

Investigating the Network Characteristics of Two Popular Web-Based Video Streaming Sites

Dr. Feras Mohammed Almatarneh

University of Tabuk

Department of Computer Science, College Duba Saudi Arabia

Doi: 10.19044/esj.2017.v13n33p305 [URL:http://dx.doi.org/10.19044/esj.2017.v13n33p305](http://dx.doi.org/10.19044/esj.2017.v13n33p305)

Abstract

The determinants of the strategies to be employed by video streaming sites are application (mobile devices or web browsers) and container of the video application. They affect video streaming network characteristics, which is often the traffic flow, and its quality. It is against this background that studies on streaming strategies suggested the need to investigate and identify the relationship between buffer time, video stream protocol, packet speed and size, upload time, and waiting period, specifically to aid network administrative support in case of network traffic bottlenecks. In view of this, this study investigates the network characteristics of YouTube and Vimeo, using experimental methodology, and involving WireShark as network analyzer. Google Chrome and Firefox are the web browsers employed, while packet size, protocols, packet interval, TCP window size and accumulation ratio are the metrics. Short ON-OFF, Long ON-OFF, and No ON-OFF cycles are the three streaming strategies identified. It is further shown that both Vimeo and YouTube employ these strategies but the choice depends on the container of the video streamed.

Keywords: Network characteristics, streaming strategies, video streaming sites, traffic flow

Introduction

Video streaming is one of the techniques for video distribution over the internet. It helps in lecture and news broadcast, and allows users' access, without any geographical constraints (Karki, Seenivasan, Claypool, & Kinicki, 2010; Tan & Zakhor, 1999). Streaming video is described as the process of moving "compressed images" over the internet for instant display at its points of arrival, which is the viewers' end, after few buffer seconds (Zhou, Hua & Xiao, 2007; Tech Support, 2014). However, streaming video poses challenges for users with limited bandwidth, and this often leads to unsatisfactory users' experience (Rao et al., 2011). The growing number of

video streaming sites and sophistication of web-based applications also suggest the need for video quality assurances (Nielsen, 1999). From the users' end, network performance measures, such as quality of experience (QoE), Network Performance, Overall Network Performance, should be assessed positively to certify the healthiness of the video streaming sites and, in few cases, the video applications (Maier, Feldmann, Paxson, & Allman, 2009).

Streaming strategies are identified as Non-ACK, ON-OFF cycle and Bulf TCP Transfer (Maier, Feldmann, Paxson, & Allman, 2009), and the conditions that affect video streaming quality are metrics like packet size, protocols, packet interval, TCP window size and accumulation ratio (Abdeljaouad, Rachidi, Fernandes, & Karmouch, 2010; Kim, & Ammar, 2006). In view of understanding the choice of streaming strategies to achieve good video quality, studies have always been investigating network characteristics of video streaming sites. Video streaming services are judged based on the perceptual experiences and aesthetics derived by the user (Khalil, Laghari, Molina, & Palau, 2011; Wang, Banerjee, & Jamin, 2003; Nielsen, 1999). The key to information-driven network service support lies in the understanding of the network characteristics. In view of this, this study aims to investigate the traffic flow, as network characteristic, of YouTube and Vimeo –being the mostly accessible video streaming sites in Malaysia. The objectives of this study is to (a) identify their streaming strategies, and (b) compare their streaming qualities based on highlighted metrics. The second section of this study presents the review of related literatures, third section outlines the experimental methodology employed by the study; fourth section discusses the findings; and the last section concludes.

Review of literature

Video Streaming Applications

Video streaming is the process of real time delivery of video clips to users through designated web-based sites and/or mobile phone applications (Saxena, Sharan, & Fahmy, 2008; Maier, Feldmann, Paxson, & Allman, 2009). There are, at least, 44 video streaming sites (Apostolopoulos, Tan, & Wee, 2002), with varying users' control mechanism, but under two categorizations. First, these are video streaming sites where users can create, distribute and manage videos to be viewed by other users. YouTube and Vimeo are examples in this category. Second, these are commercial video streaming sites where users pay to watch interested videos. Examples of this category are Clip Blast and NetFlix.

YouTube is listed as the most famous website for video streaming with thousands of videos on all ranges of topics. It supports all video formats, with customized video replies and pages. However, YouTube

allows excess spam comments and its videos are not exclusively compatible with blog. Vimeo is another popular streaming application with sophisticated video sharing. Metacafe, Hulu, Veoh, ABC.com, ArchaeologyChannel, BigContact, BroadbandSports, Clipshack, CollegeHumor, Comcast Xfinity, Crackle, Current.com, amongst others, are examples of video streaming sites, though with different content themes and users' policies. YouTube and Vimeo are the mostly accessible and watched in Malaysia unlike NetFlix, Clipblast or Hulu, and this places them as the only video streaming sites that can be used in the experimentation seeking to identify video streaming strategies.

Concepts and Architecture of Video Streaming Applications

The concept and architecture of video streaming applications are the operating condition, such as point-to-point, multicast (or broadcast), the pre-encoded (stored) and the real time streaming (Apostolopoulos, Tan, & Wee, 2002). It also describes the characteristics of the packet travelling mode, and this is to show whether a video streaming site support constant-bit-rate (CBR), variable-bit-rate (VBR), packet-switched or circuit-switched. These characteristics in the video streaming process are discussed under the following sub-sections.

Point-to-point, multicast, and broadcast communications

Point-to-point communication, which is also called one-to-one, as a concept in video communication, shows whether or not there is a back channel between the sender and the recipient. The recipient can give a feedback through such back channel if it exists, and if it does not, the sender is faced with limited information. Multicast, as another form of communication, has neither point-to-point nor broadcast communication. It lies in between the two architectures. Internet Protocol (IP) protocol multicast, as an instance of multicast, can be seen in the application layer through an overlay networks. One-to-many communication, also called broadcast communication, on the other hand, is probably the most common video communication architecture. It is ubiquitous and provides an effective content distribution (Wang, Banerjee, & Jamin, 2003).

