 2-19 because it does not show clearly the difference between the throughput of light browsing and heavy browsing.

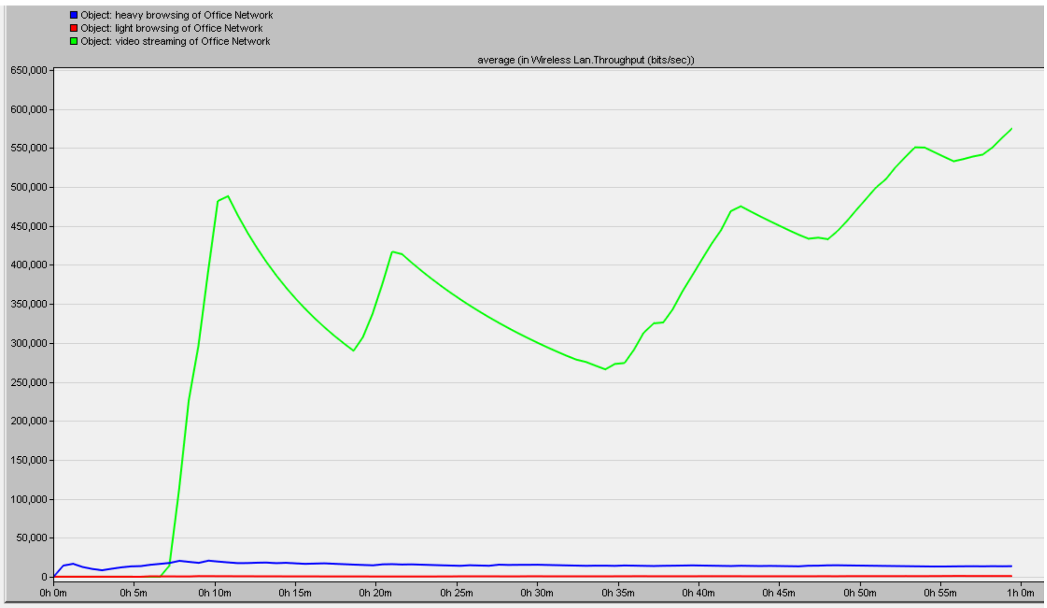


Figure 2-20: Comparison of average throughput of light browsing, heavy browsing and video streaming. The green trendline represents video streaming.

Figure 2-20 shows that the average throughput of video streaming is more than 10 times higher than that of the two other cases. This intuitive result is explained by the fact that video streaming entails transferring a much larger volume of data than regular web browsing.

Figure 2-21 compares the average delay in the three cases considered for Scenario 1.

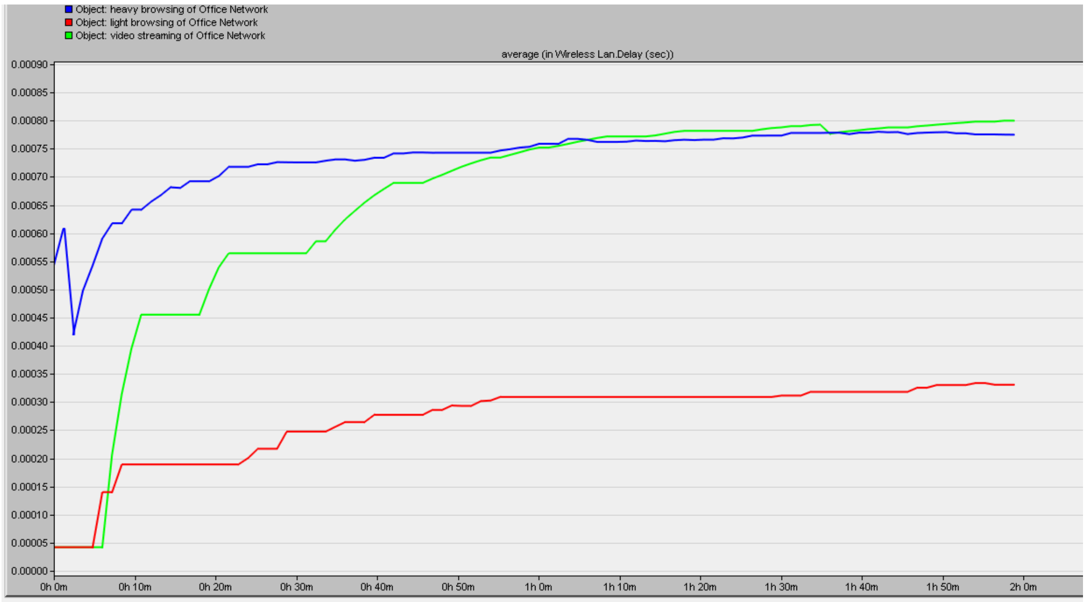


Figure 2-21: Comparison of average delay of light browsing, heavy browsing and video streaming

In Figure 2-21, heavy browsing has a higher average delay than light browsing due to the higher volume of data transferred. However, the average delay of video streaming is pretty close to that of heavy browsing. The possible reason is that we are simulating under a WiFi environment with a very high data rate so that the video streaming will not have as much delay as we assumed.

2.4.2 Scenario 2: Effect of using different video streaming resolutions

2.4.2.1 Topology

The wireless LAN parameters, such as data rate and link between router and server, from scenario 1 in Subsection 2.4.1 are retained. The range between the router and the two YouTube video nodes remains unchanged. The YouTube video streaming HTTP video browsing application is retained because YouTube uses HTTP and TCP protocols for video streaming as found earlier in the literature review in Subsection 1.2.1. Figure 2-22 illustrates the topology employed for our second scenario simulations.

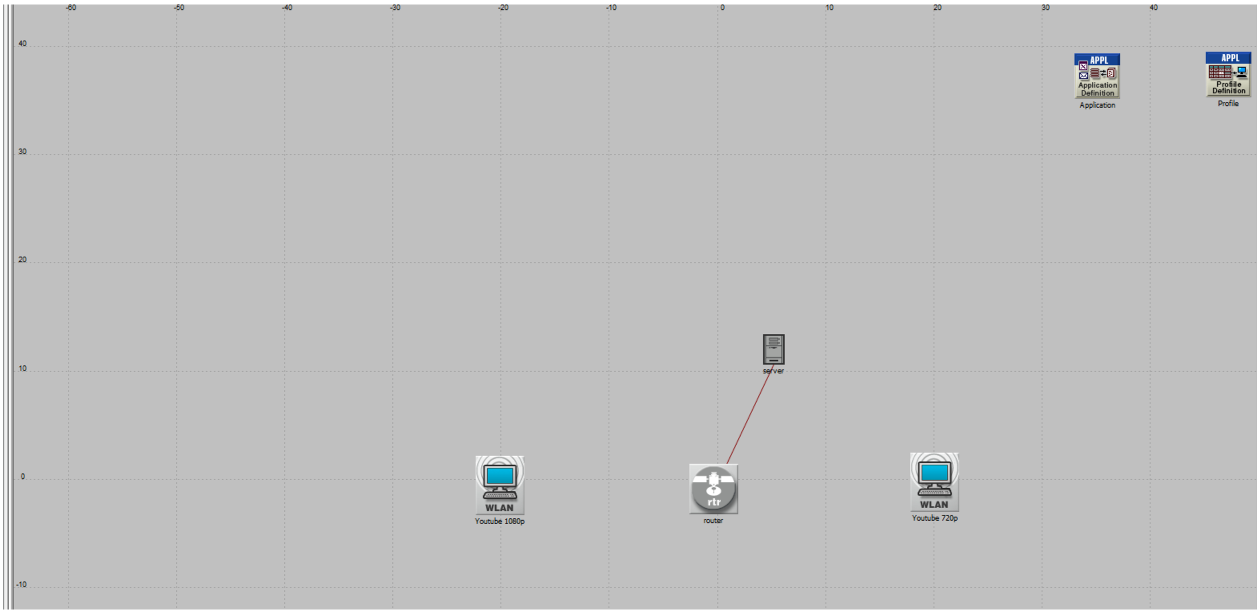
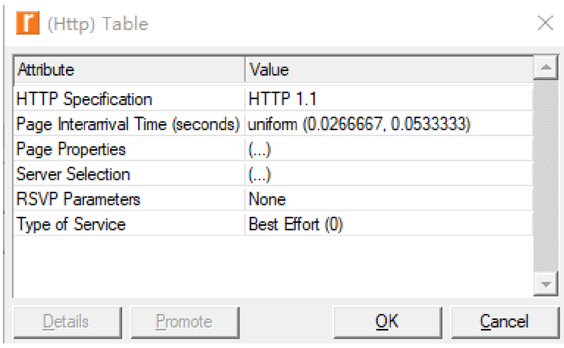


Figure 2-22: Scenario 2: topology

We will also use the YouTube application node for the next few scenarios in this project. YouTube 1080p and 720p use video codec H-264 and the average encoding speed of H-264 is between 24 fps (frames per second) to 30 fps.

