Global Journal of Engineering and Technology Review
Journal homepage: www.gjetr.org
Global J. Eng. Tec. Review 3 (4) 01 – 09 (2018)

# Improvement of Available Bitrate Streaming in Content-Delivery Networks (CDNs)

* **Nader F. Mir[1], Han W. Shih, Bojun Liu[2], Venkata S. Viswanatha[3], and Kamal M. Shaik[4]**

[1234]Department of Electrical Engineering San Jose State University

## ABSTRACT

In this paper, we present a technique through which the performance of video Adaptive Bitrate Streaming (ABS) is improved. Video providers and Content Delivery Networks (CDNs) use ABS quite often. ABS is an advanced HTTP-based technology which keeps the advantages of HTTP. An adaptive bit rate streaming traffic is first examined with an improved technique for the network using network simulator. Jitter and delay are the primary focus of the analysis when multiple rates are used in a test bed. Jitter and delay are in particular analyzed for video streaming in the presence of background traffic and used for comparison. The aim of this paper is also to analyze the end-to-end delay, throughput and loss of packets in the network that are part of the content delivery network of cloud systems. Multimedia traffic is transmitted at different bitrates starting from 30kbps, 55kbps to 120kbps. The network's performance is improved and enhanced by implementing a type of traffic shaping a technique by prioritizing and smoothening the multimedia traffic. Two different network simulators are used to implement the idea presented in this paper. The simulation involves study, analysis and comparison of performance in network with traffic shaping and without traffic shaping.

**Type of Paper:** Empirical.

***Keywords***: Video Streaming; Cloud System; Adaptive Bitrate Streaming (ABS); Content Delivery Network (CDN).

## 1. Introduction

Streaming media at a source host in a packet-switched network, is the act of constantly transmitting packets and therefore their bits to a destination host through the network. This includes webcasts, web TV, live casting, etc. and uses a streaming protocol like RTSP to transmit the data. Real time streaming is adaptive most of the times enabling the change in transfer rate based on transfer conditions like bandwidth and resolution of the client [8].
With the continuous improvement in Internet and web technologies, the need for multimedia applications and services have immensely increased. Video, as a media component, has become a principal mode of providing information to everyone. Presenting content through

*Paper Info: Revised: August 11, 2018
Accepted: December 22, 2018
* Corresponding author: Nader F. Mir
E-mail: nader.mir@sjsu.edu
Affiliation: Department of Electrical Engineering San Jose State University San Jose, California 95192, U.S.A

ISSN 0128-2905. 

video has now become conservative in industry and academics [1][2]. Visual representation of data receives more interest from people than any other form of representation. More information can be conveyed in less amount of time [3][4][5]. For many days, the media was presented to end users through downloading the content. The user should be ready to wait for the duration of the video to be downloaded before actually viewing it, which is not the usual case. Video Streaming is the technique where the video content is delivered over the Internet for immediate playback [6].

Figure 1 depicts an illustration of streaming a media object. The stream of the object's content must ideally be achieved at a constant bit rate. When the content of a media object such as a video clip is to be streamed from a source host to a destination host, the content is encapsulated into payloads of packets and the bits of each packet are continuously transmitted over the link that is attached to the source host using the available link speed.

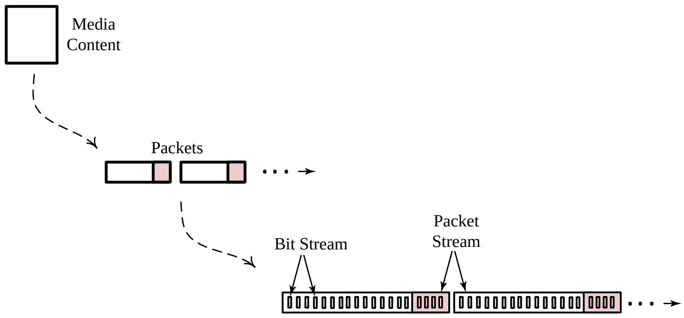


Figure1. Streaming the content of a media content.