Streaming Media over Transmission Control Protocol (TCP)

TCP at its successful launch provided services such as file transfer, electronic mail, remote logon, and proved to be a streaming media with stability, scalability and option kind field of one octet followed by length-2 octets of option data (IETF, 2012). The Additive Increase Multiple Decrease (AIMD) rule, which is also a product of TCP, provides seamless transport of streaming media (Apostolopoulos, Tan, & Wee, 2002). In a specific context

that affects video streaming, Hyper Text Transfer Protocol (HTTP) is suitable for multimedia content storage and delivery, and Adobe's Flash video (FLV) is the most appropriate container. In addition, Adobe download technology utilizes TCP in liberating video codec from packet loss and missing frame. This shows that additional software and plugins will not be needed for video streaming function. TCP is significantly different from User Datagram Protocol (UDP) which does not require synchronization between client and video streaming (Wang, Banerjee, & Jamin, 2003; McCanne, Jacobson, & Vetterli, 1996).

Streaming Media over Rate-Controlled UDP

Internet streaming system works better as a service delivery in the form of User Datagram Protocol (UDP). UDP is flexible in terms of error and rate control. The other control mechanisms are either included or substituted, and not solely based on retransmission. The departure of TCP from the AIMD rule, for rate control, presents different immediate throughput. (Rawat, 2013; Apostolopoulos, Tan, & Wee, 2002).

Video Streaming Architecture

The video streaming architecture is discussed under the following headings: video compression, application layer Quality of Service (QoS) control, continuous media distribution services, media synchronization mechanism, and protocols for streaming media (Khalil, Laghari, Molina, & Palau, 2011).

Video Compression

Raw video should be compressed before being transmitted for efficiency to be achieved. Video compression schemes are categorized into scalable and non-scalable video coding. The scalable video coding scheme, as signifies as the focus of this study, provides a video capability that keeps bandwidth fluctuations manageable (Wang, & Schulzrinne, 1999).

Application-layer QoS control.

The application layer contains of both the congestion and error control functions. It is always recommended in view of adapting to different network conditions and the presentation quality based on user request (Zhang et al., 2001; Eleftheriadis & Anastassiou, 1995; Tan & Zakhor, 1999; Wang & Schulzrinne, 1999). The congestion control prevents loss of packet and reduces delay, while the quality of the video presentation is improved by the error control function (Wang, & Schulzrinne, 1999).

Continuous media distribution services

Continuous media distribution services involves network altering, multicasts and content replications at the application level. Services are rendered through streaming servers which process multimedia data and offers interactive operations such as pause, play, fast, rewind, and forward. Continuous media distribution services have the ability of attaining QoS and video/audio streaming efficiency over the best export internet (ITU-T, 1996).

Media synchronization mechanisms

This is the main object that differentiates conventional data applications from multimedia applications. The receiver-side application, when media synchronization mechanism is utilized, gives numerous but similar media streams (ITU-T, 1998). An example to illustrate such is lips movement of a speaker being equated with the audio heard.

Protocols for streaming media

The development and standardization of these protocols are for streaming and client-server communications. Services such as network addressing, transport and session control are rendered by the streaming media protocols. The categories are network-layer (also known as Internet Protocol (IP)), transport (also known as User Datagram Protocol (UDP)), and session control (also known as real-time streaming protocol (RTSP)) protocols (Wang, Banerjee, & Jamin, 2003).

Tools in Measuring Traffic Flow

Network traffic monitoring and analysis tools are used in the detection of network failure, and detection of security threat, and therefore aids network maintenance, administration and managerial decision making (So-In, 2006). Simple Network Management Protocol (SNMP), Sniffing, Windows Management Instrumentation (WMI) and Network flow monitoring and analysis are the principal tasks in network monitoring and analysis (Rao et al., 2011). NetFlow, SNMP's network traffic information and packet sniffers are the broad categories of network tool (So-In, 2006).

NetFlow is chaired by Cisco NetFlow as an open but proprietary Internetworking Operating System (IOS). It captures network information and subsequently sends it to a monitoring host. NetFlow information presents the source and destination addresses which are used by the network administrator to identify the generator and recipients of the traffic. FlowScan, Autofocus and Fluxoscope are examples of tools under this category (So-In, 2006). SNMP's network traffic information is supported by the application layer protocol, and functions as an agent model using Management Information Base (MIB) and a set of commands to exchange

information (Zhou, Hua & Xiao, 2007). It allows retrieval of information from network devices without storing the flow and packet information. Multi Router Traffic Grapher and Cricket are examples of tools under this category (So-In, 2006). Packet Sniffer, on the other hand, collects network flow instead of requesting as experienced under SNMP. A sniffer is either hardware or software which functions as local traffic interceptor and collector (So-In, 2006). Snoop, tcpdump and Wireshack are examples of Packet Sniffer software.

WireShark, formerly known as Ethereal, is an open source packet sniffer with a user-friendly interface. It aids sorting and filtering of network parameters and supports packet capturing from both live and captured networks. In network monitoring and analysis, using WireShark, the traffic information is fed as protocols, layer 2, layer 3, BGP4, and real time data (Maier, Feldmann, Paxson, & Allman, 2009; Saxena, Sharan, & Fahmy, 2008). Sniffers, unlike NetFlow and sFlow, do not present distinct characteristics of traffic pattern and applications (Rao et al., 2011). The comparison between the network information tools and techniques to present the preference of Sniffers to others are presented in Table 1 below.

Table 1: Comparison of Network Flow Information Tools and Techniques

	Sniffer	RMON (IV)	RMON (II)	NetFlow	sFlow
Packet Capture	Y	N	Y	N	P
Interface Counters	Y	P	P	N	Y
Protocols:					
Packet headers	Y	N	P	N	Y
Ethernet/802.3	Y	N	Y	N	Y
IP/ICMP/UDP/TCP	Y	N	Y	Y	Y
IPX	Y	N	Y	N	Y
Appletalk	Y	N	Y	N	Y
Layer 2:					
Input / Output Interface	Y	N	N	Y	Y
Input / Output Priority	Y	N	N	N	Y
Input / Output VLAN	Y	N	N	N	Y
Layer 3:					
Source subnet/ Prefix	Y	N	N	Y	Y
Destination subnet / Prefix	Y	N	N	Y	Y
Next hop	N	N	N	Y	Y
BGP4					
Source AS	N	N	N	P	Y
Destination AS	N	N	N	P	Y
Destination Peer AS	N	N	N	P	Y
Communities	N	N	N	N	N
AS Path	N	N	N	N	N
Real Time Data Collection	Y	Y	Y	P	Y
Configuration without SNMP	N	N	N	Y	Y
Configuration via SNMP	N	Y	Y	N	Y
Low Cost	Y	Y	N	N	Y
Scalable	N	P	N	N	Y
Wire-Speed	Y	Y	P	P	P