We can calculate the page inter-arrival time for both YouTube 1080p and 720p by using the frame rate, which are 30 fps for 1080p and 24 fps for 720p. For our simulation, the page inter-arrival time of 1080p is set between 0.0333333 s and 0.0666666 s, the page inter-arrival time of 720p is set between 0.0266667 s and 0.0533333 s. The page size of both 720p and 1080p are set as short video which is approximately 10000 to 350000 bytes.

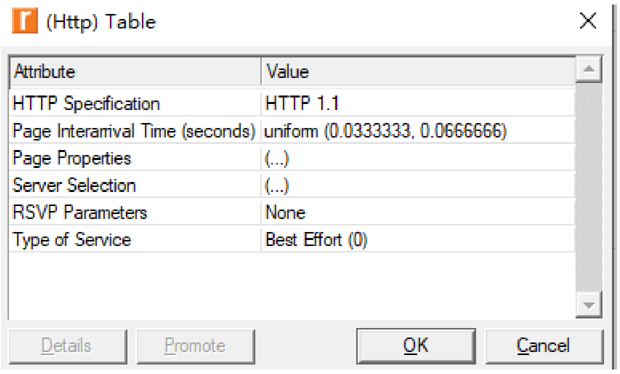
Figures 2-23 and 2-24 show the HTTP tables for YouTube 720p and 1080p respectively.


(Http) Table

Attribute	Value
HTTP Specification	HTTP 1.1
Page Interarrival Time (seconds)	uniform (0.0266667, 0.0533333)
Page Properties	(...)
Server Selection	(...)
RSVP Parameters	None
Type of Service	Best Effort (0)

Details Promote OK Cancel

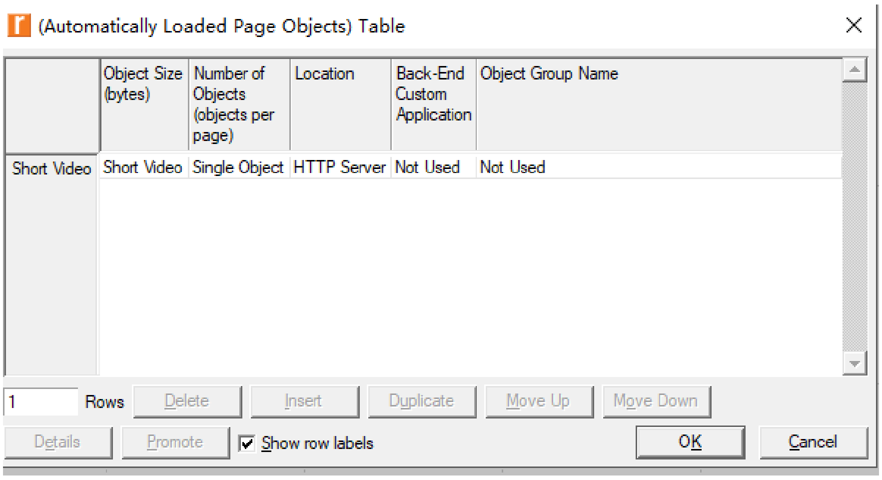
Figure 2-23: Scenario 2: YouTube 720p HTTP table



Attribute	Value
HTTP Specification	HTTP 1.1
Page Interarrival Time (seconds)	uniform (0.0333333, 0.0666666)
Page Properties	(...)
Server Selection	(...)
RSVP Parameters	None
Type of Service	Best Effort (0)

Figure 2-24: Scenario 2: HTTP table for YouTube 1080p

Figure 2-25 shows the object size for YouTube 720p and 1080p.



	Object Size (bytes)	Number of Objects (objects per page)	Location	Back-End Custom Application	Object Group Name
Short Video	Short Video	Single Object	HTTP Server	Not Used	Not Used

Figure 2-25: Scenario 2: object size for YouTube 720p and 1080p

2.4.2.2 Scenario 2 results

Figure 2-26 shows that the throughput of 1080p is more than 10000 bps higher than that of 720p as is expected. This is because YouTube 1080p has a longer page inter-arrival time, therefore its frame rate is higher than that of 720p.

Usually 1080p has a frame rate of 30 fps and 720p has a frame rate of 24 fps. With a higher frame rate, 1080p can show more frames per unit time than 720p, which explains why its throughput is higher than that of 720p.

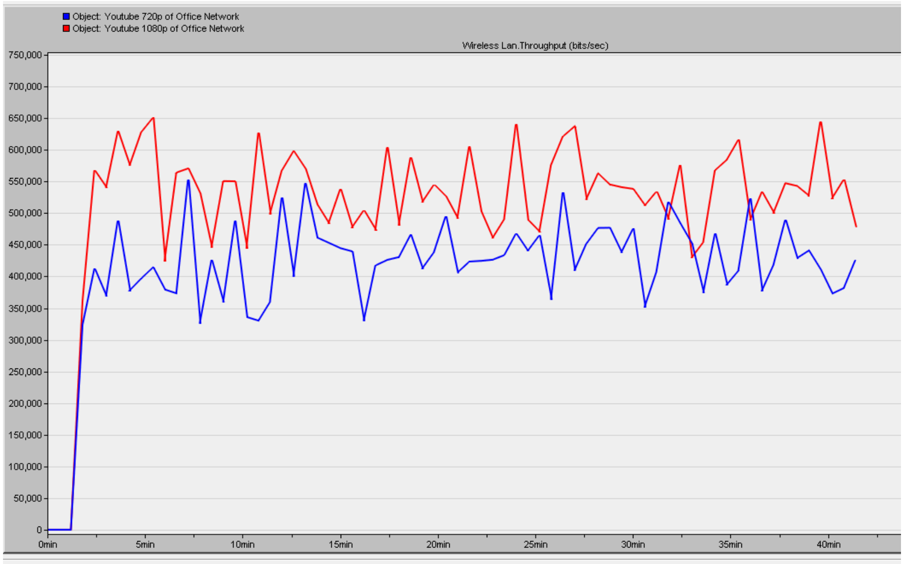


Figure 2-26: Scenario 2: Throughput with 720p and 1080p streaming

Figure 2-27 shows that the delay of 1080p is higher than that of 720p. This is expected, because the delay is highly dependent on the throughput. Since 1080p has a higher throughput, it also has a higher delay than 720p.

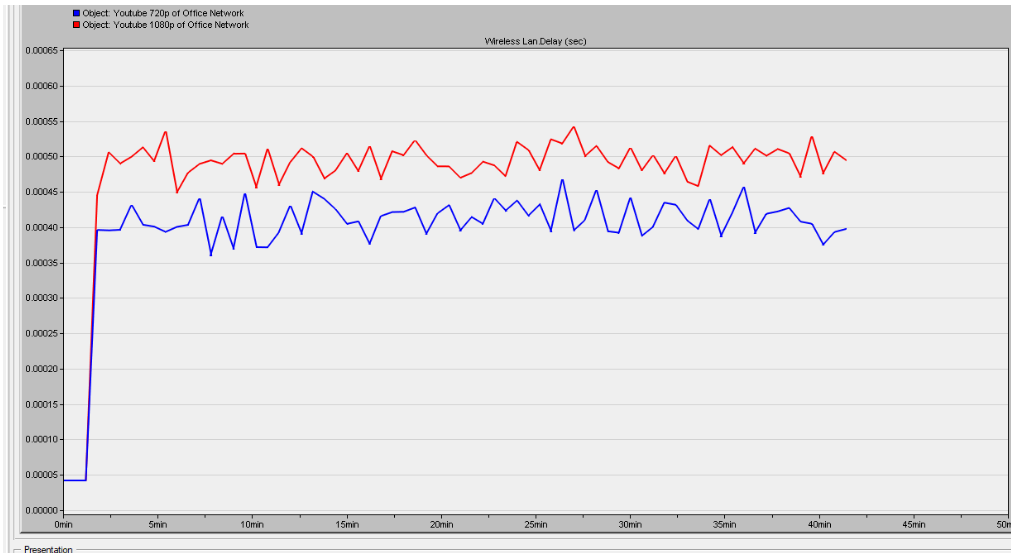


Figure 2-27: Scenario 2: Delay with 720p and 1080p streaming

As can be seen in Figure 2-28, 1080p receives more data traffic than 720p from the server. This is due to YouTube 1080p having a higher frame rate. Therefore the server needs to transfer more data per unit time to the streaming client requesting the 1080p video.

