Webology (ISSN: 1735-188X)
Volume 19, Number 2, 2022

A Low Latency Adaptive Video Streaming Framework To Control The Congestion And Reduce The Switching Times Between Quality Levels

Wafa A. Alqhtani[1] , Ashraf A. Taha[2] , Maazen S. Alsabaan[3]

[1,3]Department of Computer Engineering, King Saud University, Riyadh, CO 11543, Saudi Arabia.

[2]Department of Computer Networks, City of Scientific Research and Technological Applications SRTA-City, Alexandria CO 21934, Egypt.

Abstract-- Video streaming over the Internet is considered the most essential communication technology in recent years, especially with the spread of the SARS-CoV2 pandemic where the communication between people and the work via online videos. This technology faces various challenges that may be obstacles for the customer and the service provider to deliver videos with a flexible quality and without interruption. The high congestion value of the frames on the buffer of the client is one of these challenges. Consequently, the video latency is increased, and thus reduced the video throughput with more interruptions and low quality. The paper is suggested an adaptive video streaming framework by introducing a low latency in an online application to control the congestion and reduce switching times between levels to make video streaming more seamless. The low latency scheme aims to increase the bitrates to optimal values to avoid the congestion of frames on the buffer with a minimum delay value. Analyzing the video load based on three Parameters: latency, Buffering, and throughput of the dynamic system will control the congestion to reduce the switching between quality levels and consequently improve the performance.

Keywords: adaptive video streaming; latency; congestion; buffering; throughput; quality switches.

I. Introduction

With the growth of technology, service providers must develop methods that satisfy the user and with a preferred quality. Increasing the video viewership that satisfies the Quality of Service (QoS) requirements is considered one of the most important challenges that the user faces by running different video streaming applications smoothly and without interruptions. Applying video streaming over the internet has multiple challenges such as video interruptions, video quality, delay time, and switching levels. To address these challenges, we need to adapt video streaming by using control algorithms that help to stream videos in smoother, better quality, and improve variable network parameters. The video streaming application has two types; Video on Demand (VoD) and Live Streaming. VoD transmits its contents into chunks (fragments) over the internet and allows the receiver to decode and playback video. Smooth streaming helps the user to view the video delivered at the desired time without waiting for the entire video while live (real-time) streaming may spread the video contents transmitted using fragments simultaneously to all users at the same time as they are watched by the users or in the lowest delay time. Adaptive video streaming has three main protocols; HTTP live streaming, Microsoft smooth streaming, and Hypertext Transfer Protocol (HTTP) Dynamic Streaming. The latest standard for Moving Picture Expert Group (MPEG) is established and the widely used multimedia standard is Dynamic Adaptive Streaming over HTTP (DASH) or MPEG- DASH which is a challenge to solve the difficulties of media delivery to multiple devices. To play the video contents, the DASH client acquires a Media Presentation Description (MPD) which can be provided using HTTP, email, broadcast, or other transports protocols. By parsing the MPD file, the DASH client identifies the program schedule, media-content accessibility, data types, resolutions, minimum and maximum bandwidths, encoding technology, component positions on the network, and other content features. Using this information, the DASH client chooses the appropriate encoded alternative and streams the segments using HTTP GET demands. The client continues fetching the consequent segments and monitoring the network bandwidth variations after suitable buffering is permitted for network throughput variations. The client agrees on how to adapt to the offered bandwidth depending on its measurements by fetching segments of different options with lower or higher bitrates (Suzen, 2017). Based on the previous concepts mentioned above, the main contributions of this paper are:

- Improve the quality of video streaming under network load changes, by determining some challenges that video streaming faces such as Latency, buffering, and throughput and its video quality effecting and video streaming investigating with control congestion to provide video with suitable quality concerning network parameters variations.
- Develop an adaptive video streaming framework that defines the optimal average bit rate corresponding to the appropriate quality switching level, by designing a Low Latency in a video streaming delivery to balance the average video bit rate to the proper quality switches with the transition to higher switches more seamless.
- Propose a system framework to decrease Latency and buffering values by controlling congestion to adapt the quality switching.

Our research can adjust the video quality switch to the optimal bit rate level using low latency with a limitation of (about 1 sec) for DASH Streaming [2]. When it is closed to the limit value, there will be more buffering due to congestion and low throughput. Imposing these conditions of control congestion and switching between quality levels to balance between bitrates and switching quality to adapt video streaming smoothly.

The organization of the paper is as follows; Section II presents a literature review. Section III presents the proposed quality adaptation framework and the performance parameters. Section IV describes the simulation and test scenario. Section V discusses the results and discussions. Section VI briefs the paper's conclusion and recommends future works.

II. LITERATURE REVIEW

In (Jeroen, et. al., 2018), the authors recognized the problem of reducing the video contents start-up as little as possible, as they improved the Quality of Experience (Qo E) with four frames which include the video contents delivery with a short latency. A large amount of bandwidth is needed for the 360◦ videos, and to reduce the bandwidth must use Field of Video (Fo V) adaptive streaming by coding predicted Fo V region at a higher quality or lower quality, this will develop 360◦ videos streaming on low latency and Fo V-adaptive coding. This approach faces multiple challenges: First, the coded predictive on Fo V may not be aligned in successive frames, this causes long prediction periods, so, must use accurate rate control and minimize the latency. Second, how to update all the regions in a frame continuously because this causes the possibility of error propagation, and thus the quality and accuracy of time prediction decrease. Therefore, we can meet these challenges by using adaptive coding and rate control.

