A Novel CDN Testbed for Fast Deploying HTTP Adaptive Video Streaming

Miran Taha
ETSI Telecommunication
Department of Telematics System Engineering
Technical University of Madrid (UPM)
Av.Complutense 30, 28040, Madrid, Spain
abdullah@dit.upm.es

ABSTRACT
Multimedia service providers are widely using HTTP adaptive streaming technology over Content Delivery Networks (CDNs) in Internet. Content distribution in network congestion situations and request redirection are the common challenges facing CDNs performance in delivering multimedia content to end-users. In this paper a CDN testbed is designed that consist of two mechanisms in order to fast deliver the segmentation of adaptive video streaming and redirect clients request to appropriate surrogate servers that hold copy of replica video content. Evaluation results prove the efficiency of the designed testbed according to vary bandwidth, delay and packet loss. The candidate application layer protocol to push proactively media segments to surrogate servers is faster than others and the clients redirection to appropriate surrogate servers based on the performance of server load and network congestion provides better experience to end-users.

CCS Concepts
• Networks→ Network protocols → Network protocol design
• Networks→ Network performance evaluation → Network performance analysis.

Keywords
CDN; Multimedia Distribution; HTTP Adaptive Streaming (HAS); Redirection.

1. INTRODUCTION
Content Delivery Networks (CDNs) are large-scale distributed servers (cache servers or surrogate servers) strategically deployed in multiple geographical locations in order to replicate contents as shown in Figure 1. They first evolved in 1998 [1]. CDNs provide services that improve data access by enhancing bandwidth and minimizing access latency. Examples of commercial CDNs are Akamai, Amazon CloudFront, EdgeCast, Limelight, Level-3 and many others. The infrastructure of a CDN includes distribution of object replicas, a request-routing mechanism, content delivery to end-users and accounting. The distribution infrastructure may push and/or pull to move content from the origin server to surrogate servers. The request routing infrastructure is responsible for directing client requests to the appropriate surrogate servers. The content delivery infrastructure consists of a set of surrogate servers that deliver copies of object replicas to end-users.

Highly successful deployment of over the top (OTT) service in CDNs multimedia streaming in both live and on-demand applications is generating a high percentage of down streaming Internet traffic [2]. The bandwidth usage of Netflix as multimedia service provider climbs to nearly the 37% of the Internet traffic and it counts to consume more bandwidth than YouTube, Amazon and Hulu [3]. Finally, the 72% of all the Internet video traffic will cross over CDNs by 2019 [4].

Netflix and YouTube widely use HTTP adaptive streaming (HAS) technology [5], which is used for adapting the video quality to the current network conditions in order to deliver and adapt a video in its best possible quality. In HAS, the video content is encoded into several bitrates (qualities/representations) and each bitrate is packetized into small segments called chunks. The addresses of segments are stored in an XML file called media presentation description (MPD)/manifest file. Clients can easily retrieve video content from a server or multiple servers in a CDN once read the manifest file. There are several companies that have the proprietary of implementing adaptive multimedia streaming in Internet such as Apple HLS [6], Adobe HDS [7], Microsoft (MSS) [8]. MPEG-DASH [9] is the ISO/IEC standard. Most of the systems are using similar mechanisms for adaptive streaming while they apply different characteristics and formats. However, delivery multimedia contents over CDNs may encounter the following problems. 1) Bandwidth consumption: if there is not a suitable application protocol over reliable transport protocol to push proactively multimedia content to surrogate servers then it will cost higher to transmit and consume a lot of bandwidth. 2) Storage cost: lack of an algorithm in push-base scheme to decide which content should be outsourced and where it should be placed among surrogate servers. 3) Clients redirection: in the request routing mechanism, redirection is another issue; proximity, shortest route and server load are parameters that can be applied to provide better quality of experience (QoE) in delivery multimedia content to end-user.

The aim of this paper is to design a lightweight CDN testbed in order to easily implement mechanisms, protocols and control network traffics to delivery adaptive multimedia content. Our contributions can be summarized as follows:

1. Design a virtualized CDN testbed to provide both distribution and request routing mechanisms for delivery adaptive multimedia streaming to end-users.
2. Reduce bandwidth consumption by selecting a candidate protocol to avoid network congestion and fast deploy adaptive video segments among surrogate servers.

MOBIMEDIA 2016, June 18-19, Xi'an, People's Republic of China

DOI 10.4108/eai.18-6-2016.2264163

3. Selecting the right surrogate server based on the minimum traffic load metrics (high bandwidth, low latency, low packet loss) and determining server load to end-users in order to provide better QoE.

The remainder of this paper is organized as follows: Section two presents the related work; Section three shows the proposed design and details of implementing the proposed design; Section four illustrates some experimental results; and finally section five highlights some conclusions and future work.

