Towards Dynamic Network Condition-Aware Video Server Selection Algorithms over Wireless Networks

Hyunwoo Nam*, Kyung-Hwa Kim†, Doru Calin‡ and Henning Schulzrinne†
*Department of Electrical Engineering, Columbia University, New York, NY
†Department of Computer Science, Columbia University, New York, NY
‡Bell Laboratories, Alcatel-Lucent, Murray Hill, NJ

***Abstract*—We investigate video server selection algorithms in a distributed video-on-demand system. We conduct a detailed study of the YouTube Content Delivery Network (CDN) on PCs and mobile devices over Wi-Fi and 3G networks under varying network conditions. We proved that a location-aware video server selection algorithm assigns a video content server based on the network attachment point of a client. We found out that such distance-based algorithms carry the risk of directing a client to a less optimal content server, although there may exist other better performing video delivery servers. In order to solve this problem, we propose to use dynamic network information such as packet loss rates and Round Trip Time (RTT) between an edge node of an wireless network (e.g., an Internet Service Provider (ISP) router in a Wi-Fi network and a Radio Network Controller (RNC) node in a 3G network) and video content servers, to find the optimal video content server when a video is requested. Our empirical study shows that the proposed architecture can provide higher TCP performance, leading to better viewing quality compared to location-based video server selection algorithms.**

Keywords*—Video Streaming, Mobile Wireless, HTTP Progressive Video, Video Server Selection Algorithms, Over The Top Applications

I. Introduction

Today's popular video content delivery systems deploy CDNs. Video content providers such as YouTube and Netflix stream videos to clients through their own CDNs or the networks provided by third parties such as Akamai [1] and Limelight [2]. When a client requests a video, a video content provider uses video server selection algorithms in order to decide which video content server the client downloads the video from. The video selection mechanisms and policies are designed for providing high availability, server load-balancing and minimizing the cost for delivering video contents to clients [3], [4].

In this paper, we analyze the video server selection algorithm of YouTube. We conducted our experiments while playing YouTube videos on PCs and mobile devices over two wireless networks (3G and Wi-Fi) under varying network conditions. Through the measurements, we found that a client downloads a YouTube video from the same video content server regardless of the hardware specification, the operating system (OS) and the video application running on a client's device. Instead, the network attachment point of a client is considered as a key factor for the video server selection algorithm of YouTube.

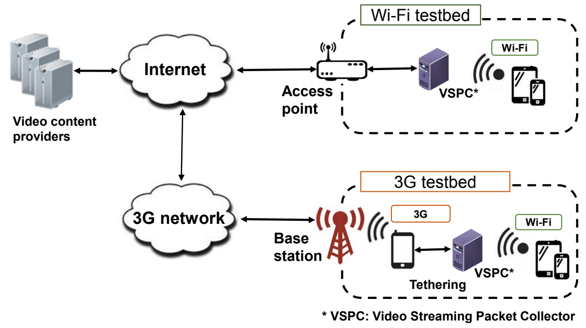

Figure 1: A testbed for analyzing video server selection algorithms while downloading videos over Wi-Fi and 3G networks

YouTube's video server selection mechanism takes into account various factors such as user proximity, server load and popularity of video content [4]. During our measurements, we discovered that there may exist other available video content servers that contain the same video content and offer higher TCP performance than the video content server assigned by YouTube at the video requested time. After carefully analyzing our measurements, we surmise that YouTube's DNS-based location awareness algorithms cause this problem. YouTube delivery cloud typically assigns a video content server which is geographically close to a client [3]–[5]. However, the network conditions between a client and a video content server can be unstable, although the video content server is located close to the client.