In (Yixiang, et. al., 2020), the paper proposes and develops a proposal for coding and adaptive streaming Fo V with low latency and high-efficiency video coding, by using the coding at the granularity of frames, not of segments, regions will be coded using time prediction at different rates, they also use encoding for each frame to reduce errors and reduce latency as shown in figure 1. Where dark grey: PF tiles, coded at the rate Re. Light grey: PF+ and RI tiles, coded at the rate Rb. Green: user's Fo V, which falls in PF, PF+, RI, and un-coded tiles. They also modeled the quality rate for the coding regions based on the time prediction values, and they modeled the quality rate for the non-coded regions using the errors resulting from the coding, and they used improved Q-R modules with BW and Fo V prediction accuracy, to adapt the bits assigned to the coding regions. The results of this proposal reduce the average latency to less than 100 ms with different Bandwidth and contributed to improving the quality of the average WS-PSNR by 2 to 10 dB.

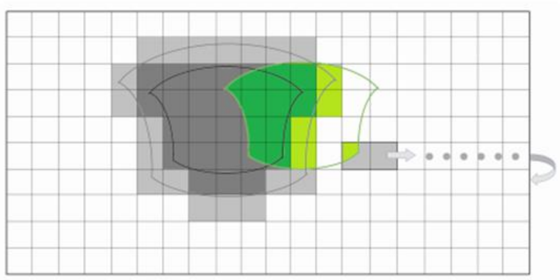


Figure 1. Different regions in a frame using the ERP format.

In (Mariem, 2019), a framework to avoid rebuffering and delay is proposed by using HTTP/2 characterizes of priority frames delivery with bandwidth fluctuation to encode the frames in different HTTP/2 streams and develop, model, and select the essential frame. Algorithms are used to evaluate and test real network and video traces. A model of 360◦ video delivery is provided using client demand tiles in 1 sec and regulating its unsuitable decisions 500 ms later. So, they use HTTP/2 priority, weight, and stream features by sending each tile in different HTTP/2 streams, then using algorithms that use different weight approaches to schedule the video delivery of tiles.

Video streaming requires high user QoE by using high Bitrate to avoid interruptions, rate fluctuations, and Latency. The design of low latency of live video streaming by generating dynamic models and control procedures is presented in (Liyang. et. al., 2020). They adapt video bitrate with low latency on a dynamic network and minimize latency by using Model Predictive Control (MPC) framework. The method has divided the segment (1sec) into multiple chunks (200 ms) then encoding and streaming, then capture the information of video rate, video buffer, latency, and video freeze. After that, they choose the optimal solution of video-rate in bandwidth estimation. They evaluate the performance of the models and algorithms by simulating real network traces, which they can develop the performance with a latency of 2 to 5 seconds.

In (Wafa, et. al., 2020), the authors proposed a method to transmit video smoothly using a quality adaptation algorithm that depends on increasing bitrates and minimizing switching between quality levels, which will give good quality. They analyze the performance of bitrates by control system parameters CPU usage, and Memory Usage, the results have minimized switching between the quality of level by computing the limitation of the parameters (CPU and Memory) to adapt video streaming more efficiently.

On the sensor device, there is a time between generating the data from the sensor and collecting it on application, this time will be long because of the network, using the Internet of Things (IoT) application may develop the system of the sensor device to produce services with a time guarantee. In (Adnan, 2019), the proposed system can guarantee time requests with IoT applications to improve data delivery with less latency. They develop three algorithms Initiate the graph, Random request, and Pull/Push request, then simulate node interactions on all algorithms by sending a request and receiving a reply. Finally, they use Pull and Push requests to find updated information on low latency. They find the best delivery with less latency with the clustered graph by using pull and push phases, but on random phase, the hybrid graph is the best delivery with less latency.

In the last two years, using live video streaming is increased because of the spread of covid-19. So, the client needs to have a high Qo E and less latency with adapting to network changes. One of the streaming services is HTTP Adaptive Streaming, which depends on MPEG-DASH techniques, which use the DASH Available Bit Rate (ABR) algorithm to increase Qo E. In (Craig, et. al., 2020), They propose Standard Low Latency Video Control with ABR algorithm and calculate standard deviation and mean of bandwidth and latency by using a sliding window. They evaluate Stallion and DASH dynamic algorithms on network changes. Stallion increase bitrate (throughput) so there is less latency.

In [10], they have written reports which include the status of live broadcasting, the importance of raising the level of quality and its development, and the most important applications used, which constitute the largest percentage in the world such as live sports, and there are dedicated and important applications such as live trade or health care, which a video streaming Adaptive is important. This report also clarifies that 3-sec latency is the ideal solution, which will be more available in 2020. Decreasing the time to reach it faster can be attained by lowering the latency.

Using HTTP chunked transfer encoding will reduce latency and maximize QoE of video streaming. On (Abdelhak, et. al., 2019), Propose a framework to measure bandwidth and predict low latency streaming by using a sliding window depending on the moving average method to compute chunk-level bandwidth, online linear adaptive filter to predict bandwidth based on an adaptive recursive smallest amount of squares (RLS) algorithm, and ABR control to calculate and select optimal bitrate. The results achieve reduce of latency from 2.3 to 3 sec. and high-performance values of bitrates and QoE, see figure 2.

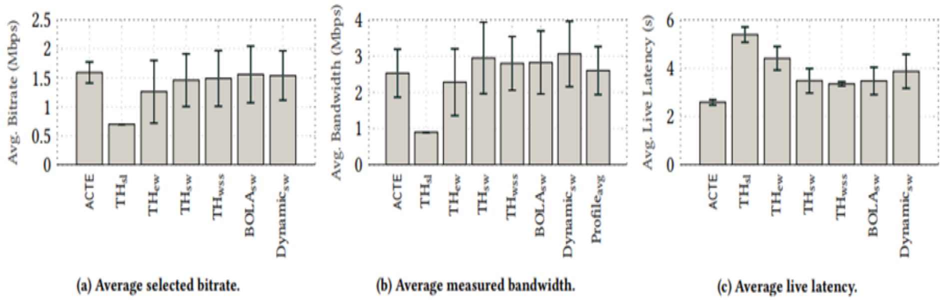


Figure 2. Average selected bitrate, measured bandwidth, and live latency for different ABR over 10 runs. (Abdelhak, et. al., 2019)

In (Mary, et. al., 2015), the authors introduce an algorithm for quality adaptation to regulate video streaming quality levels for every user by measuring a variation in the channel experienced by that user. The algorithm periodically measures the throughput of the channel and analyses throughput variance between pairs of consecutive measurements. A decision to select the quality is made by comparing the channel variance to a threshold. Through extensive testing using real datasets, representing multiple video sessions, the paper demonstrates that the proposed algorithm reduces the number of quality switching decisions made with keeping a high average bitrate regarding other adaptation schemes.