[Key: N = features NOT supported, P = features PARTIALLY supported, and Y = features FULLY supported]

Methodology

The experiment is conducted using YouTube and Vimeo as the two different video streaming sites. These sites are the mostly accessible sites through Malaysian IP addresses. HTML5, Flash and HD are the three container sets for the videos streamed. Also, both Wired and Wireless network environments. In characterizing the network of the video streaming sites, Protocols, packet size, packet interval, download size and accumulation ratio are used, and WireShark is employed as the network analyzer. The stages in the experimental methodology are (a) dataset creation and labelling, (b) tool gathering, (c) measurement technique, and (d) comparative study and reporting (Rawat, 2013).

Dataset Creation and Labelling

Six (6) different dataset are created and labelled under this section, comprising of three (3) datasets for each of the YouTube and Vimeo videos. *YouTubeHD*, *YouTubeHTML5*, and *YouTubeFLASH* are the labels for the YouTube videos of all the three container sets. *VimHD*, *VimHTML5*, and *VimFLASH* are the labels for the Vimeo video, respectively. The same 1.46minutes length video is converted into the HD, HTML5 and Flash video format using a trial version of Videoconverter.com. The length of the video is maintained across board, using time as a control variable. The difference in the container allows comparative analysis of the influence of container in the video streaming quality (Rao et al., 2011).

Tools Gathering

The tools for the experimentation study in this work are (a) Google Chrome and Firefox as mainstream web browsers (Rawat, 2013; GoogleChrome.Com), (b) WireShark (WireShark.Com), and (c) a PC machine of Windows Operating System (Micosoft.Com). WireShark is used for the capturing of the packet exchanges between the video application streaming servers and web browser. It is a software packet sniffer that supports packet headers, IP/ICMP/UDP/TCP, IPX and Ethernet/802.3.

Measurement Technique

The two browsers used, as earlier hinted, for the video streaming were Mozilla Firefox 4.0 and Google Chrome 10.0. Window 8 PC is employed with both wired and wireless internet connection in investigating the network characteristic of the streaming sites. The browsers were able to play all the video formats after successful installation of necessary plugins. YouTube uses webM as default codec for HTML5 video streaming, and Chrome and Firefox possess inbuilt feature for the same purpose. The wired internet service is accessed through a RPG cable connected to a Router

socket, and the wireless connection has a download rate of 1.05Mbps and upload rate of 0.75Mbps.

Metrics Used

The network traffic of the video streaming applications are measured through the following metrics: protocols, packet size, packet interval, accumulation ratio and TCP Window Size. The metrics are discussed under their respective sub headings.

Protocol

Transmission Control Protocol (TCP) is the protocol used to investigate the video streaming because data transport among nodes in the network is the focus of the study. TCP has the best IP suite.

Packet Size

The size of formatted video stream being transferred over the internet. It is measured in Bytes.

Packet Interval

This is the time difference in the arrival times of successive packets. It defined the traffic of the packet, and measured in seconds (s).

Accumulation Ratio

This is the rate at which the video encodes and defined the conversion rate of the video from the original to viewable format. It is measured in Megabytes per second (Mbps).

TCP Window Size

TCP window size is also called TCP window scale option. It is the TCP-allowed size that influences the increase in the window size. It is measured in KiloBytes (KB).

Comparative Study and Report Writing

A detailed description of the strategies used for the video application streaming is presented. Tabular representations of the experimental data, and the result of the investigations based on each of the streamed video clip under the respective web browser are also presented. Based on this, findings are reported using graphical representations, and discussions on the comparisons of the network characteristics of the traffics of the video applications are presented.

Experimental results and findings
Data Collected with the Wired Network

Two (2) different network facilities; wired and the wireless environments, are used for the experiment. Table 2 presents the findings from the Wired Network.

Table 2: Findings from the Wired Network

Service		YouTube			Vimeo		
Container		Flash	HTML5	HD	Flash	HTML5	HD
Mozilla	Time (Packet Interval) (Seconds)	0.63	0.65	0.45	3.60	0.80	3.30
	Download Size (Bytes)	58.00	58.00	72.00	62.00	58.00	58.00
	Encoding Rate (Bytes/seconds)	92.06	89.02	160.00	17.22	72.50	17.57
	TCP Window Size (KiloBytes)	7.84	8.66	6.74	8.45	5.35	0.20
Google Chrome	Time (Packet Interval) (Seconds)	1.69	1.88	0.23	2.64	9.25	1.93
	Download Size (Bytes)	68.00	66.00	66.00	66.00	68.00	66.00
	Encoding Rate (Bytes/seconds)	40.23	35.10	286.95	25.00	7.02	34.19
	TCP Window Size (KiloBytes)	5.24	18.75	12.74	18.85	6.22	20.24

Wired/Mozilla

The download size data in a wired network with Mozilla as the browser for each of the video container is presented in Table 3

Table 3: Download size data under a wired internet network with Mozilla as the browser

Download Size				
		YouTube		Vimeo
	Download Size (B)	Time(s)	Download Size(B)	Time(s)
Flash	58	0.63	62	3.6
HTML5	58	0.65	58	0.8
HD	72	0.45	58	3.3

Figure 1 below presents the graph of the rate of Download size data under a wired internet network with Mozilla as the browser