Video traffic in this paper is set up using the adaptive bit rate streaming (ABS) [2]. ABS is a management technique for streaming live or pre-recorded media over computer networks. ABS can be included in most real-time media transmission protocols. The most practical deployment of ABS is the one built for transmission over HTTP in networks. Through the adaptive bit rate streaming, viewing experience of a diverse range of devices over a broad set of connection speeds can be made possible.
In the next sections, we first demonstrate an architecture as a backbone for high-speed networking and then the video streaming traffic is present. We use simulation to simulate traffic shaping applied over adaptive bitrate streaming.

## 2. Testbed Network Architecture

The testbed network is composed of subnets, links and nodes which are used in our experiment. Figure 2 shows the startup wizard used and selected network as our default network. For selecting the network devices using the object palette window that is already provided inside Opnet modeler. A server could be a separate hardware or a piece of software running inside a home computer. It has a static IP address and is never switched off. Each client is a personal computer that is used by a normal user to request data. It has a dynamic IP address and gets its IP address from DHCP pool. It sends a HTTP request to the server and uses TCP to connect to the client. Application config is one of the most important objects in this paper. It is used to set the definitions for an application. We have used application config to set video streaming application in this paper.

In this paper, we first create a profile with video application inside it. Then we will be applying that profile to the servers and the clients so that there are able to stream video application data. In the network architecture of this paper, 10BaseT is used as links to connect the network devices. These are the standards that are used in local area networks. As seen in Figure 2, two additional servers which are streaming data to the same client and to the other clients at the same time are considered. This way, the node that receives data from the server is node 6. The other nodes, are node 8, 5, 7, and 9 are mainly to create background traffic. Here background traffic is put mainly to test the network about how it performs under high load.

For the purpose of variety, the number of nodes can always be increased but not the number of routers in between them. This means that there are only two routers that handle all the traffic flow. The analysis was made on how this network performs when the sending bit rate between server and client is 30 Kbps, 55 Kbps and 120 Kbps respectively.

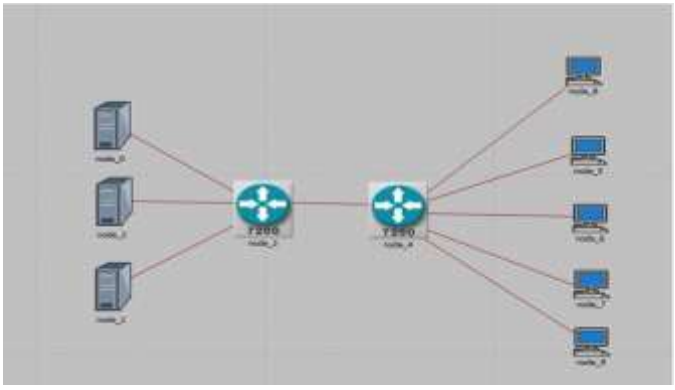

Figure2. Basic network with servers and clients

In the application definition, we selected the video conferencing option only to simulate the video streaming traffic. In the video conferencing table, we choose the streaming multimedia service and set the frame arrival time and frame size. Moreover, in the traffic mix column, we can choose the percentage of the type of the traffic. We can select all discrete, all backgrounds, or the mix.

In this paper, we set different bitrates every 200 seconds and 500 packets per second in 15 minutes as shown in Fig.2. The maximum value of bitrate is 18,000,000 bits per second, while the minimum value of bitrate is 6,000,000 bits per seconds. In the Traffic Mix column, we select all background options so the dashed line would only accept the background traffic.

The network is built up with servers, clients, and two routers. In this paper, we have four scenarios of different amounts of servers and clients with three different TCP advertised window sizes, which are 65535, 4095, and 2047. The maximum segment size(MSS) is equal to 1452. A configuration scheme is shown in Figure 3. One server corresponds to one client, One server corresponds to two clients, Two servers corresponding to one client, and Two servers corresponding to two clients, respectively.