In the paper (Bin, et. al., 2021) using the HTTP Live Streaming (HLS) protocol, the effective adaptive video streaming model has been designed. This model was evaluated based on studying the effect of the difference in clip size on productivity, response time, number of pause events, and frequency of bitrate, and then they analyzed data using simulation. This study achieved high productivity and lower response time using the video adaptive streaming network.

The interest of recent studies in raising the quality of video streaming for the client has increased on low-latency streaming. For standard low-latency video control, authors in (Gutterman, et. al., 2020) design a novel scheme of adaptive bitrate (ABR), Stallion, using a sliding window. This scheme is used to measure the mean and standard deviation of the bandwidth and latency to calculate and compare with the dynamic algorithm of standard DASH corresponding to different conditions. Stallion shows nearly twice the value of bitrate and four times of reduction in the number of stalls.

On the other hand, in (Lim et. al., 2020), we see that they have developed the ABR selector by playback control and throughput measurement modules, that improve the quality of users' experience using the Low on Latency Model. These evaluations show the module of throughput measurement performs better than the default implementation provided by using dash.js.

III. Proposed Framework

A Low Latency with a congestion control framework is proposed to achieve a balance between Bitrates and quality switches by decreasing latency and Buffering will make the video stream more seamless without interruption using the dynamic system to control performance parameters and make the algorithm more achievable. Using the MPEG-DASH model of media transfer in the adaptive video streaming over HTTP, the server delivers the segments of video contents to the client by using an MPD file which contains multiple intervals; each interval has a period with the starting time and consists of one or multiple adaptations set which provides data about media components and its different encoded varieties. The adaptation set has multiple representations and each has an encoded alternative of a similar media component. These representations change from others by bitrate, resolution, number of channels, or other features. The representation has one or more segments, media stream fragments in temporal sequence. The segment has a URL addressable location on a server that can be downloaded by HTTP GET with byte ranges. Also, the client has DASH Control and player, it controls the content delivered from the server by parsing the MPD files to play it on the player using the information in the segment presentation. The multimedia content is sent over HTTP protocol and MPEG encoded using MPEG-DASH technology, MPEG is the standard digital container layout for transmission with multimedia storage.

In the system model of DASH streaming, we obtain the MPD file on the server-side and use Apache Jmeter with video streaming plugin on the client-side to parse MPD file and analyze the performance parameters to collect and analyze media content transfer and access to control parameters, latency, congestion, and throughput, which makes the video more smoothly and without delay or interruption, thus raising the quality and achieving a balance between the average bit rate and the switching quality levels to adapt video streaming.

PERFORMANCE PARAMETERS:

Video over the internet has multiple challenges for performance and the dynamic system has many variables of performance parameters. Our system will focus on the congestion control that affected and increased the latency and buffering. Latency is a time of delay from request to response and the Latency of HTTP video streaming limitation is between 1 to 5 seconds, but in live streaming is less than 1 second. The time between video requests to display there are serval processes: encoding, buffering, bandwidth, and data transmission, that’s an increase Latency. On VoD the client can use the Buffering technologies to be enhanced QoS, but this may increase the latency if there is congestion on the buffer.

In our system, using the analyzing tool to obtain latency and throughput of video content. This information will use it to calculate the number of frames that will buffer as in eq. (1) and compute if the value of latency exceeds 1 sec. or not, to control the congestion. After control congestion will produce a proposed strategy to balance between average bitrates and quality switching levels, to reduce switching quality levels, and Maximize throughput as in eq. (2) to make the video stream more smoothly without delay or interruptions.[2]

$$\text{Buffering (number of Frames)} = \frac{\text{Limit of Latency (1000 ms)}}{\text{Value of Latency per frame}} \qquad (1)$$

$$\text{Throuphput} = \frac{\text{Number of Requestes}}{\text{Total Time}} \qquad (2)$$

IV.SIMULATION AND TEST SCENARIO:

Our objective is to design a Low Latency framework with control congestion to adapt quality switching levels. The Performance parameters such as Latency, Buffering, and Throughput will be studied to achieve this purpose. If latency increases, more frames buffered so will have congestion, so maximizing the average bitrates for the current quality level using control congestion before shifting to a higher level. The CPU and memory usage results of all adaptation sets (according to the bitrates in the MPD file) neglect the parameters that have the biggest values from the CPU and memory usages limitation, these results are shown in the study (Wafa, et. al., 2020). Now, other results of Latency, Buffering, and throughput averages of all adaptation sets will be shown. The resolution value has multiple bit rates; every bit rate has a value that affects Latency. 320 x 240 resolution takes 922.1714 ms latency, however, at a higher resolution of 480 x 360, we observed 491.3714 ms of latency. Switching to the higher resolution uses less time latency, we can observe that an 854x480 and 1280x720, and 1920x1080 resolution has less value of latency and more throughput, which means that smaller resolutions such as 320 x 240 are close to the Latency limits (1 sec).