2. RELATED WORK

In this section we will review some related work and present the background of CDN applications.

2.1 Background CDN applications

Nowadays, most of the traditional websites are changing to multimedia websites. For this reason, the expectation of users regarding quality of service (QoS) increased. CDNs have been widely deployed to deliver the contents from content providers to a large community of geographically distributed users. CDN use many servers and distribute the load among them so that user request will be served from the nearest server and bypassing congested network paths [10]. In addition, CDNs are used to reduce latency between the end-users and the web-services by sending the cached contents surrogate servers to users on behalf of the main servers [11]. Web objects, live-streaming media, emails and social networks are among the services that use CDN technology [12].

2.2 CDN Testbeds

A wide range of techniques has been developed, implemented and standardized for improving the performance of CDNs. However, most CDN providers do not take advantage of these techniques. Moreover, academic CDNs based on real testbeds like PlanetLab, are treated mostly as black boxes or require the volunteer involvement of many individuals [13]. Virtual Networks over User Mode Linux (VNUML), Manage Large Networks (MLN) and NETKIT are used to create experimentation scenarios, but quite different from conventional virtualization management infrastructure tools, which are oriented to production data center and achieving optimal performance to end-users [14]. Therefore, the development of novel techniques in such environments is quite difficult or impossible. Also the lack of efficient CDN simulation tools has been highlighted in several works such as performance and interoperability. It is crucial to develop a reliable and useful testbed for evaluating and validating the performance of CDNs [13].

2.3 Bandwidth consumption and storage cost

When multimedia service providers make a contract with a single or multiple CDN operators, high network congestion and long distance between content providers to CDN servers bring issues to think on protocols and mechanisms to reduce bandwidth consumption and storage cost. Application layer protocols based on TCP and UDP are designed to deal with the characteristics of delivery objects in congestion networks and high delay between client and server [15]. The Hypertext Transfer Protocol (HTTP) over TCP is designed to optimize video streaming application in Internet and simplify deploy procedures. Unlike the use of Real-time Transport Protocol (RTP) over User Datagram Protocol (UDP), HTTP is easy to configure and is typically granted traversal of firewalls and network address translators (NATs), which makes it attractive for multimedia streaming applications [16].

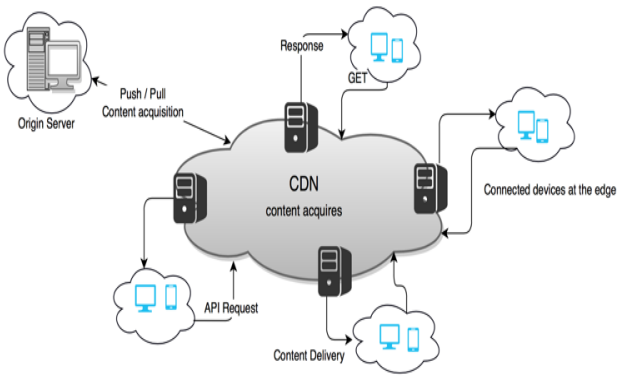

Figure 1. CDN Architecture.

In CDNs, FTP is used to guarantee a reliable data transmission to all replica servers and surrogate servers can fetch content from multiple sources via MFTP. Furthermore, there are protocols at higher layers that deal with cache management in order to fetch content. Such protocols includes the Internet Cache Protocol (ICP), Hypertext Caching Protocol (HTCP), Cache Array Routing Protocol (CARP) and Cache Digests [17]. In [18] bandwidth and storage cost issues are discussed, NP-complete is considered as the optimal placement of the outsource objects in CDNs servers. Latency based-approach is developed by [18] that uses adaptive, non-parameterized techniques to place outsource objects to CDNs servers. The approximation algorithm in [19] is developed to provide theoretical analysis of time and space complexities in order to minimize cost storage, latency and consumption bandwidth in push and pull schemes.

2.4 Redirection mechanism

CDNs take advantage of the DNS protocol to resolve domain name to its corresponding IP address to redirect clients to proximity surrogate server. Moreover, DNS has some problems [20] regarding the lookup latency (webpage become more complex) and update latency (latency to update which is not easy to select suitable time to live, TTL). TTLs adversely affect the lookup performance and increase network load [21]. CDN-based DNS prototype in [21] is designed to improve DNS lookup time compared to legacy DNS. Regarding the disadvantages of DNS request route, [22] stated that this mechanism has high latencies of data delivery and network congestions. In addition, they have introduced the service-oriented routers (SoRs) in order to reduce disadvantages of the DNS-based request route. These SoR routers have ability to capture information of packets to redirect clients to right surrogate servers. Transportation of the redirection problem is defined in [23], the object function of the paper is to minimize the cost of serving a video request from user population x using server y. They used latency and packet loss metrics to redirect clients to optimal cache server. End-users based on DNS redirection, can lead to significant latency, which impact the QoE of the HAS services. [24] focus on evaluating the impact of HTTP-based redirection in federated CDNs to increase the QoE of HTTP-based adaptive streaming through providing two novel policies to rewrite client request to correct CDN servers.