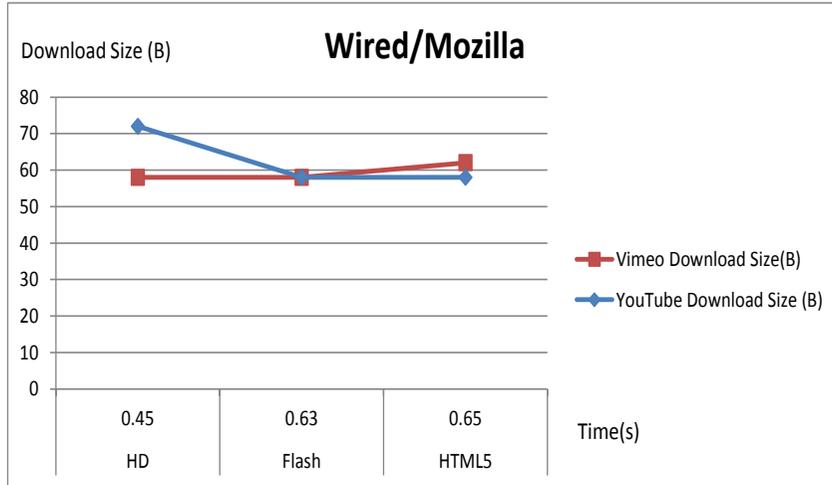


Figure 1: Download size data under a wired internet network with Mozilla as the browser
 Table 4 presents the TCP Window Size data under a wired internet network with Mozilla as the browser.

Table 4: TCP Window Size data under a wired internet network with Mozilla as the browser

TCP Window Size						
	TCP Window Size (kB)	Time(s)	TCP Window Size (kB)	Time(s)		
Flash	7.84	0.63	8.45	3.6		
HTML5	8.66	0.65	5.35	0.8		
HD	6.74	0.45	0.2	3.3		

Figure 2 presents the graph of the rate TCP Window size data in a wired internet network, and Mozilla as the browser.

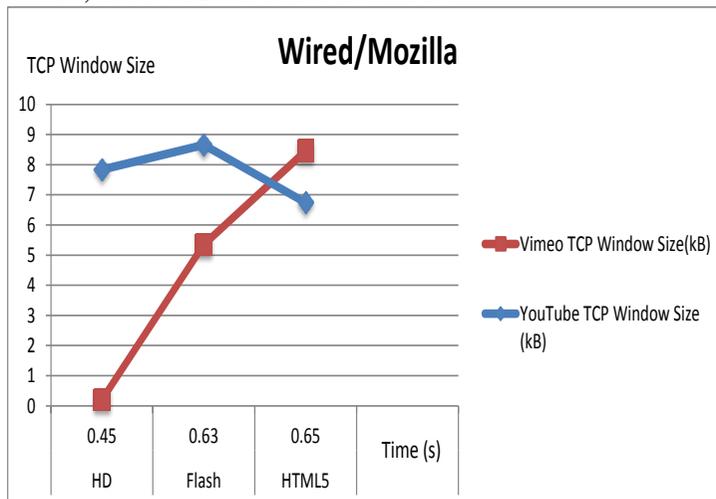


Figure 2: The rate of TCP Window size data in a wired internet network and Mozilla as the browser

Table 5 presents the Encoding Rate data in a wired internet network with Mozilla as the browser.

Table 5: Encoding Rate data in a wired internet network with Mozilla as the browser

Encoding Rate				
	Encoding Rate (B/s)	Time (s)	Encoding Rate (B/s)	Time(s)
Flash	92.06	0.63	17.22	3.6
HTML5	89.02	0.65	72.5	0.8
HD	160	0.45	17.57	3.3

Figure 3 below presents the graph of Encoding Rate data in a wired internet network with Mozilla as the browser

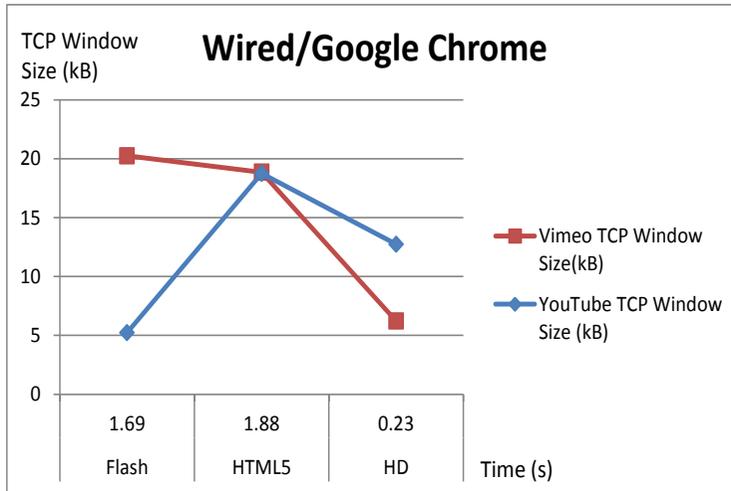


Figure 3: Encoding Rate data with a wired internet network, and Mozilla as the browser

Wired/Google Chrome

Table 6 presents the rate of Download Size data in a wired internet network with Google Chrome as the browser

Table 6: The rate of Download Size data in a wired internet network, and Google Chrome as the browser

Download Size				
	Download Size (B)	YouTube Time(s)	Download Size(B)	Vimeo Time(s)
Flash	68	1.69	66	2.64
HTML5	66	1.88	68	9.25
HD	66	0.23	66	1.93

Figure 4 below presents the graph of the rate of Download Size data in a wired internet network with Google Chrome as the browser

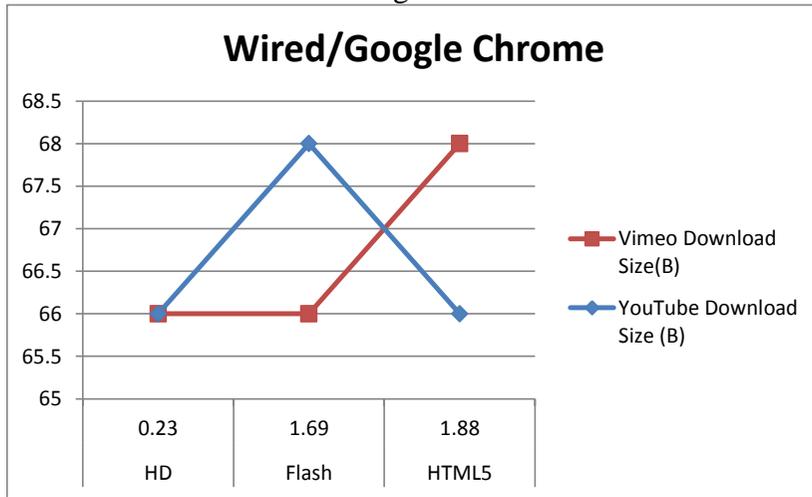


Figure 4: The rate of Download Size data in a wired internet network with Google Chrome as the browser