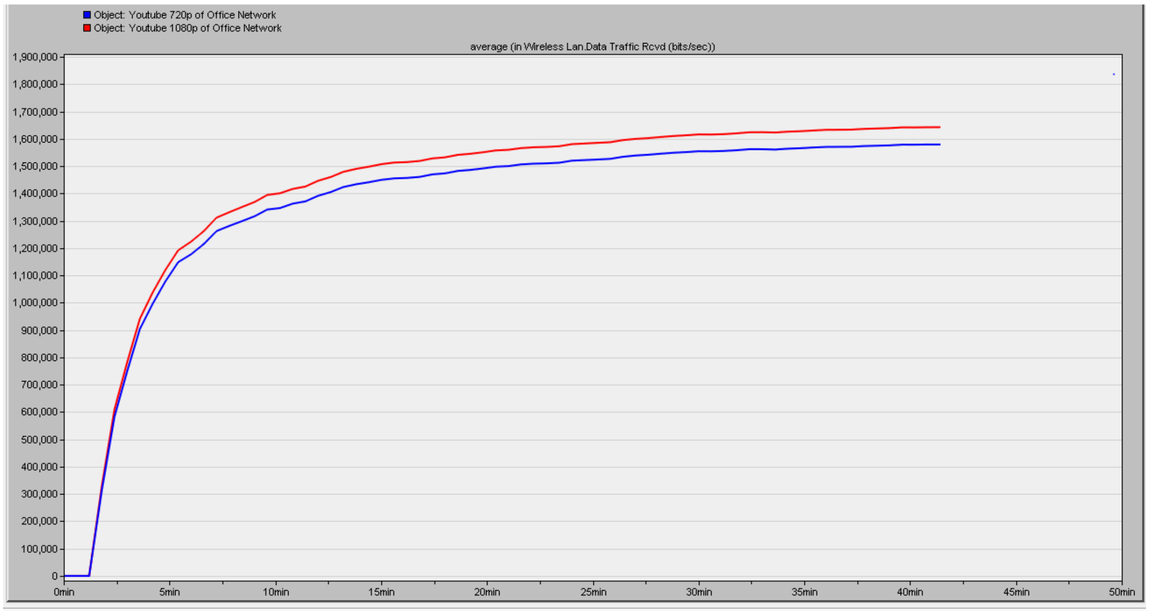


Figure 2-28: Scenario 2: Data traffic received with 720p and 1080p streaming

2.4.3 Scenario 3: Effect of frequency band, data rate & WiFi technology

2.4.3.1 Scenario 3 topology

Scenario 3 investigates three aspects:

- the impact of the WiFi technology employed in Subsection 2.4.3.2.
- the effect of the frequency band (2.4 GHz or 5 GHz) employed in Subsection 2.4.3.3.
- the effect of changing the data rate in Subsection 2.4.3.4.

The topology employed in this scenario is very similar to that used in scenario 1 in Subsection 2.4.1, with the only difference being substituting the video streaming node in scenario 1 by the YouTube 1080p node from scenario 2 in Subsection 2.4.2. Figure 2-29 shows the topology implemented in scenario 3.

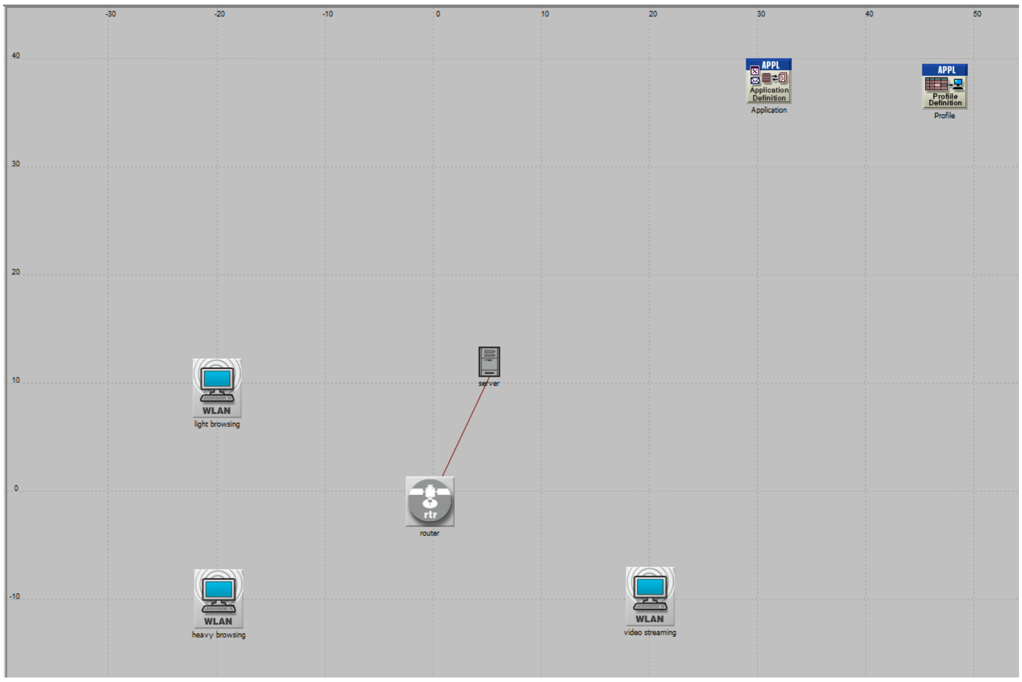


Figure 2-29: Topology employed for Scenario 3: A WLAN client streams videos at 1080p.

2.4.3.2 Impact of WiFi technology

To study the impact of the specific WiFi technology used, we chose IEEE 802.11g operating with a data rate of 24 Mbps and IEEE 802.11n with a data rate of 26 Mbps since their data rates are arguably very similar to each other. Figure 2-30 shows the average throughput achieved with this pair of WiFi technologies. The average throughput of IEEE 802.11n is higher than that of IEEE 802.11g by more than 60000 bps.

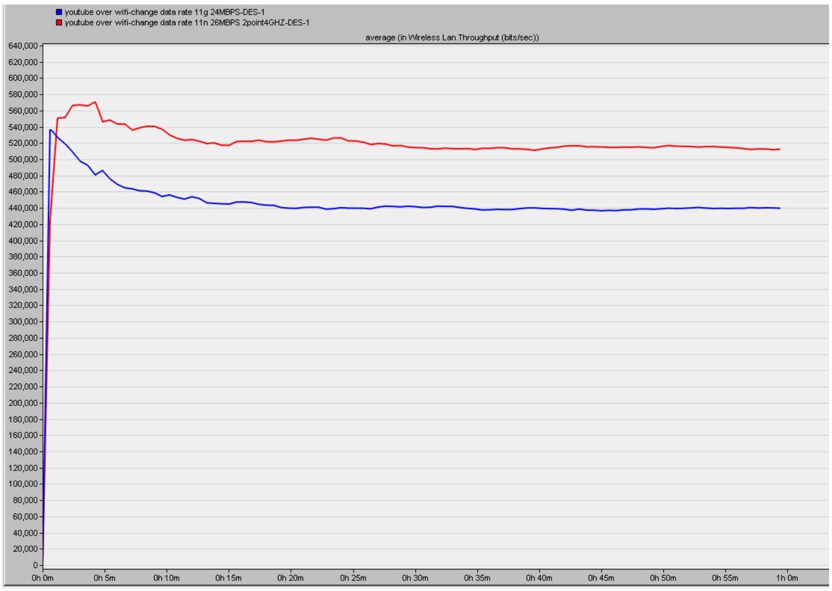


Figure 2-30: Scenario 3: Average throughput with IEEE 802.11g at 24 Mbps (blue graph) and IEEE 802.11n at 26 Mbps (red graph).