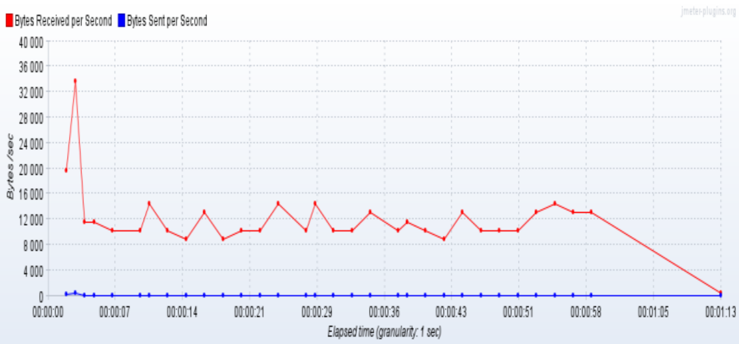
The Latency is increased if the switch to the level of small value bitrate, this will cause to be congestion because there are no buffering frames on the client-side. Throughput will be reduced from a higher level of bitrate to a smaller level of bitrate due to the congestion that occurred after the high latency value. See Figures (3 to 14).

Table 1. Parameters Values of Adaptation Set.

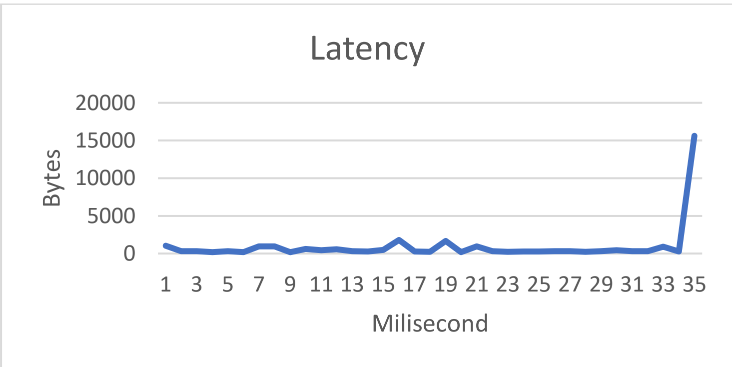
Resolutions	Data rates	CPU Usage	Memory Usage	Average of Latency	Buffering	Throughput
320 x 240	47 kbps 92 kbps 135 kbps	8 % 8 % 9 %	120 MB 126 MB 1126 MB	922.1714 ms 491.3714 ms 513.7222 ms	1 frame 1 frame 1 frame	0.48866 0.52328 0.55338
480 x 360	182 kbps 226 kbps 270 kbps 353 kbps 425 kbps	12% 12% 13% 13% 13%	117 MB 117 MB 125 MB 127 MB 127 MB	505.975 ms 524.2683 ms 580.9722 ms 520.5 ms 623.6923 ms	1 frame 1 frame, sub-frame 1 frame sub-frame 2 frames	0.55338 0.55734 0.50424 0.51487 0.52291
854 x 480	538 kbps 621 kbps	19% 20%	132 MB 143 MB	603.5833 ms 552.8684 ms	2 frames 2 frames	0.5019 0.52582

1280 x 720	808 kbps 101, 103, 17 Mbps	39% up 42%	161 MB Up 210 MB	622.6944 ms 655.8056, 502.1944, 838.7222 ms	2 frames 2 frames 2 frames 2 frames. sub-frame	0.50848 0.49793 0.50958 0.47741
1920 x 1080	202, 206, 303, 308, 402, 407 Mbps	Up 60%	Up 230 MB	606.3611 ms	Up 2 frames	0.50682

The throughput average of the resolution value 320x240 of bitrate 47.0kbp is decreased due to the high latency value that occurred from the congestion of frames on the client. On another side, the latency average value on this resolution is increased due to the small bitrate and congestion, see figure 3.

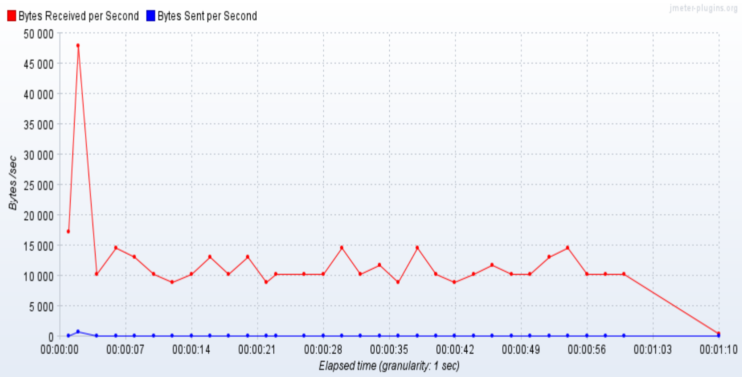


A: Throughput, 320x240 47.0kbps.

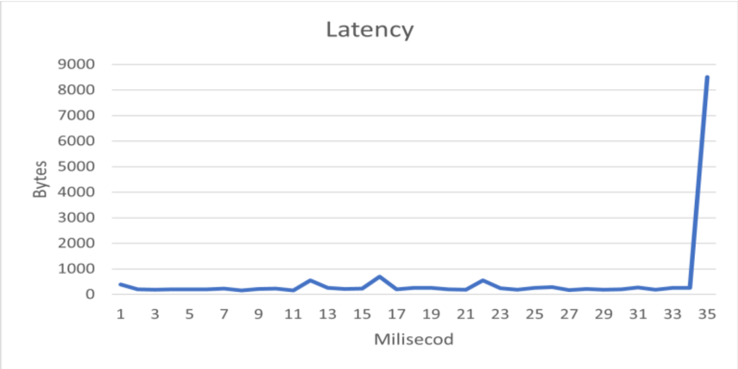


B: Latency, 320x240 47.0kbps.
Figure 3. Throughput and Latency Values of 320x240 47.0kbps

At the next level of the adaptation set, the throughput average of the resolution value 320x240 of bitrate 92.0kbp is decreased due to the high latency value that occurred from the congestion of frames on the client. On another side, the latency average value on this resolution is increased due to the small bitrate and congestion, the latency values at this level differ from the higher latency values in the previous level, see figure 4.