In parameter setting, we set switch timeout values equals to 2 seconds since we want to avoid multiple quality levels switched at one time. The maximum video buffer size is 10, and the minimum rebuffering size is equal to 8, which is used to trigger the quality level judgment. In bitrate calculation, the estimated bitrate is defined as follows:

(PacketTimeArrayLength)*(VideoPacketSizePerSecond[VideoQualityIndex])/Tsum.

In the above formula, we use the quality index as an array to store different values. Therefore, once the quality index has been changed, the bit rate would be changed correspondingly. The PacketTimeArrayLength value is up to 500. Tsum is the summation of time of all the video packets.

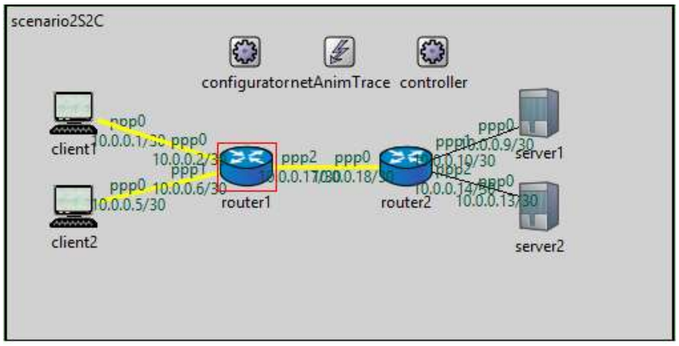

Figure3. Basic network with configuration parameters

## 3. Methodology And Flowchart

The flowchart in Figure 4 shows how we achieve ABS in this paper. Once the client achieves full buffer status, it would calculate the bitrate and decide if the quality level could increase. Once the video buffer size is smaller than rebuffer size of 8, the quality level should be decreased.

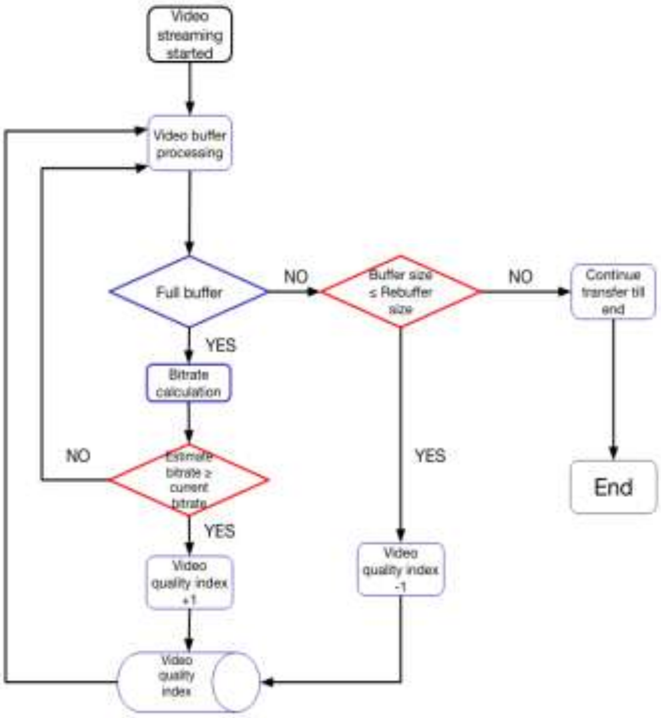


Figure4. Performance analysis flow chart

## Creating Variable Bitrates

Bitrates are created at 30 Kbps, 55 Kbps and 120 Kbps. 10BaseT links to connect the server and routers and routers to clients. The maximum bit rate that can be transmitted is 10 Mbps and maximum supported bit rate on link 10BaseT.