To find the optimal video content server, a client-based mechanism may be used to analyze the network conditions between the client and video content servers when a video is requested. As Balachandran et al. [6] studied in their paper, however, the client may not be able to effectively trace the network conditions due to the lack of direct knowledge of access networks and up-link bandwidth constraints. Niven-Jenkins et al. [7] have introduced the use cases for Application-Layer Traffic Optimization (ALTO) with CDNs. They addressed that using ALTO for video streaming services can be useful to find the optimal video content servers based not only on distance, but on other factors as well (e.g., current server load and packet loss rates). In order to deploy ALTO systems, however, network service providers need to install additional ALTO servers, and video content providers operate their own ALTO clients.

In order to solve this problem, we propose to discover the optimal video content server based on the dynamic network

	Video 1	Video 2	Video 3	Video 4
All devices	173.194.7.27	173.194.7.176	173.194.7.16	173.194.53.171

(a) On different client devices

	Video 1	Video 2	Video 3	Video 4
All browsers	173.194.7.27	173.194.7.176	173.194.7.16	173.194.53.171

(b) Using different mobile browsers

Network status	Video 1	Video 2	Video 3	Video 4
Stable and Unstable	173.194.7.27	173.194.7.176	173.194.7.16	173.194.53.171

(c) Under varying network conditions

Networks	Video 1	Video 2	Video 3	Video 4
Wi-Fi	173.194.7.27	173.194.7.176	173.194.7.16	173.194.53.171
3G (AT&T)	74.125.0.74	74.125.0.80	74.125.0.70	173.194.31.43

(d) Over different network interfaces

TABLE I: An analysis of YouTube video server selection algorithms while playing sample videos on PCs and mobile devices

conditions between a corresponding edge node of a wireless network and video content servers. The edge node can be an ISP router in a Wi-Fi network, a RNC node in a 3G network and a Packet Data Network Gateway (P-GW) in an Long Term Evolution (LTE) network. We do not require any additional implementation neither at the server nor at the client. In our proposed architecture, an edge node is designed to conduct the following two objectives:

- **Caching addresses of video content servers:** It caches addresses of video content servers while videos are delivered through the node.
- **Finding the optimal video content server:** It enables to select a preferred video content server among the cached video content servers that contain the same video content based on measured packet loss rates and current RTTs.

To show the feasibility of our approach, we have developed the Video Streaming Packet Collector (VSPC) that collects the addresses of the video content servers, captures and analyzes TCP/IP and HTTP packets while downloading YouTube videos over Wi-Fi and 3G networks (Figure 1). Our evaluation proves that a video content server chosen by our proposed dynamic network condition-aware video server selection algorithm typically provides more reliable viewing experiences with higher TCP performance than the distance-based algorithms.

The remainder of the paper is organized as follows. In the second section, we elaborate on the analysis of YouTube video server selection algorithms. In Section III, we focus on finding problems that the video server selection algorithms of YouTube may assign a video content server with unstable network conditions to a client. Our proposed solutions are described in Section IV. We evaluate our proposal in Section V and look at related work in Section VI. Finally, we summarize our conclusions in Section VII.

II. An Analysis of YouTube Video Server Selection Algorithms

In our previous work [8], we investigated the mechanism of how a YouTube video is delivered to a client. A video player running on a client's device sends an HTTP GET message that contains a video identification and information of the device. A front-end server of YouTube maps the video id to the video server name, and returns it to the client. The server name is resolved to an IP address by the client via a DNS query to a DNS server. Finally, the client sends another HTTP GET message to the video content server for downloading the video content over HTTP. According to recent studies [3]–[5], in addition to using a fixed hash-based scheme that maps a video id to a unique hostname, YouTube uses two different approaches to perform load-balancing based on the location of a client and the current load.

- **Using DNS resolution:** A logical video server name is mapped to a physical video server via DNS resolution. YouTube typically allocates multiple IP addresses to one video server name, and picks one of physical video servers that is geographically close to a client. A client may be assigned to a farther location in order to avoid high traffic load on a video content server.
- **Using dynamic HTTP redirection:** YouTube utilizes an HTTP redirection mechanism to dynamically redirect a client's access to a non-busy video content server. In this case, a video content server sends an HTTP 302 message asking to download the video from another server when a client requests a video.