Figure 2-31 shows the average delay with this pair of WiFi technologies. The average delay of IEEE 802.11g is higher than that of IEEE 802.11n by more than 0.0008 s.

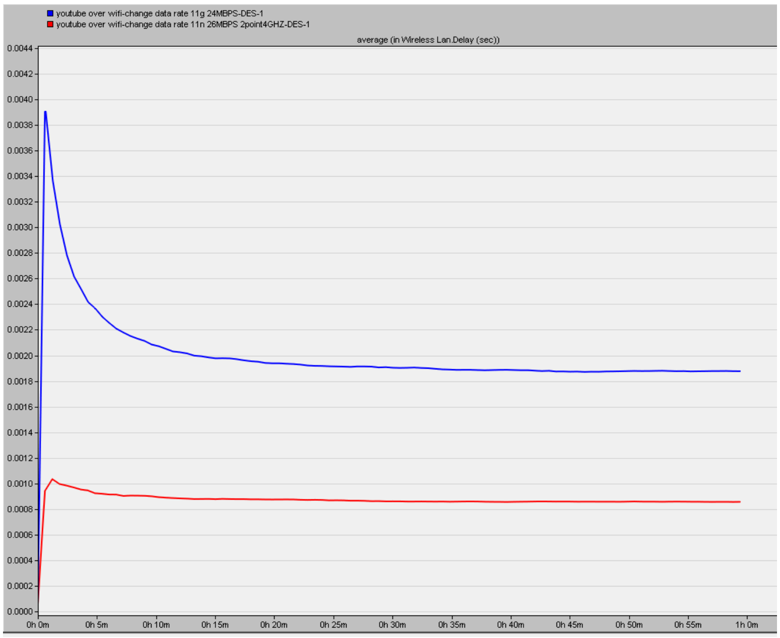


Figure 2-31: Scenario 3: Average delay with IEEE 802.11g at 24 Mbps (blue graph) and IEEE 802.11n at 26 Mbps (red graph).

The improved throughput and delay experienced with IEEE 802.11n is more likely thanks to its higher data rate of 26 Mbps than to any inherent superiority in technology over IEEE 802.11g.

2.4.3.3 Effect of frequency band

To investigate the effect of frequency band, we employed the dual-band capable IEEE 802.11n. We used the same data rate of 26 Mbps and physical characteristics with the YouTube 1080p node.

Figure 2-32 shows the average throughput experienced in the two frequency bands. Since the two graphs are very close to each other, we can conclude that within the parameters of our experiment, the throughput performance is independent of the choice of frequency band.

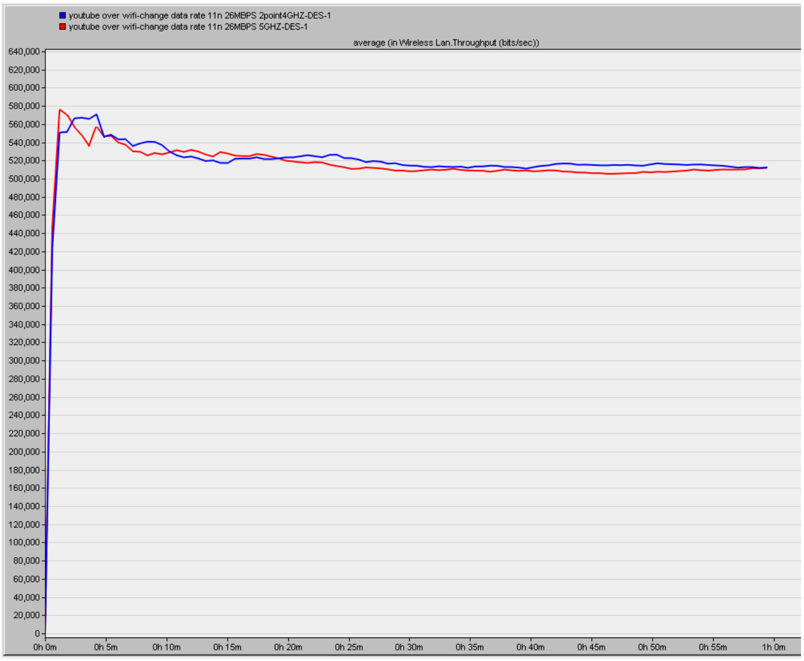


Figure 2-32: Scenario 3: average throughput of IEEE 802.11n at 26 Mbps: 2.4 GHz (blue graph) and 5 GHz (red graph).

Figure 2-33 shows the average delay experienced in the two frequency bands. Since the two graphs are very close to each other, we can conclude that within the parameters of our experiment, the delay performance is independent of the choice of frequency band.

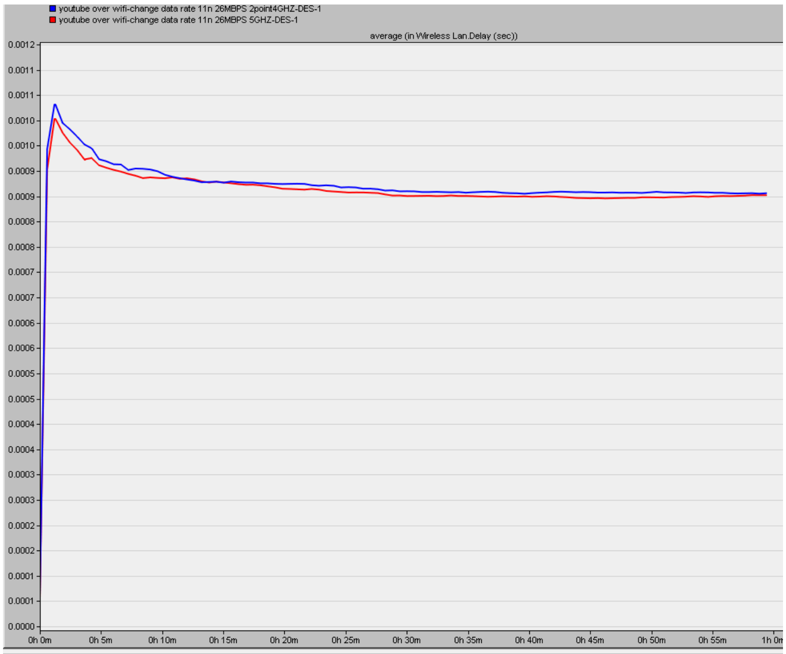


Figure 2-33: Scenario 3: average delay with IEEE 802.11n at 26 Mbps: 2.4 GHz (blue graph) and 5 GHz (red graph).

The frequency band chosen does not have a significant effect within the parameters of our experiment because the WiFi client was very close to the WiFi access point and therefore attenuation effects did not come into play. Furthermore, since there were not a large number of clients sharing the same WiFi connection, the client could enjoy the full capacity of the WiFi connection.

2.4.3.4 Effect of data rate

Finally, we investigate the effect of changing the data rate on the performance experienced by the YouTube 1080p node. We employ the IEEE 802.11n standard in the 2.4 GHz band.

Figure 2-34 shows the average throughput experienced with the chosen data rates. As expected, with the higher data rate, the throughput experienced is higher than at the lower data rate.