Editing the attributes in the links configurations individually for each links. Entering the configuration of the links, then under the traffic information tab, new row is created and allot the average packet size as default size used in Ethernet i.e. 1500 bytes of MTU. After setting up the maximum transmission unit The traffic load (bps) to assign the bit rates. The paper will be using G723 Voice_bps to generate the required bit rate.

So when the traffic load option is edited that is provided in the link configuration. A new window is appears to set the different bit rates. The simulation starts at 0 seconds and start sending a stream at this particular point of time. Now setting the bit rate here by editing the tab and giving in 30,000 bits/second. In the same way different bit rates at 55Kbps and 120 Kbps can be created.

The application config object in object palette is configured. Any applications can be created here like FTP data, http data, video conferencing data. So in this paper video streaming data (Low Quality) is chosen to stream from server to client .i.e the paper enables the video stream application in between the server and the client. Multiple applications can be enabled between the server and the client. This can be done by creating new rows inside the application definitions and enabling a new application. But in this paper network behavior can be observed when packets at different bit rates are send between server and client both with and without traffic shaping. Video application is enabled now.

## Implementing Traffic Shaping

Traffic shaping is one of the most important aspects of our paper. It is mainly used here to give priority to video application data so that End to end delay is low in between the server and the client. There is traffic shaping attribute already available in the simulator which makes it easy to enable. In figure 2.4.1, configuration of traffic shaping profiles is shown. As profiles were created for video stream application in profile config, Now it's time to create profile for traffic shaping and then assign it to traffic policies.

## Performance Analysis results

In Figure 5, the green line indicates the end-to-end delay at 120 Kbps between server and the client. The red line indicates the End to end delay at 55 Kbps. The blue line indicated the End to end delay at 30 Kbps. The minimum End to end delay here is0.01728 and maximum End to end delay is 0.01742. Here End to end delay can be observed at 120 Kbps is very high when compared with 55 Kbps and 30 Kbps. This is because when the server is transmitting data at 120 Kbps, then server is sending more data on the link when compared with 55 Kbps and 30 Kbps. This results in more congestion on the network when compared with other bit rates which results in increased End to end delay. As there is no traffic shaping all packets data applications is considered same and there is no priority in here. So when the buffer gets full, it will consider the all the packets same and this results in dropping the video stream packets or delaying the video stream packets which in turn also increases the End to end delay between the server and the client. The networks behavior can be observed and End to end delay between servers and client when traffic shaping is implemented.

Here traffic shaping is implemented on the network. The set up of traffic shaping is already shown in the previous section. Traffic shaping is implemented such that video streaming application traffic is given more priority and processed faster such that the End to end delay with the server and client is reduced. In Figure 5, End to end delay between the server and the client at different bit rates i.e. at 30 Kbps, 55 Kbps and 120 Kbps can be observed below.

In the above graph that End to end delay is always higher for the network when traffic shaping is disabled in the network whereas End to end delay is lower in the network, traffic shaping was enabled. From the graph that the End to end delay when the bit rate is 120 Kbps is as low as 0.01714 when traffic shaping is enabled whereas End to end delay when bitrate is 120 Kbps is as low as 0.01732 when traffic shaping is not enabled.
In Figure 6, the comparison of end-to-end delay of the network with traffic shaping enabled and end-to-end delay of the smaller network with and without traffic shaping is shown. Here in blue line indicated the End to end delay between the server and client in the smaller network without traffic shaping.

The minimum delay here is 0.01728 and maximum is 0.01743. Green line indicates the End to end delay of the larger network with traffic shaping enabled. From the graph that the maximum End to end delay here is 0.01719 and lowest End to end delay is 0.01712. When packet shaping is enabled in the network then the End to end delay in the network has been reduced. This is because when a number of packets arrive at the router, it processes then one after the other. But when traffic shaping is enabled it processes the video application data early when compared with other traffic. This caused the network with traffic shaping enabled to have lower End to end delay when compared with the network that does not have traffic shaping. This chapter concludes here as The paper was successfully implemented and achieved the performance enhancement in a network by implementing the performance enhancements techniques.