Testbed setups - We focus on analyzing YouTube server selection algorithms, taking into account various conditions such as requesting videos on different devices (PCs and mobile devices), using various applications running on diverse operating systems (Windows, Mac OS X, Linux, iOS and Android) over Wi-Fi and 3G networks, under varying network conditions. Figure 1 shows our testbed setups for the analysis of YouTube video content server selection algorithms. During the measurements, we played hundreds of randomly selected videos from a diversity of genres (e.g., movie, music video, live concert and sports), popularity, length (from ten minutes to one hour) and video quality (high quality and high definition).

As a baseline analysis, we conducted the following experiments:

A. Requesting videos on different devices using the same video application over Wi-Fi networks

YouTube videos were requested on different devices at the same time and from the same place over Wi-Fi networks under the same network condition. During the measurements, we played the videos using the stand-alone application provided by YouTube on mobile devices and Chrome browsers on PCs. As shown in Table Ia, we found that the addresses of video content servers chosen by YouTube remain the same regardless of the hardware specifications and the operating systems running on clients' devices.

B. Requesting videos on the same mobile device using various applications over Wi-Fi networks

YouTube videos were requested on iPhone 4S using different video applications from the same time and place over Wi-Fi networks under the same network conditions. We requested the sample videos using various stand-alone browsers running on iOS devices such as Safari, Chrome, Mercury and Puffin. Table Ib shows that the addresses of video content servers selected by YouTube remain the same, regardless of the types of video applications running on clients' devices. We have conducted the same experiments on PCs and Android devices, and seen the same experimental results.

C. Requesting videos on the same mobile device over Wi-Fi networks under varying network conditions

We played the sample YouTube videos on iPhone 4S over a Wi-Fi network under different network conditions: *stable* and *unstable*. In order to create unreliable network conditions, we intentionally injected load in the network using a common network testing tool, Iperf. We also placed the devices that cause interference at 2.4 GHz, such as baby monitors and cordless phones between the client and the Wi-Fi access point. The RTT between the clients and the video content servers was 16.04 ms on average under stable network conditions, while it was 566.2 ms under fluctuating network conditions. As shown in Table Ic, network conditions between a client and a Wi-Fi access point do not influence the video server selection algorithms of YouTube.

D. Requesting videos on the same mobile devices via different wireless network interfaces

The sample YouTube videos were requested on two iPhone 4S devices at the same time and from the same place over Wi-Fi and 3G networks. Table Id shows that the client via a 3G network was established to a different video content server, compared to the client via a Wi-Fi network. However, it is difficult to confirm that YouTube considers the radio interfaces for the video server selection process. When we analyzed the user agent information in the HTTP GET message, we did not find any differences over Wi-Fi and 3G networks. Instead, YouTube takes account of DNS-based location awareness [4]. When a video request occurs, for example, the local DNS server operated by a network service provider asks the YouTube DNS server for the address of a video content server that the client downloads the video from. Based on the IP address of the DNS resolver operated by a network service provider, YouTube assigns a geographically close video content server to the client. Hence, the addresses of video content servers can be different, although clients accessing via Wi-Fi and 3G networks request an identical video from the same time and place. For example, we requested the sample YouTube videos on PCs and mobile devices via different network service providers (e.g., 3G networks - AT&T and Wi-Fi networks - Columbia Univ., Time Warner Cable and Verizon) around the Columbia campus. We collected in total 8,194 IP addresses of video content servers, and found that the clients accessed via different networks were assigned to different sets of video content servers for the same video contents.

E. Requesting a video on the same mobile device from the same place over Wi-Fi and 3G networks during 24 hours

A YouTube video was requested on iPhone 4S from the same place over Wi-Fi and 3G networks. We played the sample video every ten minutes for 24 hours. For each video content server, we calculated the frequency of how many times it was selected by YouTube when the client requested the video. Our experimental results indicate that there exist main video content servers that are frequently called and others that are seldom assigned to clients. In the experiments, for example, most of the video traffic (99.89% via Wi-Fi networks and 85.44% via 3G networks) came from one or two different video content servers.