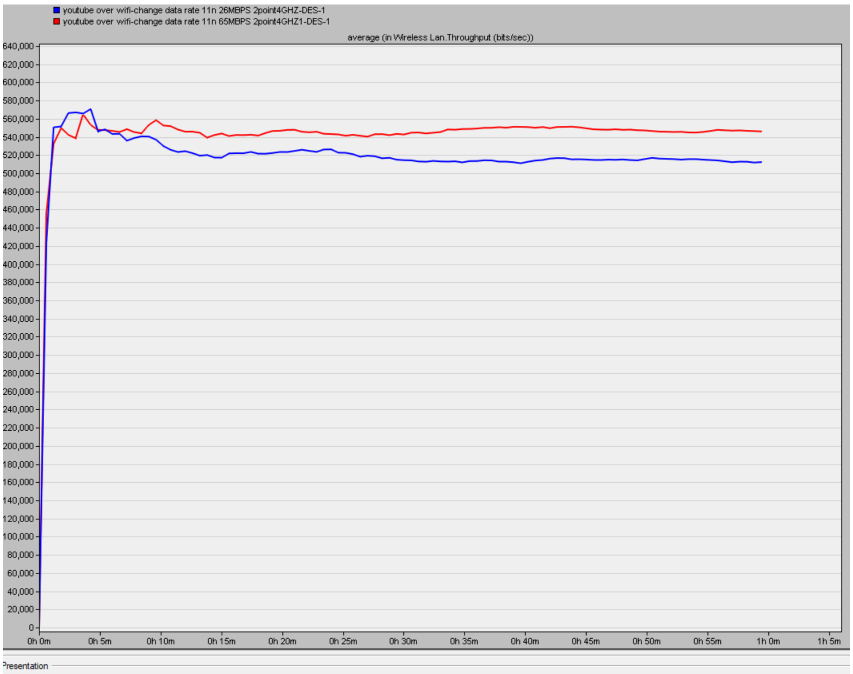


Figure 2-34: Scenario 3: average throughput of IEEE 802.11n at 2.4 GHz: 26 Mbps (blue graph) and 65 Mbps (red graph).

Figure 2-35 shows the average delay experienced with the chosen data rates. The WiFi client experiences more delay at the lower data rate.

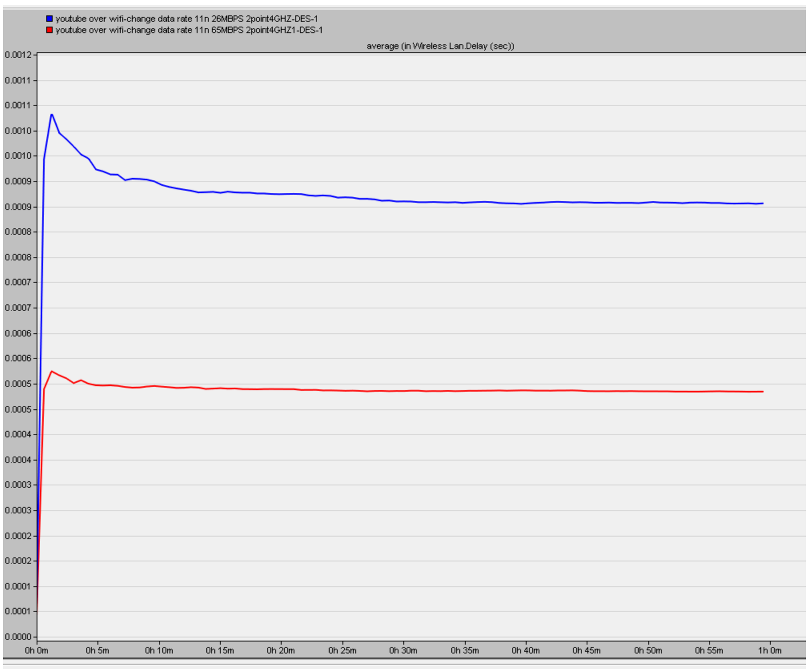


Figure 2-35: Scenario 3: average delay of IEEE 802.11n at 2.4 GHz: 26 Mbps (blue graph) and 65 Mbps (red graph).

2.4.4 Scenario 4: Effect of varying the range to the WiFi access point

Scenario 4 employs the same WLAN parameter settings as scenario 3 in Subsection 2.4.3, except that the YouTube 1080p node was moved almost 200 meters away from the router. The objective is to investigate the effect of changing the range to the WiFi access point on the performance of the 1080p node. Physical characteristics, data rate and IEEE 802.11 technology employed are varied in this set of experiments. The range from the user to the router in Scenario 4 is exaggerated on purpose to better show the impact of the range.

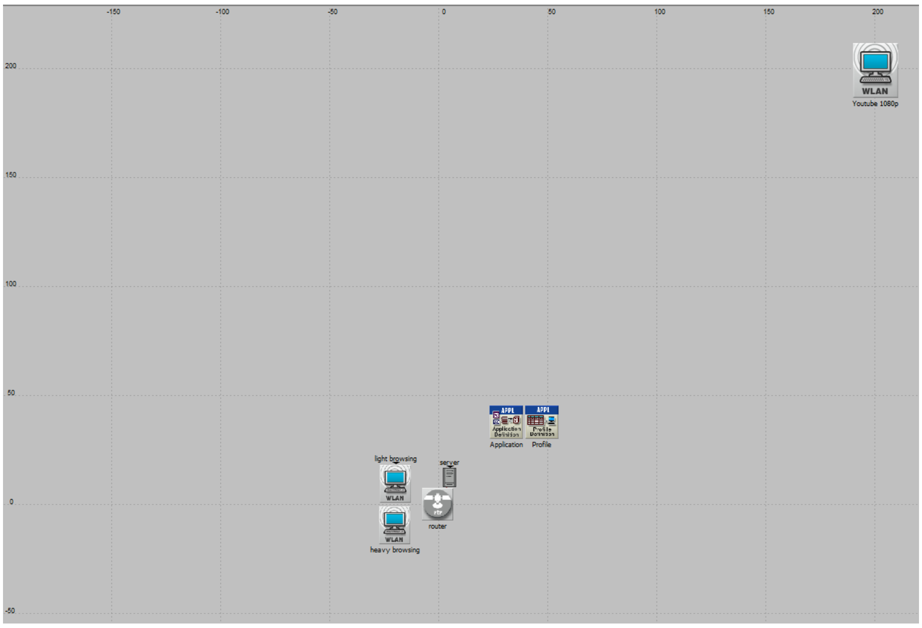


Figure 2-36: Scenario 4: topology

Figure 2-37 shows the throughput achieved at data rates of 39 Mbps (blue graph) and 65 Mbps (red graph) when employing IEEE 802.11n in the 2.4 GHz band. The throughput dropped to zero in the 65 Mbps case for some unexplained reason. We suspect that we made a mistake in our simulation but we have not been able to debug this issue.

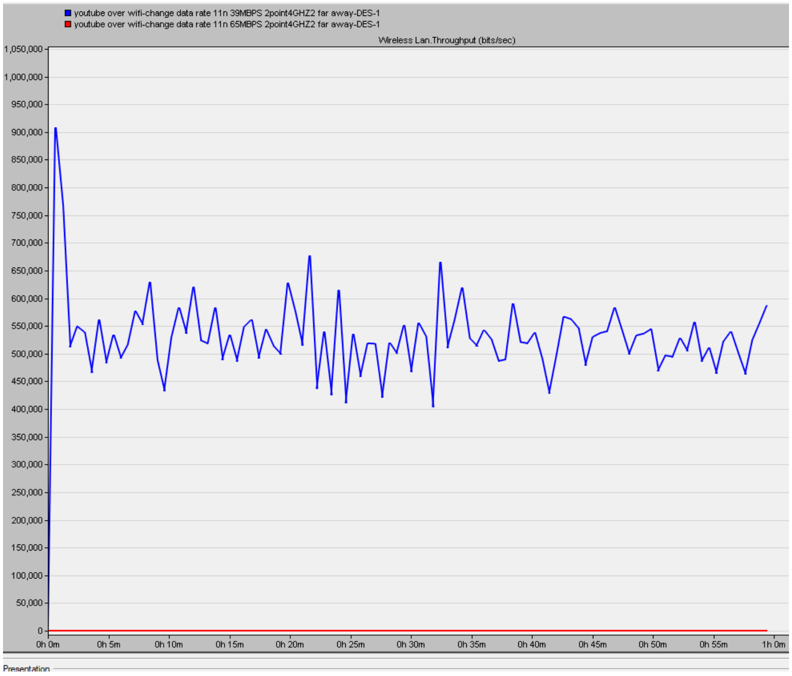


Figure 2-37: Scenario 4: throughput with IEEE 802.11n at 2.4 GHz: data rate of 39 Mbps (blue graph) and 64 Mbps (red graph).