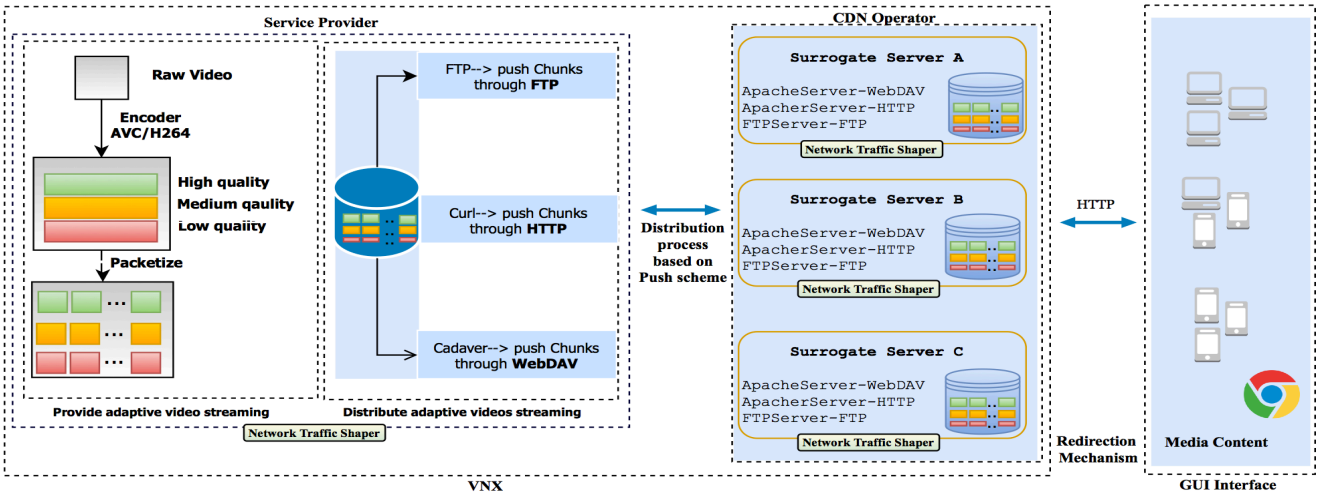

Figure 2. CDN Testbed Design.

3. SYSTEM DESIGN

This section contains the design of a CDN testbed, application layer push-based candidate protocol and algorithm and mechanism to redirect clients request to appropriate edge of networks. The testbed prototype consists of four components: Origin server (content provider), Surrogate servers (Cache servers/service provider), Network traffic shapers, and clients. These components are connected together through high transit network links as it is shown in the Figure 2. Thus, the testbed provides two mechanisms. First, proactive content distribution using candidate application layer protocol to de transmit video content and application service to surrogate servers. Second, redirection clients request to an appropriate CDN surrogate servers that have the copy of objects.

To deliver service to clients, content provider proactively replicate media segments over surrogate servers. When the client access to the domain name of original server, the clients interact with origin server to connect an appropriate surrogate server will be assigned by the system in order to provide application service and list of available videos. Each component will be described in the next subsections.

3.1 Original server

Original server is a top server that provides adaptive video streaming content to clients through surrogate servers. In the origin server, raw video that is an uncompressed file will be encoded into various bitrates/qualities in order to provide adaptive video streaming over HTTP. Therefore, each bitrate packetized into several media segments/chunks and each chunk has a time duration, which might be between 2 to 10 seconds [25]. Moreover, these media segments reside in local storage of the server and media segments manifested in the XML file to allow end-user easily access to media segments in one or more servers [26].

3.2 Surrogate servers

Surrogate servers are used to keep a copy of replica adaptive video streaming content. The characteristics of these servers provide two mechanisms. First, they receive proactively a copy of the video content from original server or pull the video content from original server to surrogate servers to service clients in the case when cache servers get missing content. Second, provide service web application play back to clients.

3.3 Network traffic shapers

The network traffic shaping emulator in the testbed has utilized to control throughput of available bandwidth by prioritizing network resources and guarantee certain bandwidth based on predefined policy rules. It uses concepts of traffic classification, policy rules, queue disciplines and quality of service (QoS). The network shaping is the combination of traffic control (TC) queuing discipline Hierarchy Token Bucket (HTB) and Network Emulation (NetEm) in order to shape and control the network link's upload, network link's download, delay and packet loss ratio.