Key observations - In addition to using PCs, our analysis was conducted while playing the videos on mobile devices (iOS and Android) using various mobile applications over Wi-Fi and 3G networks under varying network conditions. Table I shows the part of experimental results. We have seen the same results when we performed the identical experiments for hundreds of YouTube videos under the same conditions. Our key findings can be summarized as follows:

- The addresses of video content servers chosen by YouTube remain the same regardless of the hardware specifications, the operating systems, the video applications running on clients' devices and the network conditions between clients and wireless access points; and
- The video server selection algorithms conducted by YouTube are mostly affected by network attachment points of clients when they request videos.

III. YouTube Often Assigns Video Content Server with Unreliable Network Conditions

In this section, we point out that YouTube server selection algorithms frequently assign video content servers with long RTTs to clients although there exit other servers that delivery the same video content with shorter RTTs. Based on extensive measurements, we surmise that YouTube's location aware video server selection algorithms cause this unwanted behavior. According to other studies [3]–[5], YouTube typically assigns a video content server which is geographically near to a client. However, our measurements show that the RTT between a client and a video content server can be longer, even though the server is close to the client.

Finding locations of YouTube video content servers - In order to prove this, we analyzed the geographical locations of YouTube video content servers. We first used the IP-to-location database [9]. The experimental results indicate that most of the collected YouTube video content servers (more than 97%) were located in Mountain View, CA. However, as Torres et al. [4] discussed in their paper, it is inaccurate to find video server locations in the large corporate networks such as YouTube and Netflix using the IP-to-location database. That is because they may hide the real locations of internal IPs for security reasons. To prove this, they measured the RTTs to the video content servers of YouTube from several ISPs, and showed that there was a lot of variation, even though the database reported that the servers were located in the same place - Mountain View, CA. Adhikari et al. [10] addressed that YouTube video content servers are distributed over more than 45 cities in 25 different countries around the world.

To avoid this problem, we used `traceroute` to estimate the locations of the last hop routers between the clients and the YouTube network. We found that in most of cases there was not much difference (less than 2 ms) between the RTTs to the last hop router and to the video content server from the client. Therefore, it is reasonable to assume that the locations of the actual YouTube's video content servers are geographically close to the last hop routers. During the experiments,

TABLE II: A hundred of YouTube sample videos were requested over Wi-Fi and 3G networks from different places during busy hours (13:00-15:00 and 19:00-20:00)

	Ratio (%) of being assigned to a non-preferred video server by YouTube			
Networks	Loc. 1	Loc. 2	Loc. 3	Loc. 4
Wi-Fi	97.87 %	97.75 %	96.62 %	70.37 %
3G	44.73 %	33.69 %	73.75 %	47.6 %

we collected the addresses of video content servers while requesting a thousand of videos from the four selected places in Manhattan, New York: Columbia campus (Loc. 1), residential area (Loc. 2), Times Square (Loc. 3) and Penn. Station (Loc. 4). In the experiments, we found that most of the last hop routers of video content servers assigned by YouTube were located in New York (68.51%) and California (17.02%), and some routers were placed in Michigan (5.17%), Georgia (4.19%), Massachusetts (3.01%) and Florida (2.1%).

Measuring RTTs between video content servers and clients - In order to examine the network conditions between clients and video content servers assigned by YouTube, we measured RTTs to video content servers from clients when they requested YouTube videos. We selected a hundred sample videos, and conducted the following experiments.

- Collecting addresses of video content servers: For each video, we first obtained IP addresses of video content servers that contain the same video content. The IP addresses were collected while we requested the video every five minutes for three days on PCs from the selected places. We observed four or five unique IP addresses of video content servers for each video. They are mostly located in NY and CA.
- Comparing RTTs: We measured the RTT between the client and the video content server assigned by YouTube. From the same time and place, we compared it with the RTTs between the client and other collected servers that contain the same video content at the video requested time.