We retained IEEE 802.11n operating in the 2.4 GHz band as technology and investigated the effect of changing the data rate from 39 Mbps to 26 Mbps. Figure 2-38 shows the average throughput graphs while Figure 2-39 shows the corresponding average delay graphs.

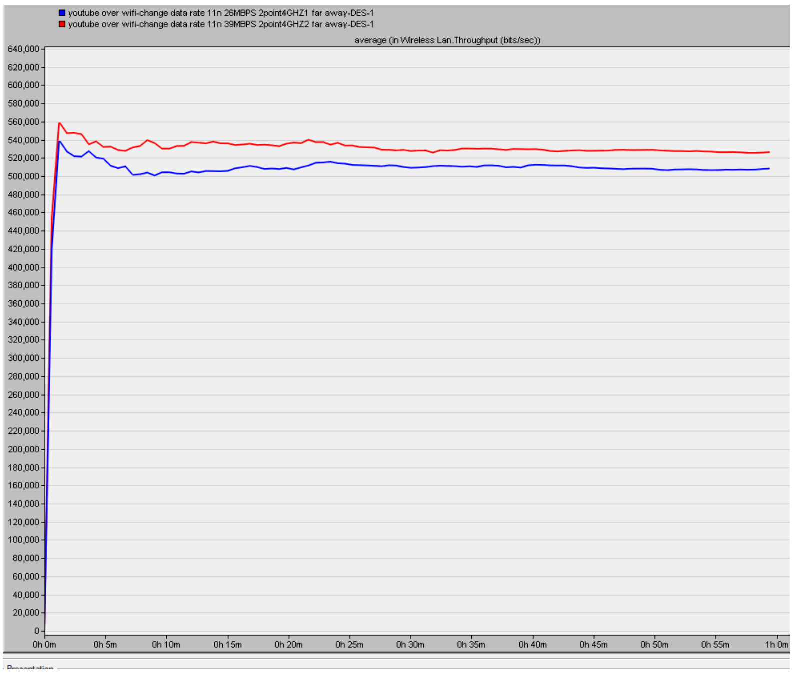


Figure 2-38: Scenario 4: Average WLAN throughput with IEEE 802.11n at 2.4 GHz: data rate of 26 Mbps (blue graph) and 39 Mbps (red graph).

As can be seen in Figure 2-38, the throughput experienced by the YouTube 1080p node remains acceptable when the data rate chosen is lower than 65 Mbps which was found to be problematic in Figure 2-37. With the streaming client node being located further away from the router, the higher data rate 39 Mbps of the WiFi connection leads to higher throughput (c.f. Figure 2-38) and lower delay (c.f. Figure 2-39), as expected. With a higher data rate (as long as it is lower than 65 Mbps), the streaming client enjoys better QoS even when located further away from the WiFi access point.

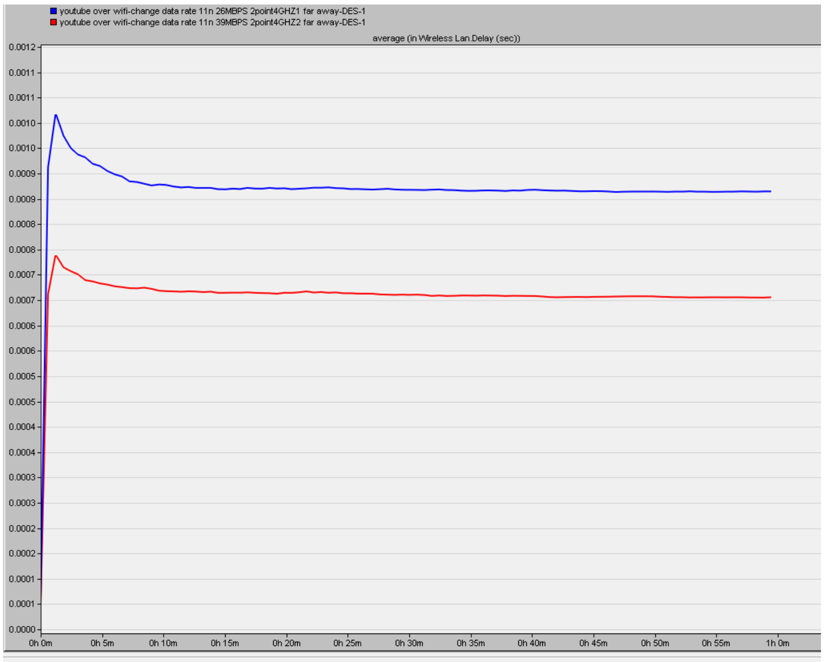


Figure 2-39: Scenario 4: Average WLAN delay with IEEE 802.11n at 2.4 GHz: data rate of 26 Mbps (blue graph) and 39 Mbps (red graph).

We then investigated the effect of increasing the range when the WLAN operates in either of the two frequency bands supported by IEEE 802.11n.

Figure 2-40 shows the average throughput achieved in both frequency bands of IEEE 802.11n operating at a data rate of 26 Mbps.

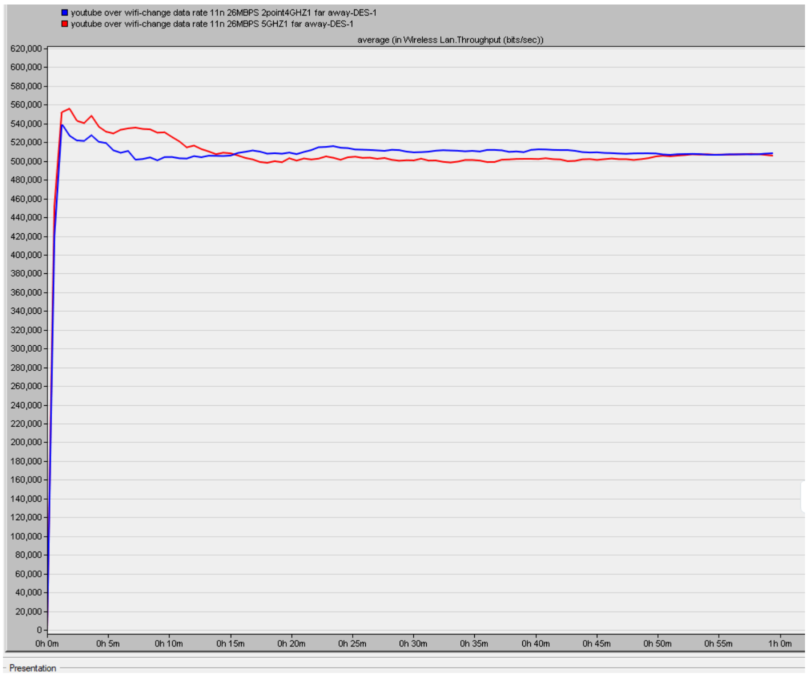


Figure 2-40: Scenario 4: average throughput in the 2.4 GHz band (blue graph) and 5 GHz (red graph) with a data rate of 26 Mbps

As can be seen in Figure 2-40, the streaming client enjoys similar average throughputs in either of the frequency bands. It is important to note that the number of WiFi clients was not a limiting factor to the performance in our simulation, therefore the higher host accommodation capacity of the 5 GHz band has no impact in our simulation.

Figure 2-41 shows the average delay achieved in both frequency bands of IEEE 802.11n operating at a data rate of 26 Mbps.