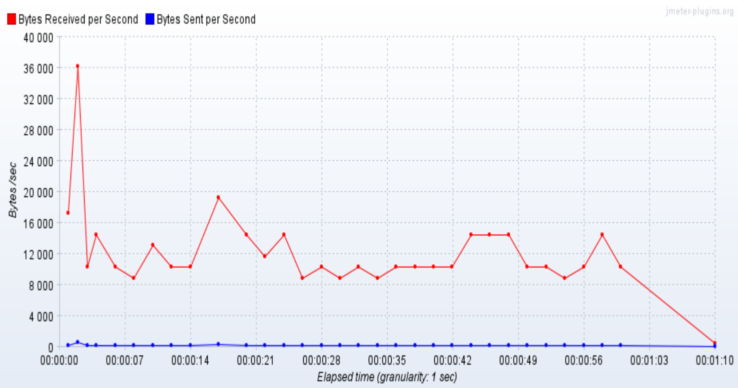
A: Throughput, 320x240 92.0kbps



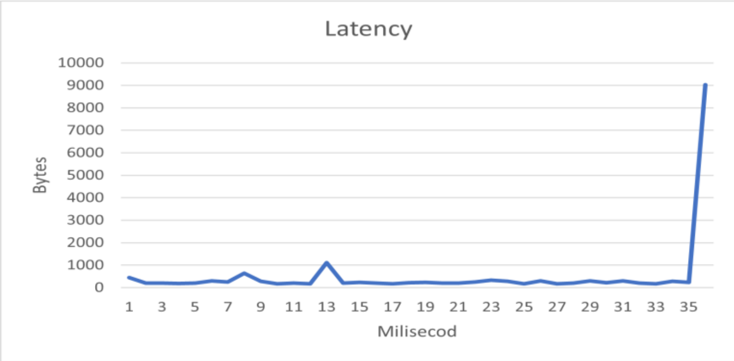
B: Latency, 320x240 92.0kbps

Figure 4. Throughput and Latency Values of 320x240 92.0kbps

At the next level of the adaptation set, the throughput average of the resolution value 320x240 of bitrate 135.0kbp is decreased due to the high latency value that occurred from the congestion of frames on the client, the throughput values at this level differ from the smaller throughput values in the previous level. On another side, the Latency average value on this resolution is increased due to the small bitrate and congestion, see figure 5.



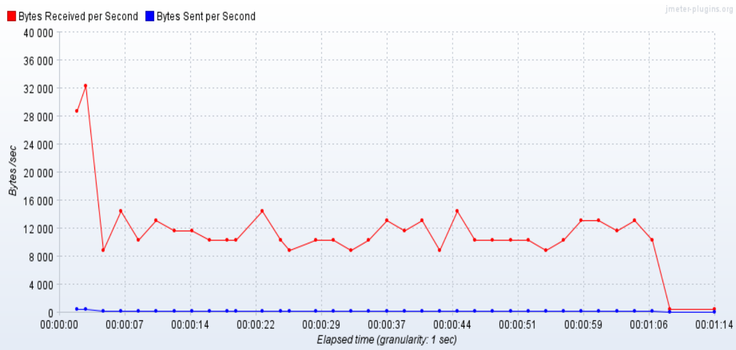
A: Throughput, 320x240 135.0kbps



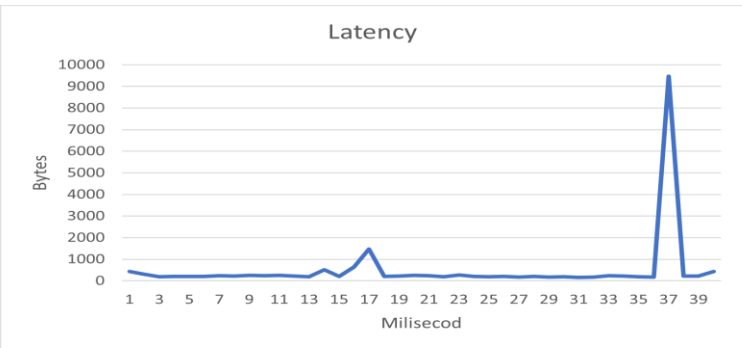
B: Latency, 320x240 135.0kbps

Figure 5. Throughput and Latency Values of 320x240 135.0kbps

The throughput average for the next level of the resolution value 480x360 182 of bitrate 182.0kbp is decreased due to the high latency value that occurred from the congestion of frames on the client, the throughput values at this level differ from throughput values in the previous level. On another side, the latency average value on this resolution is increased due to the small bitrate and congestion, the latency values at this level differ from the higher latency values in the previous level, See figure 6.



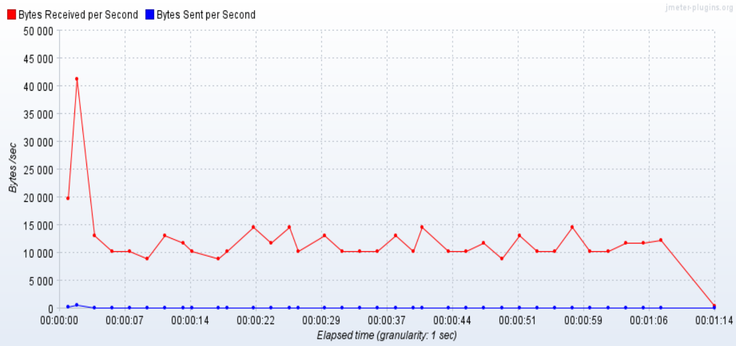
A: Throughput, 480x360 182.0kbps



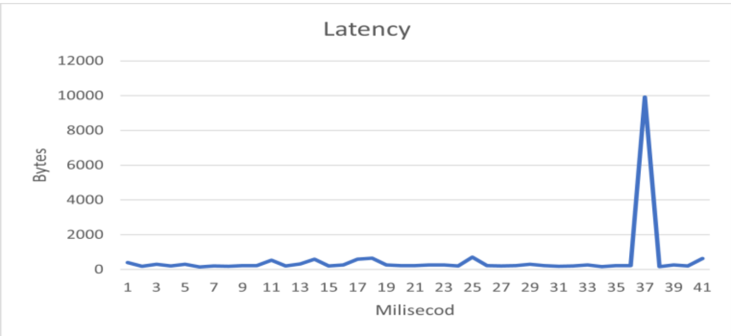
B: Latency, 480x360 182.0kbps

Figure 6. Throughput and Latency Values of 480x360 182.0kbps

If we see throughput average for the next level of the resolution value 480x360 226.0kbps is decreased due to the high latency value that occurred from the congestion of frames on the client, the throughput values at this level differ from throughput values in the previous level. On another side we can see Latency average value on this resolution is increased due to the small bitrate and congestion, the latency values at this level differ from the higher latency values in the previous level, See figure 7.