We define two terms to identify video content servers: *preferred* and *non-preferred*. The preferred video content server is a server that shows the shortest RTT among others that can deliver the same video content at the video requested time. The non-preferred video content servers are others, not defined as a preferred. We calculated the ratio of how many times YouTube provided a non-preferred video content server out of the total number of requests. Table II shows the experimental results. The ratio proves that YouTube frequently assigns a non-preferred video content server to a client. During the measurements, the average and standard deviation of the RTT were 13 ms and 9.42 ms for Wi-Fi networks while they were 63.55 ms and 21.5 ms for 3G networks.

Video content servers with long RTTs to clients degrade video QoE - After carefully analyzing the measurements, we conjecture that YouTube's distance-aware server selection algorithms result in this high ratio of being assigned to non-preferred video content servers. One may assume that this behavior is caused by YouTube's server load balancing policies. If this assumption is true, we would have observed different video content servers changing over time in our measurements. However, we found out that the IP addresses of video content servers collected during busy hours were the almost same with the IP addresses that were assigned during non-busy hours. The last hop routers of non-preferred video content servers were located near NY. This proves that YouTube's server selection algorithms more take into account user proximity rather than server load.

We note that this unwanted behavior may degrade video QoE. For example, Eq. 1 represents the TCP average throughput in terms of packet loss and RTT [11]. TCP flows with shorter RTTs gain a congestion window (CWND) advantage in the slow start phase. When a loss occurs, for example, the slow start begins from its initial CWND. With a short RTT, the CWND reaches the slow start threshold faster than a TCP flow with a longer RTT. Therefore, a client may often experience buffer underflow if the RTT is long and the network conditions are fluctuating; playing a video has to be paused until a certain amount of video content is stored in the video playout buffer.

$$TCP_{avg.\ thr} = \frac{1.22}{\sqrt{P_{loss}}} * \frac{MSS}{RTT} \tag{1}$$

Consequently, a video content server located distant from a client may provide higher speed of delivery if a video content server geographically close to a client experiences network congestion. We will elaborate on the experimental results in Section V.

IV. Dynamic Network Condition-Aware Video Server Selection Algorithms

We propose to use a corresponding edge node of a wireless network in order to assist a client connecting to the optimal video content server when a video is requested. The edge node can be an ISP router in a Wi-Fi network, a RNC node in a 3G network and a P-GW in an LTE network. The key idea is to take account of dynamic network information between an edge node and video content servers for video server selection algorithms. The overall procedures are organized into two parts: *a*) Caching addresses of video content servers when clients download videos through the edge node; *b*) Discovering the optimal video content server based on measured packet loss rates and RTTs between the edge node and the analyzed video content servers. The RTTs include the propagation delay, the queuing delay, the transmission delay and any time spent at the video content server.

A. *Caching addresses of video content servers*

An edge node records addresses of video content servers when clients watch videos through the edge node. It locally caches a hash-based database that maps the video id to the addresses of assigned video content servers. The list of video content servers can be categorized based on the video requested time, the locations of assigned video content servers and network conditions such as average TCP throughput, packet loss rates and RTTs between an edge node and video content servers measured while videos are delivered to clients.

B. *Discovering the optimal video content server*

Dynamic network information such as packet loss rates and RTTs between an edge node and collected video content servers are considered as key factors to find the optimal