Figure5. End-to-end delay at different bit rates without traffic shaping

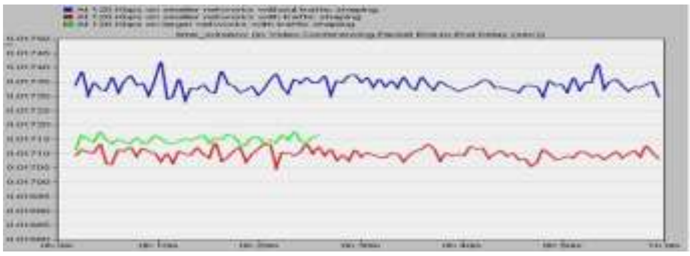

Figure6. End-to-end delay at different bit rates with traffic shaping

Figure 7 is the result based on the bitrate setting previously, and the streaming is in the background traffic. We can see that the traffic is sent in different bitrate every 200 seconds just as our setting previously. However, we can do nothing more to approach ABS since the lack of access to modify the source code.

We also used OMNET++ to carry out other aspects of performance evaluations. In the first scenario, one server corresponds to one client where we can see that the video buffer length increased from 0 to 10 (maximum). Once it achieves the full buffer status, it would calculate the bit rate and decide if the quality level could increase. Once the quality level increased, it means that the client could process more video data at the same time, which would let the video buffer decrease gradually. Once the video buffer size is smaller than 8, the quality level should be decreased. Once the quality level decreased, it means that the client could process fewer video data than before, which would let video buffer increase gradually. By repeating those procedures, the video buffer size would be stable in the interval between 8 and 10, and the quality level would be stable at 2 mostly.

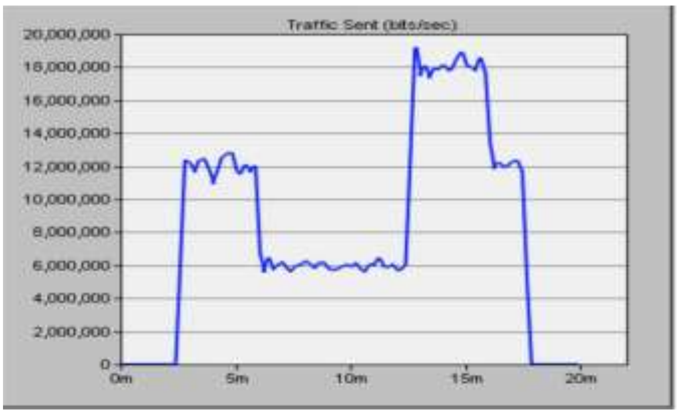

Figure7. Background traffic generated in the simulation

Figure 8 presents the quality index variation of buffer size equals to 65535, 4095, and 2047.

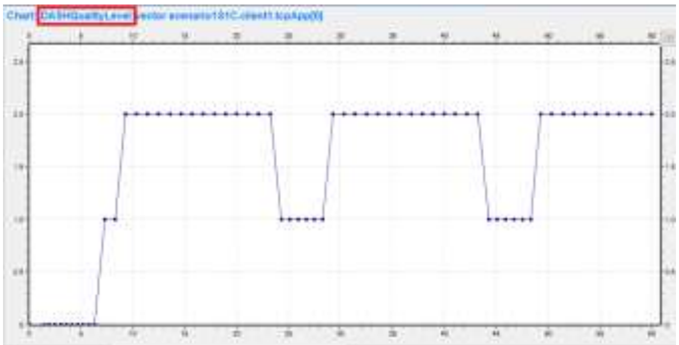

Figure8. Carried traffic load buffer size equals to 65535, 4095, and 2047

In the next scenario, we simply added one server in the network, and keep other settings the same. Basically, the simulation results of this scenario are similar to scenario 1, since we find out that the client would only connect to one of the servers in each simulation. Therefore, the transmission is just as the one-to-one process. However, we can still use this scenario to compare with scenario 4, which is two servers correspond to two clients. By repeating those procedures, we can see that in both scenarios of one client, the video buffer size would be stable in the interval between 8 and 10, and the quality level would be stable at 2 mostly. Figure 9 is the quality index variation of buffer size equals to 65535, 4095, and 2047.