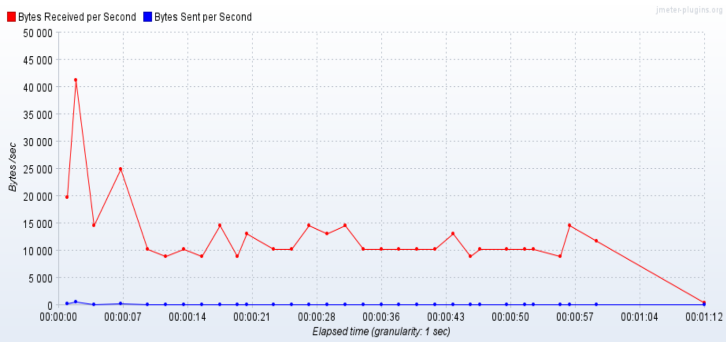
A: Throughput, 480x360 226.0kbps



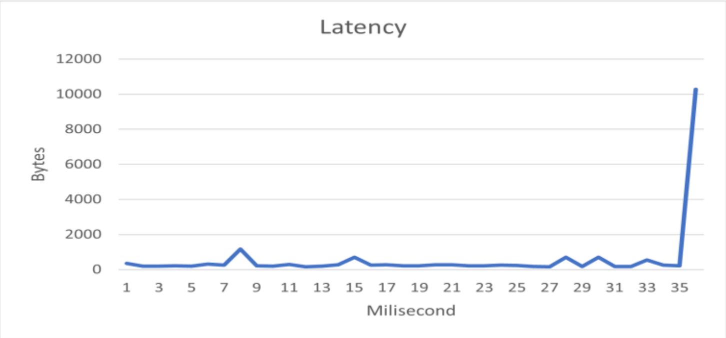
B: Latency, 480x360 226.0kbps

Figure 7. Throughput and Latency values of 480x360 226.0kbps

The throughput average is increased rather than the previous level because the average of latency values of resolution 480x360 270.0kbps is less than the previous level, see figure 8.



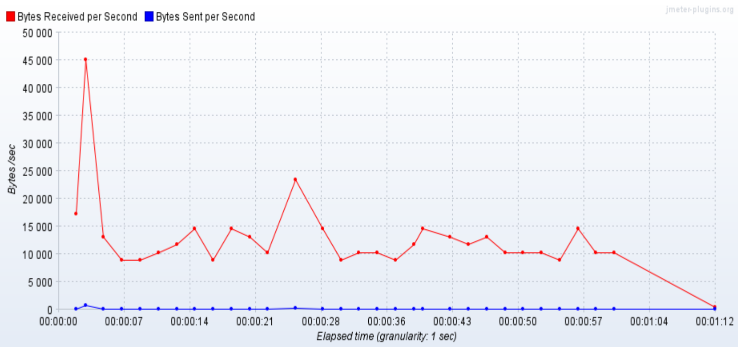
A: Throughput, 480x360 270.0kbps



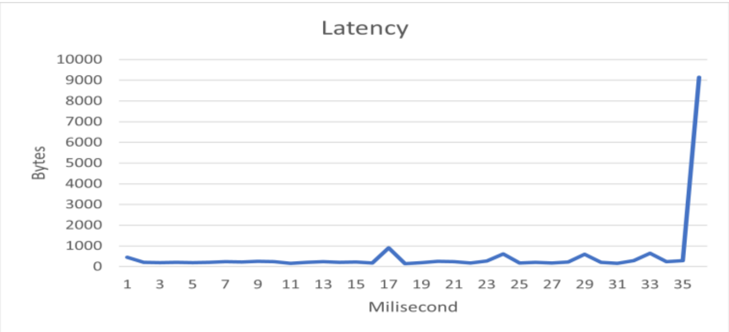
B: Latency, 480x360 270.0kbps

Figure 8. Throughput and Latency values of 480x360 270.0kbps

For the next level, see a resolution of 480x360 353.0 kbps that has more throughput and less latency average value, rather than the previous level, see figure 9.



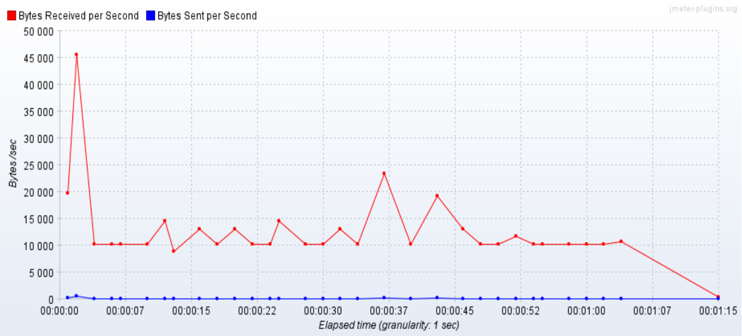
A: Throughput, 480x360 353.0kbps



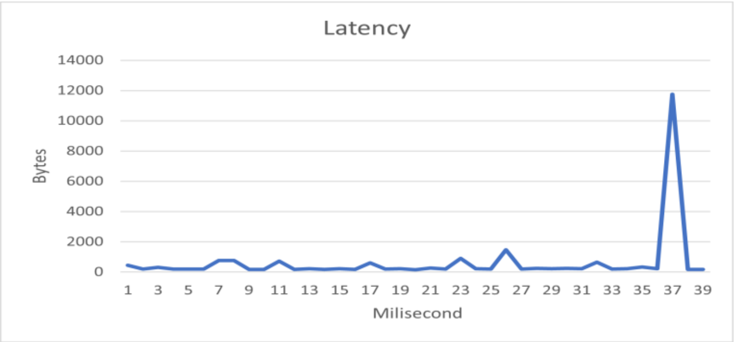
B: Latency, 480x360 353.0kbps

Figure 9. Throughput and Latency values of 480x360 353.0kbps

On 480x360 425.0kbps level, throughput is increased more than the previous level, due to the low latency that decreased when switching from a low level of bitrate to a higher bitrate level. See figure 10.