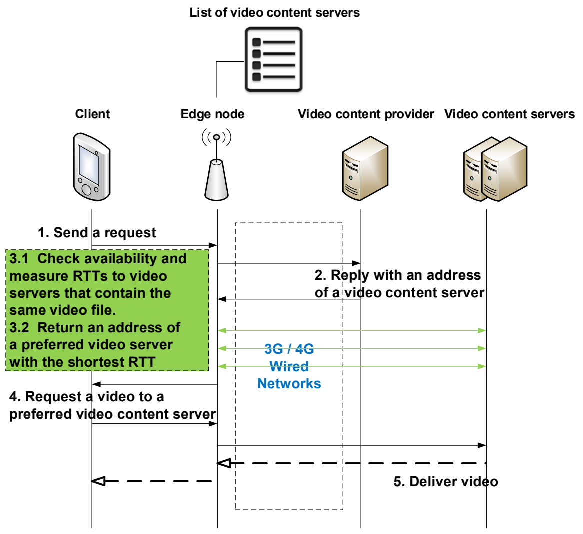
List of video content servers
Client
Edge node
Video content provider
Video content servers
1. Send a request
3.1 Check availability and measure RTTs to video servers that contain the same video file.
3.2 Return an address of a preferred video server with the shortest RTT
2. Reply with an address of a video content server
3G / 4G Wired Networks
4. Request a video to a preferred video content server
5. Deliver video

Figure 2: Selecting the optimal video content server at a corresponding edge node

video content server. For example, Figure 2 is a simplified mobile video streaming for our proposed video server selection process.

1) When a video is requested, a front-end server returns the address of a selected video content server using its own server selection algorithms (step 1 and 2).
2) Before handing the address over to the client, the edge node examines if other video content servers cached on the list are able to provide faster video streaming (step 3). It first searches a group of video content servers on the list that contain the requested video content. Secondly, it chooses a set of video content servers in the group that were recently used and showed a low packet loss rate in the previous video sessions. Then, the edge node measures the current RTTs to the chosen video content servers, including the servers on the list and the server provided by the video content provider.
3) Finally, it returns an address of a preferred video content server that shows the shortest RTT from the edge node.

Our proposed method causes a slightly longer start-up latency for a client to start downloading a video (2.1 seconds on average in our experiments), but the experimental results show that it enables to provide higher average TCP performance while playing a video.

V. Evaluation

To show the feasibility of our proposal, we have implemented our VSPC acting as an HTTP proxy server (Figure 1). The tool manipulates HTTP headers to redirect a client's access to a preferred video content server chosen by our dynamic network condition-aware video server selection algorithms. It can also analyze TCP traffic performance while downloading videos over 3G and Wi-Fi networks. The specific experimental setups are:

- We played a hundred of sample YouTube videos (360p - video bitrate 0.5 Mb/s) on PCs and mobile devices from two different places (Columbia campus and Penn. Station) over Wi-Fi and 3G networks;

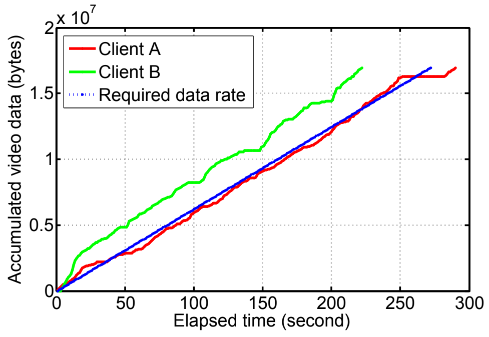

Figure 3: Experimenting with user-perceived quality while playing a sample YouTube video on two PCs over a Wi-Fi network in the HTTP proxy server-based testbed

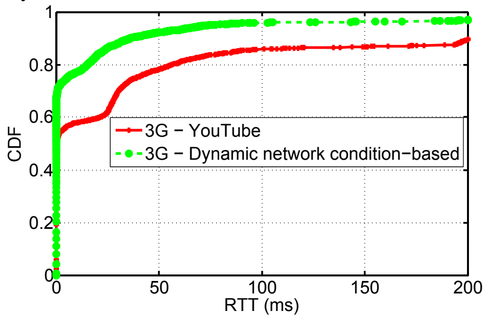

Figure 4: CDFs of RTTs measured while downloading a video over a 3G network in the HTTP proxy server-based testbed

- For each experiment, two clients simultaneously requested the video via the same network. Client A downloaded the video from the video content server selected by YouTube, and Client B accessed the video content server chosen by our dynamic network condition-aware algorithm; and
- Using `netem`, a networking emulation tool [12], we emulated the network congestion between the clients and VSPC: additional 2 ms packet delay was injected and packet loss, duplication and re-ordering rates were set up to 5%.