3.4 Clients

Clients may be computers, tablets and mobiles located outside the virtualized testbed. PCs may operate different operating system including Linux, Windows, IOS, etc. They can access the CDN servers to get service through using Chrome browser to play out video streaming. The characteristic of Chrome is to support media source extension (MSE) to handing the media segments together. Therefore users can upload and modify video content in the system as the permission given by servers.

3.5 Mechanisms

In this subsection we will describe the two mechanisms, distribution and redirection.

First the distribution mechanism, the system selects an appropriate application layer protocol that satisfactory deliver small chunks of adaptive video streaming. We use the delivery application layer protocols that are based on TCP transport layer, respectively includes: FTP, HTTP and web distributed authoring and versioning (WebDAV1) in order to choose the right application

[1] https://tools.ietf.org/html/rfc4437

layer protocol that suit to fast deploy small chunks of media content among surrogate servers.

The second mechanism is the redirection algorithm to decide which surrogate server can be used to serve clients. The adaptive request routing algorithm involves the estimation of metrics such as performance of server load and the congestion of the network links by taking advantage of internet control message protocol (ICMP) to calculate round trip time of packets and packet loss ratio Then the URL re-writing informs the clients about the selection of the nearest surrogate server and it is generated by the request-routing algorithms.

The Pseudo Code algorithm described in Algorithm 1 shows the process of client interaction with service provider to get multimedia service. We represent surrogate server by SS_k and there are N surrogate servers as k = {1,2,3…N} and represent clients by C_i, i may be {1,2, 3…M}. IP addresses of clients can be captured by content provider (original server) as denoted by CP and deliver to number of M surrogate servers, SS_k send packets asynchronously to C_i in order to calculate the average of minimum latency and packet loss. CP selects optimal result and re-rewriting the URL of HTTP client request to optimal SS_k.. Moreover, the placement of original server and surrogate servers in the CDN infrastructure is related to network metrics, such as achievable bandwidth, round trip latency (both directions) and packet loss. Each component has network traffic shaper as shown in Figure 2 to characterize different behaviors of the amount of data that the network can transfer per unit of time.

3.6 Implementation tool

Virtualization based testbeds are widely used in network environments to test protocols and applications in order to reduce cost, hardware resources needed and complexity of present networks. Virtual networks over Linux called (VNX)[14] is a tool designed to help building virtual network testbeds automatically. It allows deployment of network scenarios made of virtual machines for different types of operating systems. The VNX tool allows the definition of virtual network scenarios and controls their deployment over either a Linux server or a cluster of servers. The user can control how the virtual scenario is distributed over different cluster servers, using algorithms, redistricting rules. The main reasons of using VNX in this project are agility, easy to implement requirements over the Linux operating system, fast encode video content to adaptive bitrate streaming through command lines, availability and fast configuration of protocols, and modify performance of components.

Algorithm 1, Pseudo code redirection client request to optimal surrogate server.

Client side:
1 C_i sends GET to request multimedia web service
Original Server:
2 receive C_i request in http message
3 capture C_i IP address
4 **For** $SS_k = 1$ → N
5 send C_i IP address to N surrogate servers through http
6 **End For**
Surrogate Servers:
7 Receive the http request from original server contains IP of C_i
8 SS_k apply asynchronous task
9 **For** $SS_k = 1$ → N
10 Send 10 packets to client IP address
11 **End for**
12 Calculate average of latency && ratio of packet loss between SS_k and C_i
Original server
13 **While** SS_k do not send response or time out
14 **If** SS_k has responded
15 Calculate minimum latency [***k***]
16 **EndIf**
17 **EndWhile**
18 Rewrite URL to surrogate server [***k***] has minimum latency
Client side:
19 Receive web service and list of videos from surrogate [***k***] has minimum latency

4. EXPERIMENTS

This section describes the experiments over the CDN testbed, it includes virtualized original server, three surrogate servers and external clients that connect to the CDN testbed with high speed network connections. The device performances are summarize in Table 1. An Apache server was configured to serve http requests, Cadaver, FTP client and Curl to push media segments of adaptive video streaming onto surrogate servers. Therefore, an X2642 codec tool was used to encode the raw video 'y4m' into H.264/AVC although it provides various H.264 bitrates. MP4Box3 is used to keep the H.264 codec in a MP4 container, and then packetize the MP4 videos through MP4Box to adaptive bitrate streaming including segments and manifest file. On surrogate servers, we installed an apache server, FTP server, DAV module to support WebDAV and a squid cache server for caching contents. Client side uses Chrome browser to playback video streaming on the web application. Moreover, Iperf is used to activate measurement of quality links; it reports the bandwidth, delay and packet loss of networks.

Table 1. Device performances.