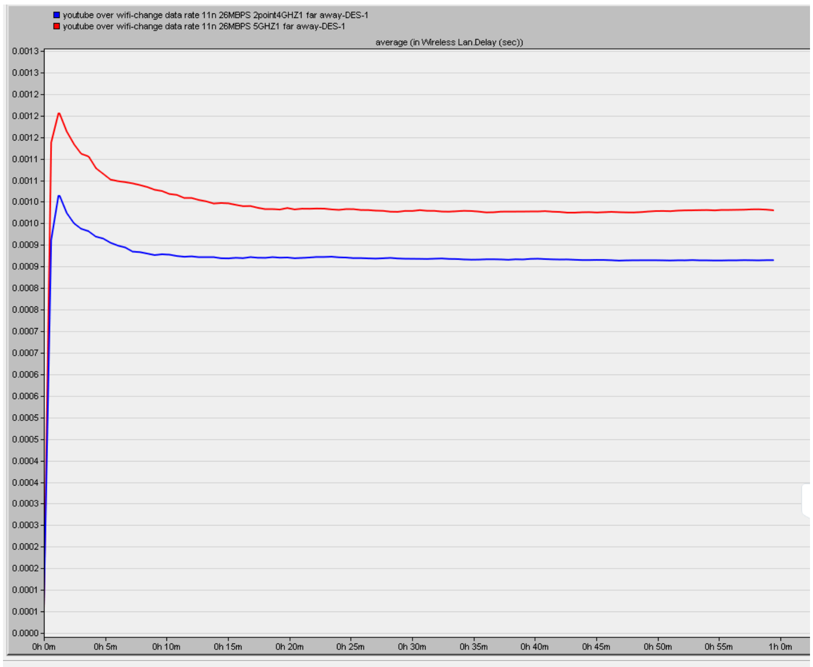


Figure 2-41: Scenario 4: average delay in the 2.4 GHz band (blue graph) and 5 GHz (red graph) with a data rate of 26 Mbps

As can be seen in Figure 2-41, when the client streams in the 5 GHz band, it experiences a higher average delay than operating in the 2.4 GHz band in this particular case where the client is further away from the router. This is explained by the fact that the power of radio signals attenuate more significantly in higher frequency bands (such as the 5 GHz band) than in lower frequency bands (such as the 2.4 GHz band). Furthermore, objects of similar size to the wavelength of 6 cm of 5 GHz WiFi readily absorb it, as was discussed in Subsection 1.2.2. These phenomena are observed when the range to the transmitter (i.e. the WiFi access point) is higher but not when the range is negligible as we found earlier in Subsection 2.4.3.3.

Finally, we compared the average data dropped in the current scenario (higher range between the client and router) under various conditions to bring together the different scenarios we investigated.

Figure 2-42 shows that when IEEE 802.11g is used, the client experiences much smaller average data dropped than when any other IEEE 802.11n case is used. When located further away from the router, the YouTube 1080p streaming node enjoys better QoS with IEEE 802.11g than with IEEE 802.11n. This result comes with the caveat that the streaming client did not have to share the WiFi connection with a large number of co-hosts.

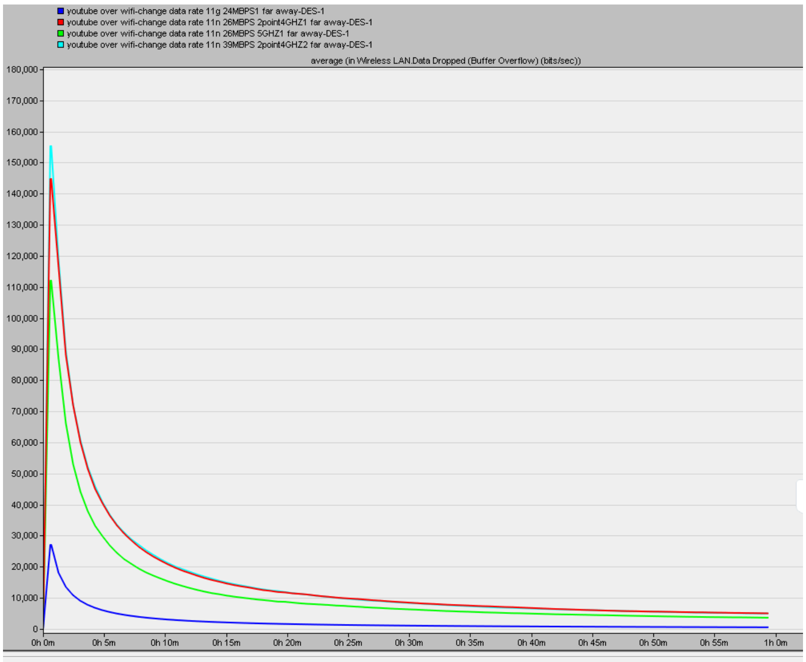


Figure 2-42: Scenario 4: average data dropped under various conditions: IEEE 802.11g with a data rate of 24 Mbps (blue graph), IEEE 802.11n (2.4 GHz) at 26 Mbps (red graph), IEEE 802.11n (5 GHz) at 26 Mbps (green graph) and IEEE 802.11n (2.4 GHz) at 39 Mbps (pale blue graph)

2.5 Summary

This chapter has discussed at length the design and simulation of experiments conducted in this project. It has described preliminary investigations such as pinging YouTube in Section 2.2 and studying a YouTube streaming trace with Wireshark 2.3. Based on these findings, the discussion moved to the experiments done using Riverbed Modeler.

The conclusions drawn on the work presented in this chapter are elaborated on in Chapter 3.

Chapter 3

Discussion and Conclusion

Section 3.1 reviews the experiments conducted in Chapter 2. Then Section 3.2 outlines how the goals of this project were achieved. Finally, Section 3.3 puts forth our recommendations for future work.

3.1 Discussion

The main aim of this project has been to simulate Youtube video streaming over WiFi with a discrete network simulator. To accomplish this aim, a number of scenarios featuring WiFi links and applications for YouTube streaming were created and simulated in Riverbed Modeler. The work done in this project can be summarized in detailed terms as follows:

- Scenario 1: Light browsing, heavy browsing and video streaming.
  Scenario 1 in Subsection 2.4.1 served as learning step in this project. It investigated how the throughput and delay experienced by a streaming client are affected by the type of browsing done. While the results presented may seem intuitive at a first glance, they serve as sanity check to confirm that we were on the right path in this project.

- Scenario 2: Effect of using different video streaming resolutions.
  Scenario 2 in Subsection 2.4.2 compared the performance of streaming at 720p to streaming at 1080p. This was implemented by changing the page inter-arrival time attribute in our Riverbed Modeler simulation. The longer page inter-arrival time of YouTube 1080p compared to YouTube 720p leads to a higher frame. This results in higher throughput experienced by the streaming client. In addition, due to the higher frame rate demanded by 1080p, the delay experienced by the client is higher in this case.

- Scenario 3: Effect of frequency band, data rate and WiFi technology employed.
  Scenario 3 in Subsection 2.4.3 focused solely on the 1080p streaming node. It investigated three aspects:
  - the impact of the WiFi technology employed (either IEEE 802.11g or n.)
    With the same data rate, the performance with IEEE 802.11g and IEEE 802.11n were observed. The improved throughput and delay experienced with IEEE 802.11n is more likely thanks to its higher data rate of 26 Mbps compared to the 24 Mbps data rate of IEEE 802.11g.
  - the effect of the frequency band (2.4 GHz or 5 GHz) exploited.
    The dual-band capable IEEE 802.11n technology was employed for this case. The average delay experienced in the two frequency bands was found to be very similar. Within the parameters of this experiment (few users sharing the WiFi resource, negligible range between client and router), the delay performance is independent of the choice of frequency band.
  - the effect of changing the data rate.
    With higher data rates, the client enjoys higher throughput and lower delay and therefore better overall QoS.
- Scenario 4: Effect of varying the range to the WiFi access point.
  Scenario 4 investigated how the physical characteristics, data rate and IEEE 802.11 technology employed affect the QoS experienced by the streaming client when they are located further away from the router. For an unexplained reason, the client had no throughput when a data rate of 65 Mbps was employed. We suspect that we made a mistake in our simulation but we have not yet been able to debug this issue. With a higher data rate of 39 Mbps versus 26 Mbps, the streaming client enjoys better QoS even when located further away from the router. The streaming client enjoys mostly the same average throughputs in either of the frequency bands provided the same data rate is available. When a client streams in the 5 GHz band, it experiences a higher average delay than when operating in the 2.4 GHz band. This is explained by the more significant power attenuation of radio signals of high frequency due to range. These phenomena are observed when the range to the WiFi transmitter is higher but not when the range is negligible as was found previously. Finally, when the IEEE 802.11g technology is employed, the client experiences much lower average data dropped than when any other IEEE 802.11n configuration is used. This results comes with the caveat that the streaming client did not have to share the WiFi connection with a large number of co-hosts in our experiments.