Video QoE tests - Considering network delivery issues, we take into account video stalling period (also known as buffer freezing) to analyze the user-perceived quality. In order to achieve this, we calculated the accumulated video data as time elapsed and compared it with the required downloading data rate. Figure 3 shows one of our video QoE experimental results over Wi-Fi networks. Client B took only 222 seconds to complete downloading the entire video content while Client A took 290 seconds, which indicates that Client A often experienced buffer freezing while playing the video. During the experiments, Client B with our dynamic server selection algorithms experienced the average of 15 seconds less buffer underflows compared to Client A.

Comparing RTTs while downloading a video - We measured RTTs until the same video content was completely downloaded from the two video content servers over 3G networks during busy hours (13:00-15:00). Figure 4 compares the CDFs of the measured RTTs. The solid line represents the CDF of RTTs between Client A and the video content server selected by YouTube, and the dotted line indicates the CDF of RTTs between Client B and the video content server that showed the shortest RTT at the requested time of the video. The experimental results indicate that our dynamic network condition-

TABLE III: TCP traffic analysis while downloading YouTube videos on PCs and mobile devices over 3G and Wi-Fi networks

Locations	Networks	Video server selected based on	TCP throughput Avg. (KB/s)	Retry Avg.	Duplicate ACK Avg.	Out of order Avg.	RTT Avg. (ms)	RTT Stdev. (ms)
Loc. 1 (Columbia Univ.)	Wi-Fi	YouTube	568.35	8.5	85.5	45	0.25	5.2
		Dynamic network status	593.82	3.1	102.6	20.3	0.15	4.05
	3G	YouTube	219.51	52.3	2105.5	750.5	3.7	21.55
		Dynamic network status	369.35	17.2	1677.2	657.5	1.15	10.8
Loc. 4 (Penn. Station)	Wi-Fi	YouTube	353.17	2.5	745.25	63.75	0.3	5.32
		Dynamic network status	430.16	1.2	272.1	18.75	0.25	5.12
	3G	YouTube	204.02	26.4	204.6	16.8	0.94	13.51
		Dynamic network status	237.33	8.66	123.16	3.16	0.55	9.31

aware video server selection algorithm provided shorter RTTs while downloading the videos, compared to YouTube's video server selection algorithms.

Comparing TCP performance - We further analyzed TCP performance from the two different places (Columbia campus and Penn. Station). From each place, we requested one hundred videos both on PCs and mobile devices over Wi-Fi and 3G networks during afternoon hours (12:00-17:00). We captured TCP/IP and HTTP packets using VSPC, and analyzed the dump files using a TCP trace tool [13]. The details of TCP performance analysis are shown in Table III. During the experiments, our proposed algorithm showed better TCP performance 146 times out of 200 experiments (73%) over Wi-Fi networks and 162 times out of 200 experiments (81%) over 3G networks. The analytical statistics indicate that our dynamic network condition-aware video server selection algorithm typically provided higher TCP performance while the video was delivered to the client.

VI. Related work

Several researchers have investigated video server selection algorithms, considering geographical locations of video servers. Torres et al. [4] found that a variety of factors such as RTTs between clients and video content servers, load-balancing, variations across DNS servers within a network and video popularity may affect the video content server selection process in YouTube. Adhikari et al. [5], [10], [14], [15] conducted YouTube infrastructure studies by collecting video traces at ISP backbone networks. They analyzed YouTube video distribution architecture, and found that YouTube deploys a large number of video caching server, that vary in size and geographical locations, in order to reduce cost and improve the end-user performance. Saxena et al. [16] analyzed how three video content providers (YouTube, Dailymotion and Metacafe) distribute their video streaming services in terms of clients' geographical locations and video characteristics such as age and popularity. Our approach differs from the prior work in two aspects: *a*) Noticeably, in some cases, we found that YouTube assigns a non-optimal video content server to a client; *b*) Unlike ALTO [7], our proposed solutions do not require any additional implementation neither at the server nor at the client.