Table 7 presents the rate of the TCP Window Size data in a wired internet network and Google Chrome as the browser

Table 7: The rate of TCP Window Size data in a wired internet network with Google Chrome as the browser

TCP Window Size					
	TCP Window Size (kB)	Time(s)	TCP Window Size(kB)	Time(s)	
Flash	5.24	1.69	18.85	2.64	
HTML5	18.75	1.88	6.22	9.25	
HD	12.74	0.23	20.24	1.93	

Figure 5 below presents the graph of TCP Window Size data under a wired internet network with Google Chrome as the browser

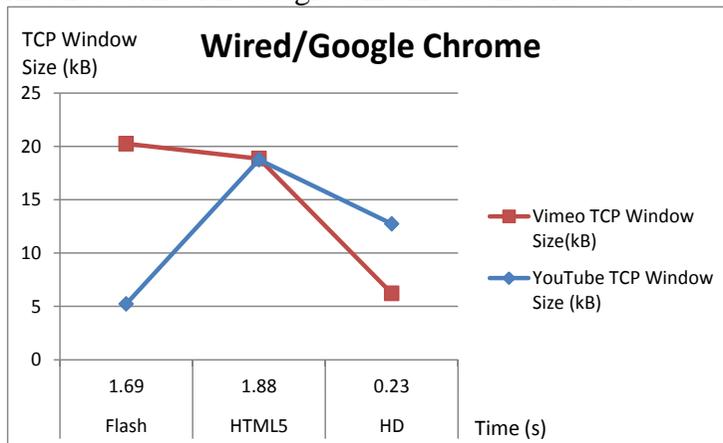


Figure 5: The rate of TCP Window Size data in a wired internet network with Google Chrome as the browser.

Table 8 presents the Encoding Rate data in a wired internet with Google Chrome as the browser.

Table 8: Encoding Rate data in a wired internet environment with Google Chrome as the browser

Encoding Rate

	Encoding Rate (B/s)	Time(s)	Encoding Rate (B/s)	Time(s)
Flash	40.23	1.69	25	2.64
HTML5	35.1	1.88	7.02	9.25
HD	286.95	0.23	34.19	1.93

Figure 6 presents the graph of the Encoding Rate data in a wired internet network with Google Chrome as the browser.

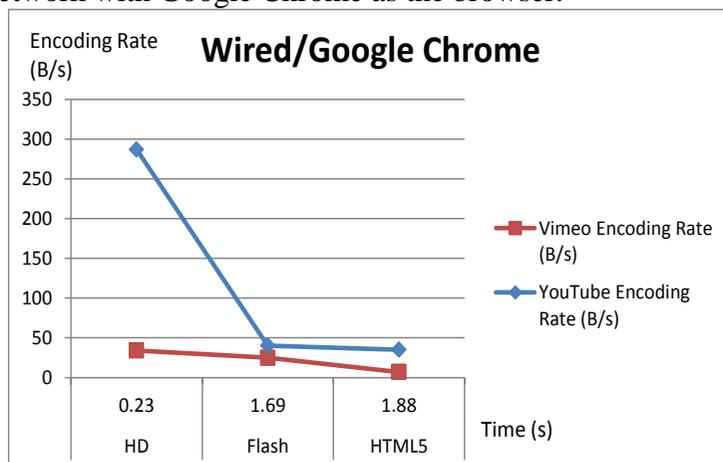


Figure 6: Encoding Rate data under a wired internet network with Google Chrome as the browser

On the other hand, Table 9 presents the overall findings from the Wireless Network

Table 9: Findings from the Wireless Network

		YouTube			Vimeo		
		Flash	HTML5	HD	Flash	HTML5	HD
Mozilla	Time (Packet Interval) (Seconds)	0.82	0.70	0.56	5.80	1.51	3.30
	Download Size (Bytes)	54.00	54.00	66.00	54.00	54.00	54.00
	Encoding Rate (Bytes/seconds)	65.65	77.14	117.85	9.31	33.54	16.36
	TCP Window Size (KiloBytes)	5.84	9.92	5.84	6.50	3.75	0.25
Google Chrome	Time (Packet Interval) (Seconds)	2.19	2.02	0.34	3.50	11.00	2.56
	Download Size (Bytes)	66.00	54.00	66.00	66.00	66.00	54.00
	Encoding Rate (Bytes/seconds)	30.13	26.73	194.11	194.11	6.00	21.09
	TCP Window Size (KiloBytes)	4.28	24.80	8.70	18.85	5.03	17.30

Wireless/Mozilla

Table 10 presents the rate of Download size data in a wireless internet network, and Mozilla as the browser

Table 10: Download size data under a wireless internet network with Mozilla as the browser

Download Size

	YouTube		Vimeo	
	Download Size (B)	Time(s)	Download Size (B)	Time(s)
Flash	54	0.82	54	5.8
HTML5	54	0.7	54	1.51
HD	66	0.56	54	3.3

Figure 7 below presents the graphical representation of Download Size under a wireless internet network with Mozilla as the browser

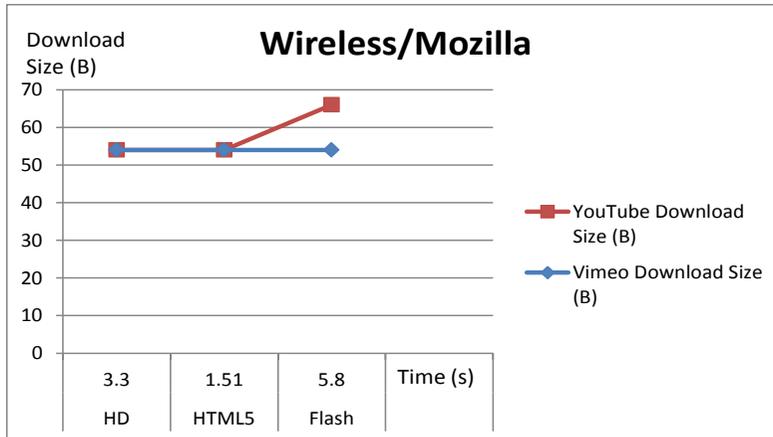


Figure 7: Download Size under a wireless internet network with Mozilla as the browser

Table 11 presents the rate of TCP Window Size data in a wireless internet environment, with Mozilla as the browser.