Item	Servers	Clients
CPU	2,4 GHz Intel /4 Core	2,4 GHz Intel /4 Core
Memory	4GB	8GB
Hard Disk	500GB	500GB
Graphic Card	-	512MB
O.S	Debian	Ubuntu, IOS

In our experiments, we select short segment length of adaptive video bitrate streaming that discussed in [5]. We chose the raw video of Big Buck Bunny and encode 7200 frames (120seconds) into three qualities/representations R = [1, 2, 3] inter-dependent from the AVC segments.

The property of each level is shown in Table 2, and then we segment each representation into the duration of 2 seconds (I-Frame = 48 frames) with different sizes. The manifest file contains the address of placement of media segments in the servers indicated by an URL. Also in order for the clients to use an appropriate play back application, open source DASHJs application is used. It is a seamless integration of dynamic adaptive streaming over HTTP based on JavaScript which uses the Media source API Google's chrome to present a flexible and potentially browser independent of the DASH player4.

[2] http://www.videolan.org/developers/x264.html

[3] https://gpac.wp.mines-telecom.fr/mp4box/

[4] http://dashif.org/

Table 2. Information of the adaptive video streaming.

Rep.	Seg.	Bitrate (Kbps)	Spatial resolution (pixel)-width	Spatial resolution (pixel)-height	FPS
@R1	60	250	640	360	24
@R2	60	500	704	480	24
@R3	60	1000	1280	720	24

In our experiments, the distribution process includes the utilization command lines such as Cadaver, Curl and FTP client to push the media segments from content provider into surrogate servers. In addition the video chunks automatically keep in the local cache of each surrogate servers. In the redirection mechanism the series of packets send between IP layers of surrogate server and clients to specify which surrogate server become an appropriate server to provide service to end-users. Also clients can see list of available videos that pushed from original server to correspond surrogate server.

To evaluate the performance of distribution the segmentation of adaptive video streaming in the CDN testbed we provide different impairment of available bandwidth, high and low distance (delay) between content provider to a surrogate server and then vary range of packet loss ratio in order to find out arrival time of video chunks. Therefore we analyzed the performance of TCP of each application layer protocol according to number of the packets that can be retransmitted, and TCP throughput. Furthermore we utilized Wireshark to capture information of sent/received packets in the networks. To apply the experiments we merely determine one surrogate server to replicate adaptive video contents. Network traffic shapers are used to impair the bandwidth, delay and packet loss. Each experiment has been repeated 10 times to get accurate results. So in the first experiment we impair the bandwidth into different ranges that consists of 275Mbps, 45Mbs, 5Mbps and 1.54Mbps in order to find out the performance of which application layer protocol waste less time in the process of replication video contents in the surrogate server, as depicted in Figure 3.

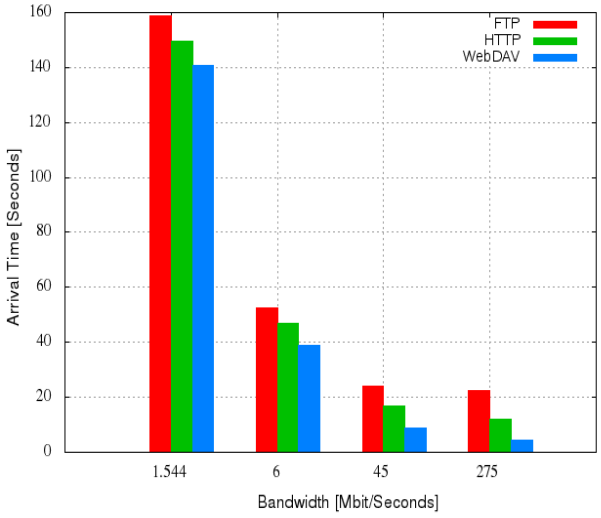

Figure 3. Arrival time in vary bandwidth.

The blue bar indicates the transmission time of entire chunks that took through WebDAV protocol, the green bar for HTTP and the red bar for the FTP transmission. The blue bar in this experiment wasted less time to deliver entire chunks of the adaptive video streaming, also expanding the bandwidth straightly affected on arrival time of chunks especially in WebDAV. Different values of delays (round trip time) in the second experiment has been proposed as shown in the Figure 4 in order to discover the effect of latency on arrival time of replication media segments on the surrogate server. The blue line is the WebDAV protocol that has been moved up slightly from 20ms to 120ms. The red line and the blue line have been turned up highly especially the red line used much more time to arrive video chunks. Therefore, increases in latency between original server and surrogate server have a huge impact on delivery chunks in FTP and HTTP. In the third experiment, as shown in Figure 5, we set the range of packet loss between 0% to 2.5% with 6Mpbs of available bandwidth and delay 20ms. In this experiment, packet loss rates highly affected of the delivery time of chunks through FTP and HTTP than WebDAV. Also WebDAV has barely moved up from its position.