3.2 Conclusions

In light of the foregoing discussion in Section 3.1, we reach the following conclusions.

- Scenario 1: Light browsing, heavy browsing and video streaming.
  When the client browses small files over the Internet, the throughput and delay are less significant. When browsing large files, the client used more data and hence more throughput was required.
- Scenario 2: Effect of using different video streaming resolutions.
  From the results of scenario 2, it is clear that a higher resolution demanded by the client and consequently, a higher frame rate will result in better performance and will give high throughput, as in the case of YouTube 1080p.
- Scenario 3: Effect of frequency band, data rate and WiFi technology employed.
  At negligible ranges to the router, the choice of frequency band has no impact on the QoS experienced by the streaming client. Secondly, IEEE 802.11n and high data rates lead to better streaming performance on the streaming client side.
- Scenario 4: Effect of varying the range to the WiFi access point.
  Due to its shorter wavelength, WiFi at 5 GHz gives worse QoS than at 2.4 GHz when the WiFi user is at an appreciable range from the router. IEEE 802.11g has more stable performance than IEEE 802.11n when the client is located further away from the router.

Our project is non-trivial because we implemented thorough simulations exploring browsing the Internet and streaming videos using WiFi. To ensure that our simulations can be related to real-life situations, we selected four specific scenarios to replicate the breadth of the experience of a single client browsing on the Internet and streaming YouTube videos in their residence using a WiFi network. The main difficulty that we faced is that Academic Edition of Riverbed Modeler did not allow us to import trace file into simulations and that we could not use the DASH functionality to more accurately simulated video streaming.

3.3 Recommendations

Following the conclusions drawn in Section 3.2, we put forth the following recommendations for future work:

- Increase the complexity of the Riverbed Modeler simulations by adding more nodes.
  Due to the duration of our simulations, we were not able to include additional clients in

the WLAN. It would be beneficial to incorporate additional users in the simulations as in real-life situations, it is very likely that several users will be sharing a WiFi connection. Furthermore, mobility nodes can be added since many WiFi clients are smartphones and tablets which are mobile.

- Implement the simulation scenarios using ns-3.
  The discrete network event simulator ns-3 has an LTE module and a DASH module. This means that ns-3 can be used to create a more realistic simulation of YouTube streaming by employing the DASH module. Furthermore, given the ever-growing popularity of LTE as technology, it would be beneficial to investigate YouTube video streaming using LTE and potentially do a comparison with using WiFi.

- If a professional version of Riverbed Modeler is available, it is recommended to import YouTube streaming trace files generated with Wireshark. Several trace files of streaming at various resolutions (480p, 720p, 1080p and 2160p). Furthermore, the DASH functionality of the full version of Riverbed Modeler may be exploited to develop simulations which replicate the actual technology employed by YouTube. Finally, System-In-The-Loop simulations can be done with a full version of Riverbed Modeler to have real-world results, instead of relying on simulated results.

Bibliography

[1] Sandvine, "The global internet phenomena report," Tech. Rep. MSU-CSE-06-2, Sandvine Incorporated, Fremont, California, U.S.A., October 2018. Published Online. [i, 1]

[2] M. Trevisan, D. Giordano, I. Drago, M. M. Munafò, and M. Mellia, "Five years at the edge: Watching internet from the isp network," *IEEE/ACM Transactions on Networking*, pp. 1–14, 2020. [1]

[3] A. Rao, A. Legout, Y.-s. Lim, D. Towsley, C. Barakat, and W. Dabbous, "Network characteristics of video streaming traffic," in *Proceedings of the Seventh Conference on emerging Networking EXperiments and Technologies*, pp. 1–12, 2011. [2]

[4] "Delivering Live YouTube Content via DASH." https://developers.google.com/youtube/v3/live/guides/encoding-with-dash. Accessed: 2020-03-31. [3]

[5] "ISO/IEC 23009-1:2014 Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats." https://www.iso.org/standard/65274.html. Accessed: 2020-03-31. [3]

[6] "ISO/IEC 23009-1:2019 Information technology — Dynamic adaptive streaming over HTTP — Part 1: Media presentation description and segment formats." https://www.iso.org/standard/75485.html. Accessed: 2020-03-31. [3]

[7] "MPEG ratifies its draft standard for DASH." https://web.archive.org/web/20120820233136/http://mpeg.chiariglione.org/meetings/geneva11-1/geneva_press.htm. Accessed: 2020-03-31. [3]

[8] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over http dataset," in *Proceedings of the 3rd Multimedia Systems Conference*, MMSys '12, (New York, NY, USA), p. 89–94, Association for Computing Machinery, 2012. [3]

[9] "Adaptive Streaming Overview." https://commons.wikimedia.org/w/index.php?curid=20054893. Accessed: 2020-03-31. [3]

[10] W. D. D. Maza, "A framework for generating http adaptive streaming traffic in ns-3," in *EAI International Conference on Simulation Tools and Techniques*, 2016. [3]

[11] S. Banerji and R. S. Chowdhury, "On IEEE 802.11: wireless LAN technology," *International Journal of Mobile Network Communications & Telematics (IJMNCT)*, vol. 3, 2013. [4]

[12] "Change the Wi-Fi Channel Number to Avoid Interference." https://www.lifewire.com/wifi-channel-number-change-to-avoid-interference-818208. Accessed: 2020-03-31. [4]

[13] "Quality of Service." https://www.cs.rutgers.edu/~pxk/417/notes/03-qos.html. Accessed: 2020-03-31. [4]

[14] "Riverbed Modeler." https://cms-api.riverbed.com/portal/community_home. Accessed: 2020-03-26. [5]

[15] "nsnam wiki." https://www.nsnam.org/wiki/Main_Page. Accessed: 2020-01-25. [5]

[16] "What are the restrictions for riverbed modeler academic edition 17.5?." https://supportkb.riverbed.com/support/index?page=content&id=S24443. Accessed: 2020-04-05. [5]

[17] Y. M. Hassan, A. Helmy, and M. M. Rehan, "Effect of varying segment size on dash streaming quality for mobile user," in *2014 International Conference on Engineering and Technology (ICET)*, pp. 1–4, April 2014. [5]

[18] M. A. Mohamed and H. F. Ibrahim, "Performance evaluation for video streaming data over lte-a networks," in *2017 12th International Conference on Computer Engineering and Systems (ICCES)*, pp. 701–707, Dec 2017. [5]

[19] X. Zhenpeng, "Video Streaming over the 802.11g WLAN Technologies." [6]

[20] S. Calzada, C. Rietchel, and T. Szajner, "Performance Analysis of a Wireless Home Network." [6]

[21] A. Singh and D. Labayo, "Performance Analysis of Video Streaming over Wi-Fi and Ethernet." [6]

[22] J. Kim, J. Zheng, and P. Bertsch, "Video Streaming over Wi-Fi." [6]

[23] M. Ng and C. H. Weng, "Video Streaming over WiFi using Riverbed Modeler." [6]

[24] "802.11n-2009 - IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput." https://standards.ieee.org/standard/802_11n-2009.html. Accessed: 2020-04-15. [12]

Appendix A

Source files for the project

In an effort to contribute to replicability in academic research and development, we provide our *Riverbed Modeler* simulation files and source files for our reports. The links provided here are also included on our project website at http://www.sfu.ca/~hdhondea/ENSC894Group2.html.

A.1 Riverbed Modeler simulations

Our Riverbed Modeler simulations files can be found at https://drive.google.com/open?id=1jkT0fj6gilaaUDCpsAAJ6afOg3kQ6gfL.

A.2 Report

This report was typeset using LaTeX. It can be reproduced by cloning the *GitHub* repository at https://github.com/AshivDhondea/ENSC894_project_report and compiling the script `main.tex` with the following command sequence:

1. `PDFLaTeX`
2. `Bibtex`
3. `PDFLaTeX`
4. `PDFLaTeX`
5. `Makeglossaries`
6. `PDFLaTeX`
7. `PDFLaTeX`