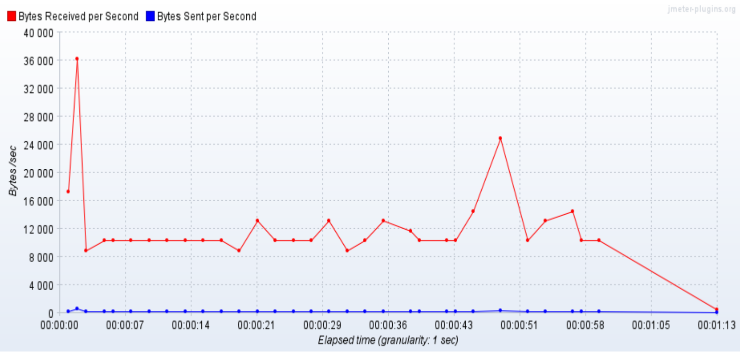
A: Throughput, 480x360 425.0kbps



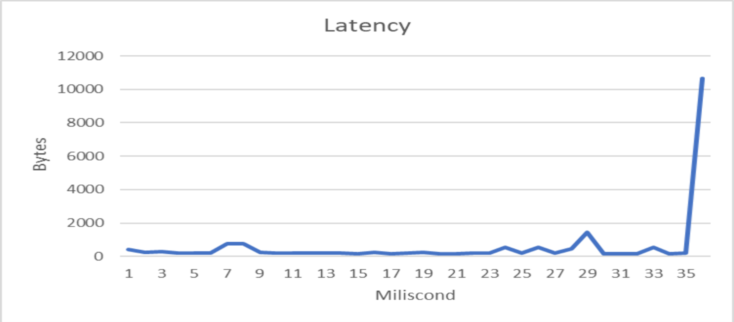
B: Latency, 480x360 425.0kbps

Figure 10. Throughput and Latency values of 480x360 425.0kbps

On 854x480 538.0kbps level has more throughput due to a high level of bitrate rather than the previous level and the latency average is decreased, see figure 11.



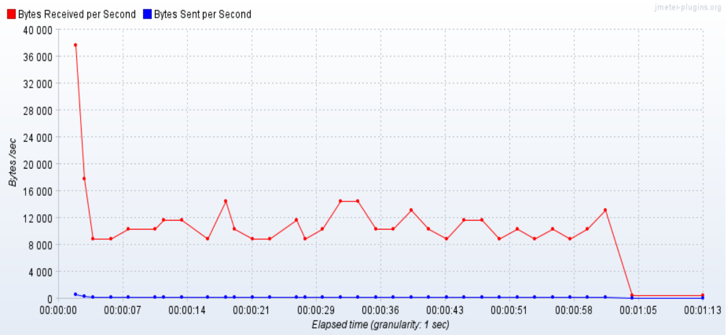
A: Throughput, 854x480 538.0kbps



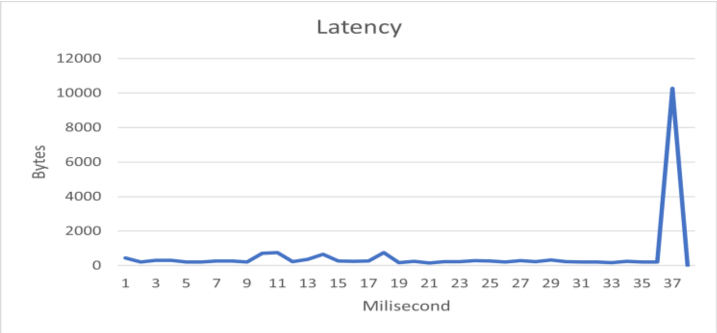
B: Latency, 854x480 538.0kbps

Figure 11. Throughput and Latency values of 854x480 538.0kbps

In the 854x480 621.0 kbps level, we have more throughput and less latency average of frames, which means if switching to a higher level, the throughput increases, and latency decreases, see figure 12.

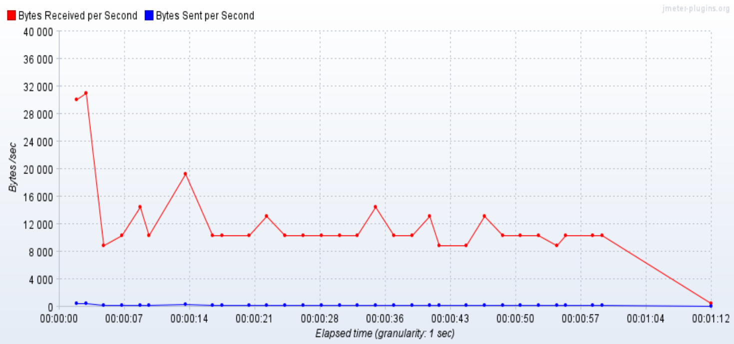


A: Throughput, 854x480 621.0kbps

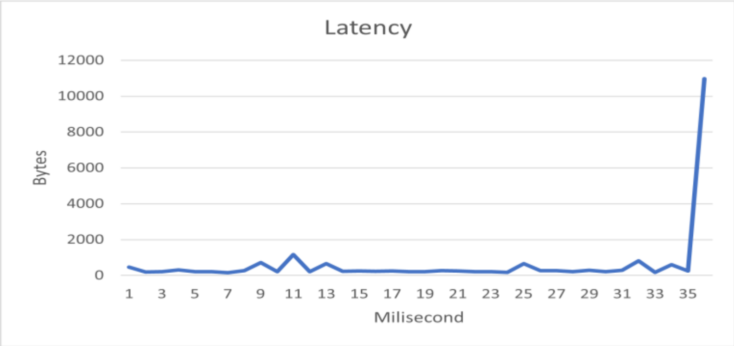


B: Latency, 854x480 621.0kbps
Figure 12. Throughput and Latency of 854x480 621.0kbps