Table 11: TCP Window Size data under a wireless internet network with Mozilla as the browser

	TCP Window Size (kB)	Time(s)	TCP Window Size (kB)	Time(s)
Flash	5.84	0.82	6.5	5.8
HTML5	9.92	0.7	3.75	1.51
HD	5.84	0.56	0.25	3.3

Figure 8 below presents the graph of TCP Window Size in a wireless internet network with Mozilla as the browser.

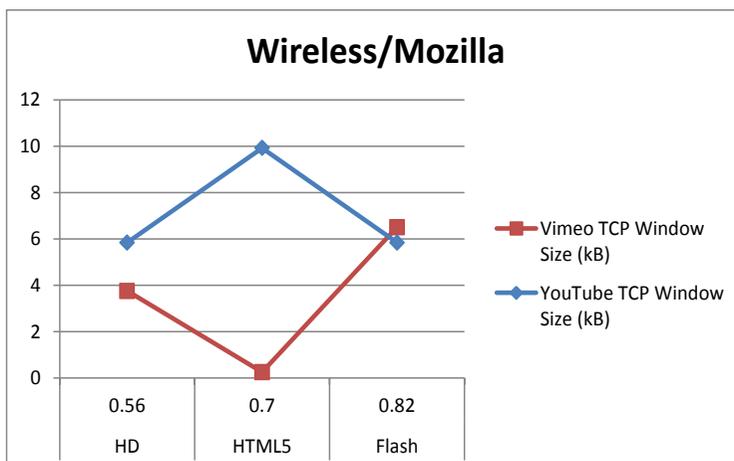


Figure 8: TCP Window Size in a wireless internet environment with Mozilla as the browser

Table 12 presents the Encoding rate data in a wireless internet network with Mozilla as the browser

Table 12: Encoding rate data under a wireless internet network with Mozilla as the browser

Encoding Rate

	Encoding Rate (B/s)	Time(s)	Encoding Rate (B/s)	Time(s)
Flash	66.65	0.82	9.31	5.8
HTML5	77.14	0.7	33.54	1.51
HD	117.85	0.56	16.36	3.3

Figure 9 below presents the graph of Encoding rate in a wireless internet environment with Mozilla as the browser.

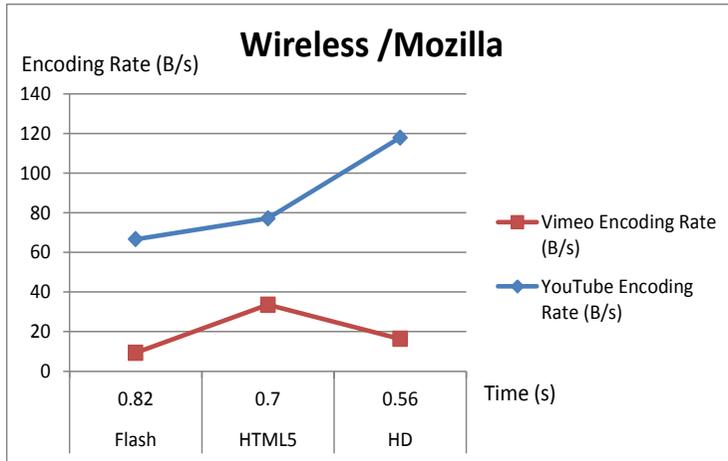


Figure 9: Encoding rate in a wireless internet environment with Mozilla as the browser **Wireless/Google Chrome**

Table 13 presents the rate of Download Size data in a wireless internet environment with Google Chrome as the browser

Table 13: Download Size data in a wireless internet environment with Google Chrome

Download Size

	Download Size (B)	Time(s)	Download Size (B)	Time(s)
Flash	66	2.19	66	3.5
HTML5	54	2.02	66	11
HD	66	0.34	54	2.56

Figure 10 presents the graph of the rate of Download size in a wireless internet network with Google Chrome as the browser.

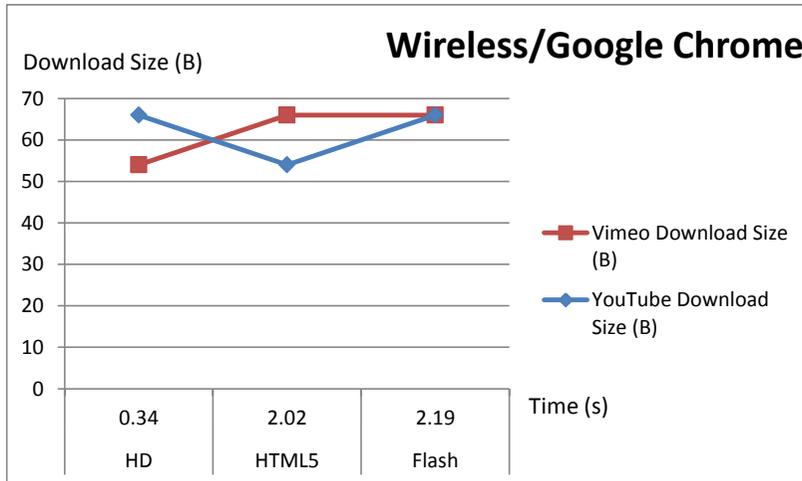


Figure 10: Download size under a wireless internet network with Google Chrome as the browser.

Table 14 presents the rate of TCP Window Size data in a wireless internet network with Google Chrome as the browser

Table 14: Rate of TCP Window Size data in a wireless internet environment with Google Chrome as the browser

TCP Window Size				
	TCP Window Size (kB)	Time(s)	TCP Window Size (kB)	Time(s)
Flash	4.28	2.19	18.85	3.5
HTML5	24.8	2.02	5.03	11
HD	8.7	0.34	17.3	2.56

Figure 11 presents the graph of TCP Window Size in a wireless internet environment with Google Chrome as the browser.