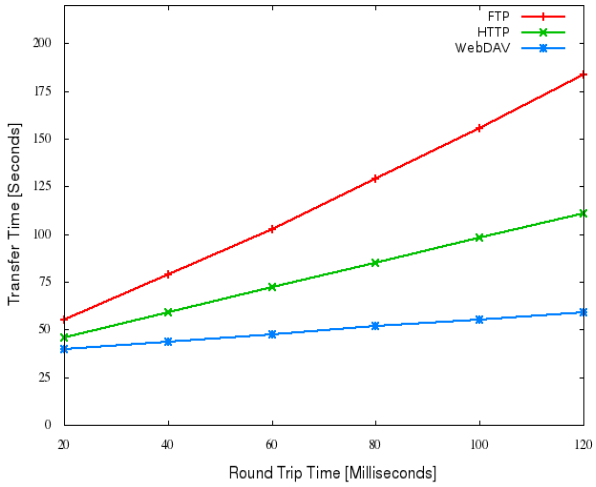

Figure 4. Effect of RTT on arrival time.

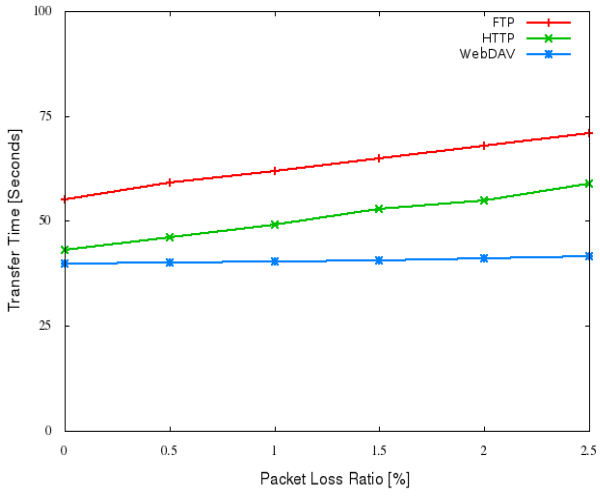

Figure 5. Effect of packet loss on arrival time.

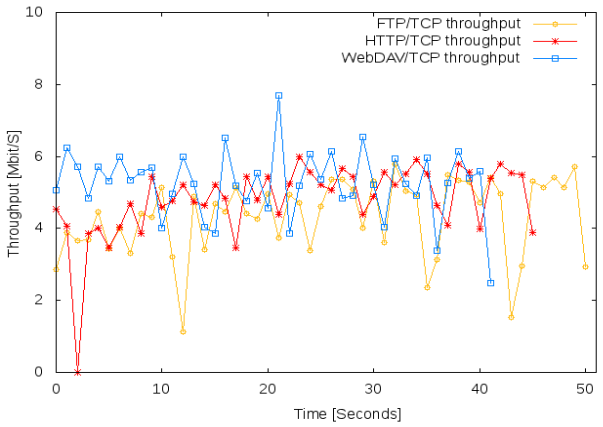

Figure 6. TCP throughput.

In the fourth experiment, we disclosed the performance of transport layer (TCP) for each application layer protocols regarding how much bandwidth they use to deliver packets as depicted in Figure 6. In this experiment we constrain the bandwidth to 6mbps, 20ms delay and 0.1% packet loss as well. The blue line uses maximum rate of available bandwidth, it is approximately between 5Mbps to 6Mbps, and its high pick rate in the second 20 to deliver video chunks and the transmission takes 40 seconds to deliver all chunks of the video. The red line in the initial delivery of chunks depredated to low value, this makes this protocol to use less packet size to arrive on surrogate server and the duration of whole time transfer of chunks took 45seconds. Also the brown line used less bandwidth to transferring packets further the brown line has been dropped to use 1.5 Mbps in second 12 and the total time was 50 seconds. In the fifth experiment we illustrate the number of packets that are retransmitted in original server through TCP transport layer of three application layer protocols. For this experiment we set 0.1 percent of packet loss, 6mbps available bandwidth and 20ms delay in the shaper. In the Figure 7 the red line indicates numbers of packets, which have been retransmitted in FTP is higher than the green line. Therefore the blue line indicates TCP of WebDAV non-packet is retransmitted. It means that packet loss and delay barely have effect on WebDAV protocol to deliver small chunks.

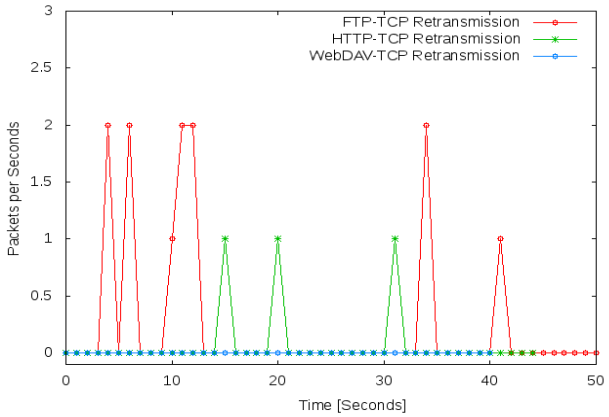

Figure 7. Packet retransmission.