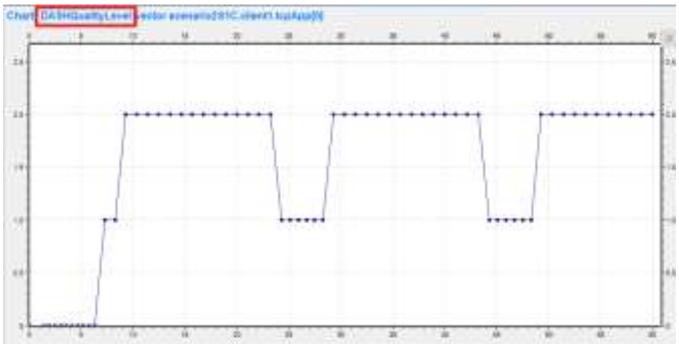

Figure 9. Carried traffic load buffer size equals to 65535, 4095, and 2047 with one additional server

Next, we added one client in the network, and keep other settings the same. Basically, most of the simulation results of this scenario are similar to scenario 1; however, if the tcp.advertisedWindow value equals to 4095, there is something different happened. Besides the setting of 4095, we have to realize that both clients would need to share the bandwidth since they all asked for video streaming but use the same links. Therefore, we tried to find out the reasons contributed to the result. Figure 10 is the quality index variation of buffer size equals to 65535, 4095, and 2047.

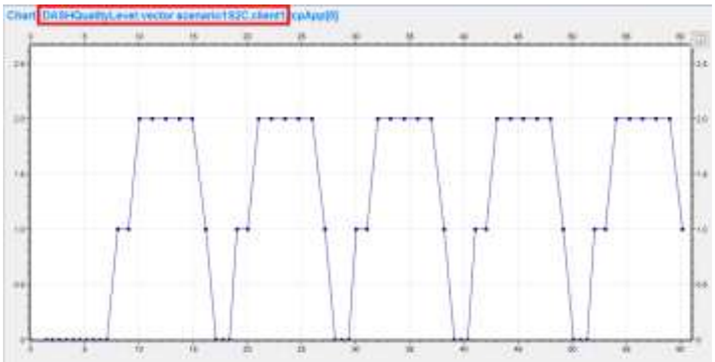

Figure10. Carried traffic load buffer size equals to 65535, 4095, and 2047 with one additional server and client

In the next scenario, we simply added one additional client and one server in the network, and keep other settings the same. Basically, most of the simulation results of this scenario are similar to scenario 3. Again, if the tcp.advertisedWindow value equals to 4095, there is something different happened. Besides the setting of 4095, we have to realize that both clients would need to share the bandwidth since they all asked for video streaming but use the same links. In this scenario, client1 is connected to server1, and client2 is connected to server2. Therefore, we tried to find out the reasons contributed to the result. Figure 11 is the quality index variation of buffer size equals to 65535, 4095, and 2047.

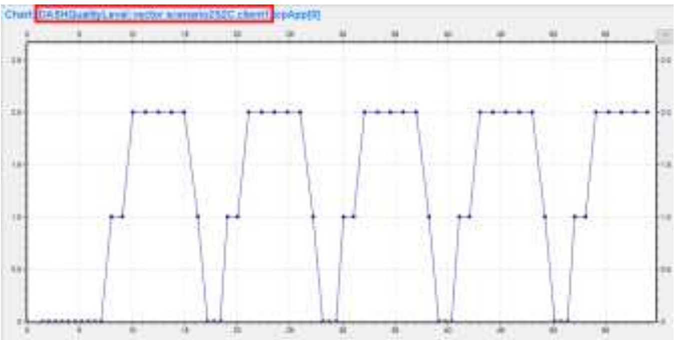

Figure11. Carried traffic load buffer size equals to 65535, 4095, and 2047 with two additional servers and two additional clients