VII. Conclusions

We have analyzed the video server selection algorithm of YouTube. During the measurements, we proved that the network attachment point of a client is considered important when YouTube assigns a video content server to a client, while other factors such as the hardware specification, the operating system and the video application running on a client's device do not affect the video server selection algorithm. Our proposed dynamic network conditions-aware approach which takes into account dynamic packet loss rates and RTTs to find the optimal video content server achieves better TCP performance than a distance-based server selection algorithms, and enables enhancing the end-user experience while playing a video.

References

[1] Akamai. [Online]. Available: http://www.akamai.com/

[2] Limelight. [Online]. Available: http://www.limelight.com/

[3] S. Kim, S.-S. Seo, J.-M. Kang, G. Pujolle, and J. W.-K. Hong, "Autonomic Resource Allocation for Video Streaming Services in Content Delivery Networks," in *Global Information Infrastructure and Networking Symposium (GIIS)*, Choroni, Venezuela, Dec. 2012.

[4] R. Torres, A. Finamore, J. R. Kim, M. Mellia, and S. Munafo, Maurizio M.and Rao, "Dissecting Video Server Selection Strategies in the YouTube CDN," in *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*, Minneapolis, Minnesota, USA, Jun. 2011.

[5] V. K. Adhikari, S. Jain, Y. Chen, and Z.-L. Zhang, "Reverse-Engineering the YouTube Video Delivery Cloud," in *Proceedings of Hot Topics in Media Delivery Workshop, co-located with IEEE ICME*, Barcelona, Spain, Jul. 2011.

[6] K. Balachandran, D. Calin, E. Kim, and K. Rege, "Clearmedia: A Proxy-based Architecture for Streaming Media Services over Wireless Networks," in *Personal, Indoor and Mobile Radio Communications (PIMRC)*, Athens, Greece, Sep. 2007.

[7] B. Niven-Jenkins, G. Watson, and N. Bitar, "Use Cases for ALTO within CDNs," IETF Draft, Apr. 2011.

[8] H. Nam, B. H. Kim, D. Calin, and H. Schulzrinne, "Mobile Video is Inefficient: A Traffic Analysis," Department of Computer Science, Columbia University, Tech. Rep. cucs-018-13, Jul. 2013.

[9] GeoIP databases and Web services. [Online]. Available: http://dev.maxmind.com/

[10] V. K. Adhikari, S. Jain, and Z.-L. Zhang, "Where Do You "Tube"? Uncovering YouTube Server Selection Strategy," in *International Conference on Computer Communication Networks (ICCCN)*, Maui, Hawaii, Jul. 2011.

[11] J. F. Kurose and K. W. Ross, Eds., *Computer Networking A Top-Down Approach*. Pearson Education; 6th edition, 2012.

[12] A. Jurgelionis, J. Laulajainen, M. Hirvonen, and A. Wang, "An Empirical Study of Netem Network Emulation Functionalities," in *Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, Maui, Hawaii, Jul. 2011.

[13] Tcptrace. [Online]. Available: http://www.tcptrace.org/

[14] V. K. Adhikari, S. Jain, and Z.-L. Zhang, "YouTube Traffic Dynamics and Its Interplay with a Tier-1 ISP: An ISP Perspective," in *Proceedings of the 10th ACM SIGCOMM conference on Internet Measurement Conference (IMC)*, New Delhi, India, Sep. 2010.

[15] V. K. Adhikari, S. Jain, G. Ranjan, and Z.-L. Zhang, "Understanding data-center driven content distribution," in *Proceedings of the ACM CoNEXT Student Workshop*, Philadelphia, Pennsylvania, Nov. 2010.

[16] M. Saxena, U. Sharan, and S. Fahmy, "Analyzing Video Services in Web 2.0: A Global Perspective," in *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Braunschweig, Germany, May 2008.