In previous experiments we learned that, WebDAV used a header size longer than HTTP and FTP. Persistent connection, pipelining and permanently connected to destination until task finish makes HTTP and WebDAV fast, easy perform better than FTP. In the last experiment, the request routing mechanism involves to sending series of IP layer packets through ICMP that reported in PING[5] command to client side, maximum range of packet size is 63353 bytes. Also sending maximum size of packets and small packet size are not the optimal solution in the experiments because maximum size is producing high congestion and CPU load on devices especially in constraint networks. Moreover small size can not find accurate results. In this experiment we employed 128-bytes, including 120 bytes of data and 8 bytes of ICMP header to check the connectivity and calculate minimum average latency between surrogate servers and clients. Once the client sends a request to original server to get multimedia service, the original server sends the client IP address (the client IP address captured through the PHP program) to surrogate servers through HTTP protocol. Surrogate servers use multi-thread process to send asynchronously packets to speed up executes tasks and reduce client page load as depicted in the Table 3. Each surrogate server sends packets of 128 bytes for 10 times and the interval between each packet is 0.2 seconds. Results of packet loss and latency are reported to original server in order to respond to client requests by rewriting the URL to optimal surrogate server and then the client gets the web page service application from optimal surrogate server, which contains the list of available videos. The client can retrieve the media segments of adaptive video streaming from its surrogate server. On the other hand if the video is not available, the cache server brings it from the original server to respond to client requests.

Table 3. Page load time.

CDN request route	Client page load
Not apply asynchronous	1.02 seconds
Apply asynchronous	0.45 seconds

In Table 4, we summarize the benchmarks between the proposed mechanism and the conventional approaches

Table 4. Comparisons between proposed approach and conventional approach.

No.	Proposed approach	Conventional approach
1.	Simple and easy to use	Complexity and limitation
3.	Content distribution process based on HTTP/FTP/WebDAV	FTP
4	Provider-related DNS (URL re-writing) service at server side offloads the burden of the other DNS server and reduces the latency of redirection.	Redirection based on DNS servers creates additional HTTP request and adds RTT (round-trip-time)

5. CONCLUSION

Our contribution in this paper was to design a CDN testbed to easily implement protocols and mechanisms in order to provide a faster and robust web multimedia delivery service to end-users. Therefore we used application layer protocols which are based on TCP layer protocol namely, i) FTP, ii) HTTP and iii) WebDAV to push proactively media segments of adaptive video streaming on to cache server/surrogate servers and a request routing algorithm

[5] https://tools.ietf.org/html/rfc2812

to redirect client requests to an appropriate surrogate server that can service the client requests. Our aim in the paper was to reduce bandwidth consumption by fast deploying media contents in CDN servers in vary and restricted network conditions, and redirect clients request to appropriate surrogate servers that be based on high availability CPU resource and minimum end-to-end delay. Results showed and analyzed that heterogonous devices can be easily connected to the CDN testbed. The process of proactive distribution through WebDAV is better than regular HTTP and FTP to replicate small media segments in high distance between the origin server and CDN operators. Therefore WebDAV can be used channels with high packet loss due to barely packets retransmission as shown in Figures 3, 4 and 5. In addition, surrogate server load and network metrics are prominent parameters to decide on request redirection.

Finally, our future work is to investigate on pull-based scheme to provide better QoE to end-users by fast retrieving multimedia contents to cache servers while content cache is missed and provide dynamic request routing mechanism to check server load and network congestion while client can not get better service from corresponding surrogate servers.

6. REFERENCES

[1] Pallis, G. and Vakali, A. 2006. Insight and perspectives for content delivery networks. *Commun. ACM*. 49, 1 (January 2006),101-106.

[2] Ganuza, J.J. and Viecens, M.F. 2014. Over-the-top (OTT) content: implications and best response strategies of traditional telecom operators. Evidence from Latin America. *Info-The journal of policy, regulation and strategy for telecommunications*, 16, 5 (Aug. 2014), 59-69.

[3] Sandvine report, Sandvine intelligent broadband networks, 2016, URL https://www.sandvine.com/trends/global-internet-phenomena/

[4] Cisco White Paper, Cisco Visual Networking Index: Forecast and Methodology, 2014-2019, Tech. Rep Cisco, 2016. URL http://bit.ly/bwGY7L.

[5] Seufert, M., Egger, S., Slanina, M., Zinner, T., Hobfeld, T. and Tran-Gia, P. 2015. A survey on quality of experience of HTTP adaptive streaming. *Communications Surveys & Tutorials, IEEE*, 17,1 (September 2015), 469-492.