Based on the four scenarios of simulation results, we could say that there are two factors may affect the video buffer status, then contribute to the variation of bitrate and video quality. The first factor is the buffer size. The maximum value of TCP advertised window is 65535, and it is more than ten times of TCP maximum segment size, which is 1452 in this paper. Therefore, the packet transmission is full of space without any congestions or flow control. The video buffer will accumulate new buffer very quickly so it is not easy to trigger the rebuffering size of 8. On the other side, if the TCP advertised window is 2047, which is smaller than double of 1452, the buffer could only store one segment. Therefore, the video buffer cannot achieve full buffer status since the accumulation of data is not fast enough. The size of 4095 is special. It is double more than 1452 but smaller than three times of 1452. Any size which is three times bigger than 1452 would get the same result as 65535. We can see that the video quality is unstable and keep changing. We think that when the quality index goes down to 1, the video buffer should rebuffer in a short time. However, since the buffer can only accept two segments, the consumption is still bigger than accumulation in video buffer when the quality index is 1. Therefore, it needs to go back to index 0 and rebuffer again with less consumption.

The second factor is the number of clients. We can see that when TCP advertised window is equal to 4095, only scenario 3 and scenario 4 have a big impact on the quality index. In our assumption, all of the scenarios should have the same effect. Obviously, the number of clients is the main factor lead to the result. By comparing the network of scenario 3 and scenario 4, we exclude the reasons of mutual effect and packet loss because client1 and client2 are receiving their own packets individually. Hence, we think one of the reason is that the data received is only half in the same period of time since they have to share the bandwidth, and it would definitely cause to slow accumulation of video buffer. The other reason is that since there are two different destinations, the router may have some internal processes to handle the traffic. Since the simulation time of two clients' scenario is much longer than one client scenario, it may indeed affect the judgment.

## Conclusion

In this paper, an implementation of traffic shaping on adaptive bitrate streaming has been presented. As a result of this implementation, the performance of the content-delivery network has been improved drastically making the network more efficient, less delay, better throughput, low packet loss. Specifically, a substantial improvement has been shown using the technique on the adaptive bitrate streaming. We analyzed to the performance of the video streaming between two different nodes, in this case streaming server at the CDN side and streaming client. We implemented various bitrates of video streaming to consider different scenarios and analyzed the end-to-end delays. Traffic shaping was implemented in order to enhance the performance and reduce the latency between the server and client. The end to end delay of various bitrate streaming without traffic shaping was also compared to end to end delay of traffic shaping with traffic shaping and the reduced latency was shown through graphs.

## References

Lammle, T. (2011). CCNA Cisco Certified Network Associate Deluxe Study Guide. John Wiley & Sons.

Chan, Design and Implementation of Quality Optimization Strategy in Adaptive Streaming, 1st ed. Taipei, Taiwan: National Taipei University of Technology, 2013.

Rao, A., Legout, A., Lim, Y. S., Towsley, D., Barakat, C., & Dabbous, W. (2011, December). Network characteristics of video streaming traffic. In Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies (p. 25). ACM.

Rao, A., Legout, A., Lim, Y. S., Towsley, D., Barakat, C., & Dabbous, W. (2011, December). Network characteristics of video streaming traffic. In Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies (p. 25). ACM.

Mir, N. F. (2014). Computer and communication networks. Pearson Education.

E. Kirmani, "University Support Center : Riverbed Technologies," 2016. [Online]. Available: https://splash.riverbed.com/docs/DOC-5391.

G. O. Mosongo, "Simulation Of Multi-Protocol Label Switching," 2009.

Chan, K., Baker, F., & Babiarz, J. RFC 4594 Configuration Guidelines for DiffServ Service Classes.