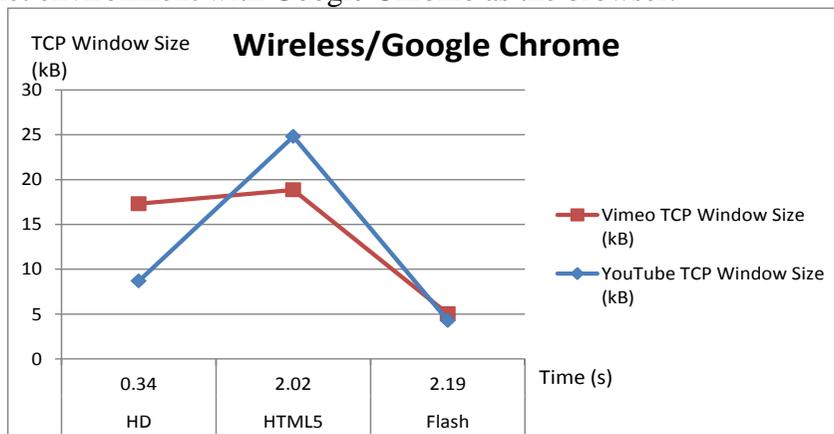


Figure 11: TCP Window Size under a wireless internet network with Google Chrome as the browser.

Table 15 presents the Encoding Rate data in a wireless internet network when the browser is Google Chrome

Table 15: Encoding Rate data in a wireless internet network, and Google Chrome as the browser

Encoding Rate				
	Encoding Rate (B/s)	Time(s)	Encoding Rate (B/s)	Time(s)
Flash	30.13	2.19	194.11	3.5
HTML5	26.73	2.02	6	11
HD	194.11	0.34	21.09	2.56

Figure 12 presents the graph of Encoding Rate in a wireless internet environment with Google Chrome.

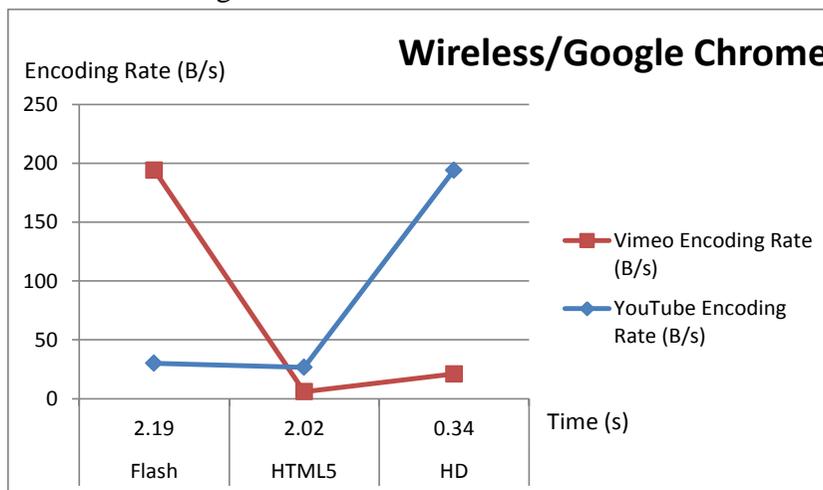


Figure 12: Encoding Rate in a wireless internet network with Google Chrome.

The streaming strategies for each of the pairs of web browser (Mozilla Firefox and Google Chrome) and video container (Flash, HTML5, and HD) for the streaming sites (YouTube and Vimeo, as observed in the experiment, is presented in Table 16.

Table 16: Streaming Strategies

Short represents Short ON-OFF, Long for Long ON-OFF, and No for No ON-OFF cycles

	YouTube			Vimeo		
	Flash	HTML5	HD	Flash	HTML5	HD
Mozilla Firefox	Short	No	No	Long	Long	No
Google Chrome	Short	Long	No	Short	Long	Short

Discussion and conclusion

The network characteristics of video streaming applications' traffic, as experimented with YouTube and Vimeo as streaming sites, Mozilla Firefox and Google Chrome as web browsers, and Flash, HTML5, and HD

as video containers are presented. An in-depth network analysis of video streaming sites acknowledges the video container of the video clip, the browser, and the streaming protocol (Wisitpongphan, & Peha, 2003). In this study, a dual experimental setups is employed where the streaming sites remains constant in each of the respective setup, but with difference container and browser. This suggests that container and browser are the contributing factors in the experiments and thus influence the observed streaming strategies.

In a typical streaming session, the video content has two phases. These are buffering and steady phases. The buffering phase has data transfer feature which is limited to the availability of the end-to-end bandwidth. The steady phase, however, has an average download rate larger than the encoding rate which is also called the accumulation ratio. The streaming strategies found, in overall, are Short ON-OFF, Long ON-OFF, and No ON-OFF cycles. The following sub-sections discuss the respective streaming strategies for the streaming sites.

Streaming strategies of YouTube

Short ON-OFF cycle is observed in the streaming of flash container across both browsers. Also, irrespective of the browser, Long ON-OFF and No ON-OFF are used for streaming HTML5 and HD containers. During the buffering phase of *YouTubeFlash* datasets, the download size and encoding rate as observed as being responsible for the experienced playback data. There is no observed relationship between the encoding rates and download amount for the HTML5 videos. This is, arguably, caused by the YouTube engineering strategy in the flash video streaming (Wisitpongphan, & Peha, 2003). The download amount is however related to the encoding rate at the steady state phase. In Google Chrome streaming, HTML5 are streamed with evenly distributed encoding rate.

Streaming strategies of Vimeo

Short ON-OFF cycles are the Vimeo streaming strategies for flash and HD videos in Google Chrome, but Long ON-OFF in Mozilla Firefox. However, irrespective of the browser, Long ON-OFF is the streaming strategy for streaming HTML5. The download size depends only on the web browser during the buffering phase, as also supported by Akhshabi, Begen, and Dovrosis, (2011), downloaded contents of related encoding rates. At the steady phase, the browser serves as the block streaming in the Long ON-OFF cycles.

The comparative analysis of the network traffic characteristics of the video streaming sites

First, web browser and video containers are factors that influence streaming processes and strategies in YouTube and Vimeo. It is only the videos of flash container that rely not on the browser because web browser does not control its rate of data transfer. YouTube has a suitable streaming strategy for HTMLs and does not have data transfer rate limit. There is, therefore, different patterns in the buffer phase involving YouTube, and not in Vimeo. Video streaming in Vimeo is directly to web browser irrespective of the video container used.