[6] Apple Inc. HTTP Live Streaming Overview, 2013.

[7] Adobe Systems Inc. HTTP Dynamic Streaming, 2013.

[8] Zambelli, A. IIS Smooth Streaming Technical Overview, Microsoft Corporation. 2009

[9] International Standards Organization/International Electrotechnical Commission (ISO/IEC), 23009-1:2012 Information Technology – Dynamic Adaptive Streaming over HTTP (DASH) – Part 1: Media Presentation Description and Segment Formats (2012).

[10] Ni, J. and Tsang, D.H. 2005. Large-scale cooperative caching and application-level multicast in multimedia content delivery networks. *IEEE Communications Magazine*, 43, 5 (May 2005), 98-105.

[11] Sinha, A., Mani, P., Liu, J., Flavel, A. and Maltz, D. 2016. Distributed Load Management Algorithms in Anycast-based CDNs. arXiv preprint arXiv:1603.00406.

[12] Saroiu, S., Gummadi, K.P., Dunn, R.J., Gribble, S.D. and Levy, H.M. 2002. An analysis of internet content delivery systems. *ACM SIGOPS Operating Systems Review*, 36(SI), 315-327.

[13] Stamos, K., Pallis, G., Vakali, A., Katsaros, D., Sidiropoulos, A. and Manolopoulos, Y. 2010. Cdnsim: A simulation tool for content distribution networks. *ACM Transactions on Modeling and Computer Simulation* (TOMACS), 20, 2, 10.

[14] Fernández, D., Cordero, A., Somavilla, J., Rodriguez, J., Corchero, A., Tarrafeta, L. and Galán, F., 2011, October. Distributed virtual scenarios over multi-host Linux environments. In Systems and Virtualization Management (SVM), 2011. *5th International DMTF Academic Alliance Workshop* on (pp. 1-8). IEEE.

[15] Gelman, J.R. and Stadler, J.S., Massachusetts Institute of Technology, 2002. Method and apparatus for improving efficiency of TCP/IP protocol over high delay-bandwidth network. U.S. Patent 6,415,329.

[16] Krasic, C., Li, K. and Walpole, J. 2001. The case for streaming multimedia with TCP. In *Interactive Distributed Multimedia Systems* (pp. 213-218). Springer Berlin Heidelberg.

[17] Molina, B., Palau, C.E., Esteve, M. and Lloret, J. 2004. On content delivery network protocols and applications. *WSEAS Transactions on Computers*, 3,6 (2004), 1981-1984.

[18] Pallis, G., Vakali, A., Stamos, K., Sidiropoulos, A., Katsaros, D. and Manolopoulos, Y. 2005. A latency-based object placement approach in content distribution networks. In Web Congress, 2005. LA-WEB 2005. Third Latin American (pp. 8-pp). IEEE.

[19] Guan, X. and Choi, B.Y. 2011, June. Push or Pull?: Toward Optimal Content Delivery. *IEEE International Conference on Communications* (ICC), 2011. (pp. 1-5). IEEE.

[20] Qin, Z., Xiao, C., Wang, Q., Jin, Y. and Kuzmanovic, A. 2014. A CDN-based Domain Name System. *Computer Communications*, 45, pp.11-20.

[21] Ramasubramanian, V. and Sirer, E.G. 2004. The design and implementation of a next generation name service for the internet. In *ACM SIGCOMM Computer Communication Review*, 34, 4 (2004), 331-342. ACM.

[22] Harahap, E., Wijekoon, J., Tennekoon, R., Yamaguchi, F. and Nishi, H. 2013, December. Router-based request redirection management for a next-generation content distribution network. In *Globecom Workshops* (GC Wkshps), 2013 IEEE (pp. 1007-1012). IEEE.

[23] Mundur, P. and Arankalle, P., 2006. Optimal server allocations for streaming multimedia applications on the Internet. *Computer Networks*, 50,18 (2006), 3608-3621.

[24] Famaey, J., Latré, S., van Brandenburg, R., van Deventer, M.O. and De Turck, F. 2013. On the impact of redirection on HTTP adaptive streaming services in federated CDNs (pp. 13-24). Springer Berlin Heidelberg.

[25] Lederer, S., Müller, C. and Timmerer, C., 2012, February. Dynamic adaptive streaming over HTTP dataset. In *Proceedings of the 3rd Multimedia Systems Conference* (pp. 89-94). ACM.

[26] Adhikari, V.K., Guo, Y., Hao, F., Varvello, M., Hilt, V., Steiner, M. and Zhang, Z.L., 2012, March. Unreeling netflix: Understanding and improving multi-cdn movie delivery. In *INFOCOM, 2012 Proceedings IEEE* (pp. 1620-1628). IEEE.