Second, there is less engineering complexities in No ON-OFF when receiving large buffer occupancy. User interruptions also have large unused bytes, and the Short and Long ON-OFF cycles support the streaming sites' application layer. The Short ON-OFF cycle has small buffer occupancy and small unused bytes. On the other hand, Long ON-OFF receives moderate buffer occupancy with moderate unused bytes on user interruptions.

Conclusion

The streaming strategies of YouTube and Vimeo, as streaming sites, are uniform with the usage of TCP as the streaming protocol. This has allowed a fair environment for experimentation and comparative analysis of the streaming strategies of both sites. In the analysis involving YouTube videos, Short ON-OFF is the streaming strategy for flash videos in both Mozilla Firefox and Google Chrome. Mozilla streaming HTML5 and HD videos, and Google Chrome streaming HD videos in YouTube have No ON-OFF cycle as streaming strategies.

On the other hand, in Vimeo, Long ON-OFF is observed when flash and HTML5 videos are streamed in Mozilla Firefox, and HTML5 in Google Chrome. The No ON-OFF is recorded only when HD videos are streamed in Mozilla Firefox, and Short ON-OFF is observed when flash and HD videos are streamed with Google Chrome. In conclusion, both video streaming sites involve all the three streaming strategies, though with varying instances which are determined by the browser used and video container.

References:

1. Abdeljaouad, H., Rachidi, S., Fernandes, & Karmouch, A. (2010). Performance analysis of modern TCP variants: A comparison of Cubic, Compound and New Reno. In *Proc. QBSC*, 2010.
2. Akhshabi, S., Begen, C., & Dovrosi, V. (2011). An experimental evaluation of rate-adaptation algorithm in adaptive streaming over http, in *MMSys '11*, 157-168

3. Apostolopoulos, J. G., Tan, W-T., & Wee, S. J. (2002). Video Streaming: Concepts, Algorithms, and Systems, Mobile and Media Systems Laboratory, *HP Laboratories Palo Alto*.
4. Eleftheriadis, A., & Anastassiou, D. (1995). Meeting arbitrary QoS constraints using dynamic rate shaping of coded digital video, in Proc. 5th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'95), April 1995, pp. 95-106.
5. GoogleChrome.Com <https://www.google.com/intl/en/chrome/browser/>
6. IETF.ORG (2015). Transmission Control Protocol: IETF RFC number 793 (www.ietf.org/rfc.html).
7. ITU-T Recommendation P.800. *Methods for subjective determination of transmission quality*, August 1996.
8. ITU-T Recommendation P.911. *Subjective audiovisual quality assessment methods for multimedia applications*, December 1998.
9. Karki, R., Seenivasan, T., Claypool, M., & Kinicki, R. (2010). Performance analysis of home streaming video using Orb. In *Proc. NOSSDAV*, 2010.
10. Khalil, R., Laghari, N. C., Molina, B., & Palau, C. E. (2011). QoE Aware Service Delivery in Distributed Environment, Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on , pp.837-842, 22-25 March 2011.
11. Kim, T., & Ammar, M. (2006). Receiver buffer requirement for video streaming over TCP. In *Proc. SPIE*, pages p. 422–431.
12. Maier, G., Feldmann, A., Paxson, V., & Allman, M. (2009). On Dominant Characteristics of Residential Broadband Internet Traffic. In *IMC '09*, page 90-102.
13. McCanne, S., Jacobson, V., & Vetterli, M. (1996). Receiver-driven layered multicast, in Proc. ACM SIGCOMM '96, pp. 117-130
14. Microsoft.Com <http://windows.microsoft.com/en-us/windows/home>
15. Mozilla.Org <http://www.mozilla.org/en-US/firefox/new/>
16. Nielsen, J. (1999). Video and Streaming Media. *UseIt.com*. Retrieved February 23, 2017 from the World Wide Web: <http://www.useit.com/alertbox/990808.html>
17. Rao, A., Lim, Y-S., Barakat, C., Legout, A., Towsley, D., & Dabbous, W. (2011). Network Characteristics of Video Streaming, ACM CoNEXT 2011, Tokyo, Japan
18. Rawat, D. B. (2013). *Computer and Network Security: An Experimental Approach*. UK, CreateSpace Independent Publishing Platform; 1st Edition.

19. Saxena, M., Sharan, U., & Fahmy, S. (2008). Analyzing video services in web 2.0: a global perspective. In *NOSSDAV '08*, page 39-44
20. So-In, C. (2006). A Survey of Network Traffic Monitoring and Analysis Tools, Retrieved from http://www1.cse.wustl.edu/~jain/cse567-06/ftp/net_traffic_monitors3/index.html on 28th, Feb., 2017
21. Tan, W., & Zakhor, A. (1999). Real-time Internet video using error resilient scalable compression and TCP-friendly transport protocol, *IEEE Trans. on Multimedia*, 1 (2), 172-186
22. Tech Support Planet (2014). Best Video Streaming Sites, Retrieved from <http://www.techsupportalert.com/5-Best-Free-Video-Streaming-Sites.htm> on 26th Feb., 2014.
23. Wang, X., & Schulzrinne, H. (1999). Comparison of adaptive Internet multimedia applications," *IEICE Trans. on Communications*, 82 (6), 806-818.
24. Wang, Z., Banerjee, S., & Jamin, S. (2003). Studying streaming video quality: from an application point of view. In *Proc. ACM Multimedia*, 2003.
25. WireShark.Org <http://www.wireshark.org/>
26. Wisitpongphan, N., & Peha, J. M. (2003). Effect of TCP on Self-Similarity of Network Traffic, *Proceedings of 12th IEEE International Conference on Computer Communication and Network (ICCCN)*
27. Zhang, Q., Wang, G., Zhu, W., & Zhang, Y-Q. (2001). Robust scalable video streaming over Internet with network-adaptive congestion control and unequal loss protection, submitted to Packet Video Workshop, Kyongju, Korea, April 2001.
28. Zhou, Hua & Xiao (2007). A Web-based Management System for H.264 Live Video Broadcasting, *Second International Mutlisymposium on Computer and Computational Sciences*