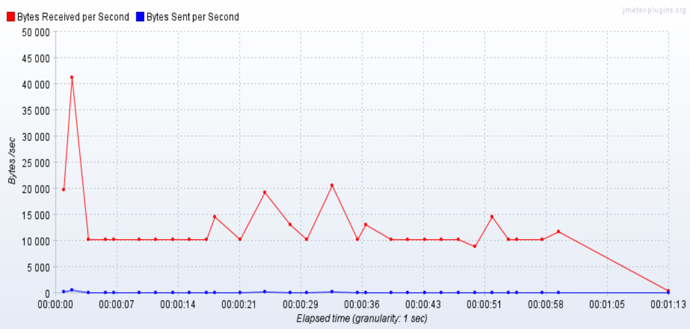
In levels of resolution 1280x720 and bitrates (808kbps, 1.1Mbps, 1.3Mbps, 1.7Mbps), the throughput measures are increased more than one of each level, if the switch from 808kbps to 1.7Mbps we see throughput increased and the latency decreased because have a high range of bitrates, see figure 13 (A to H).



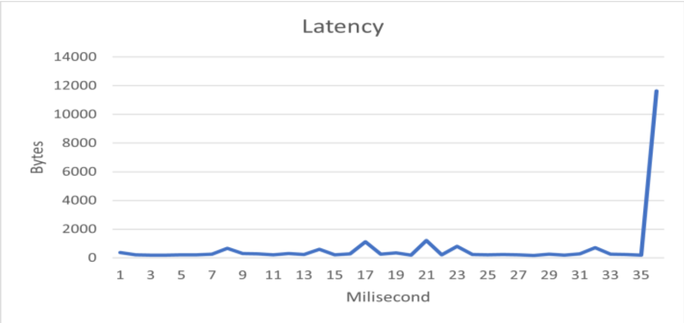
A: Throughput, 1280x720 808.0kbps



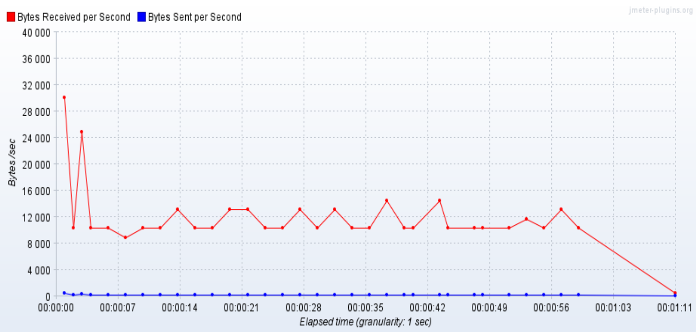
B: Latency, 1280x720 808.0kbps



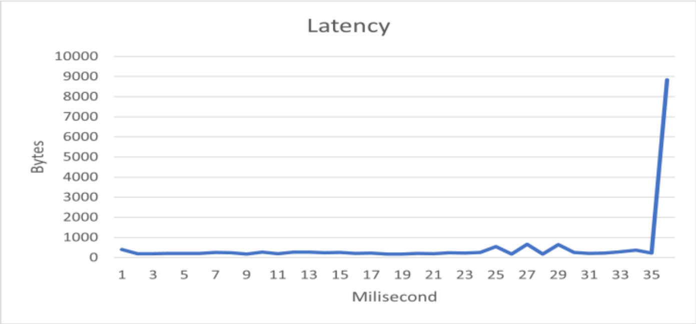
C: Throughput, 1280x720 1.1Mbps



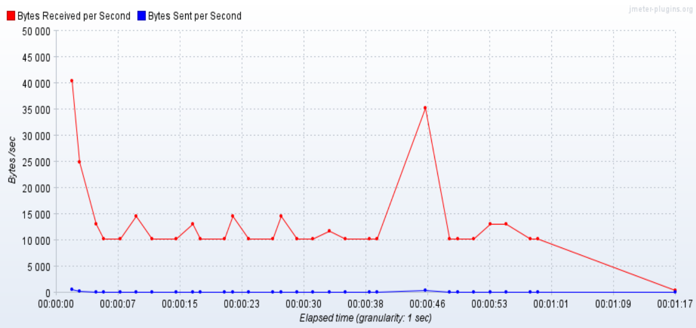
D: Latency, 1280x720 1.1Mbps



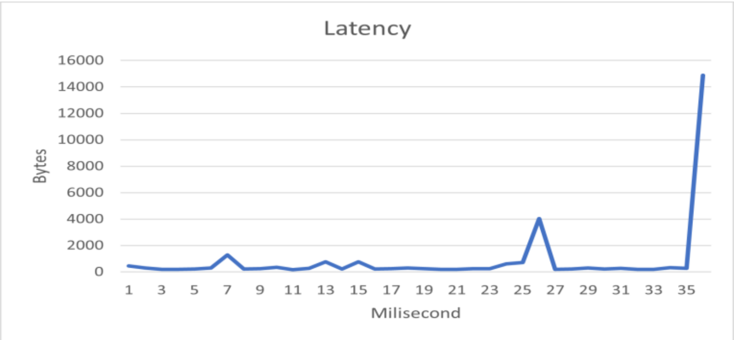
E: Throughput, 1280x720 1.3Mbps



F: Latency, 1280x720 1.3Mbps



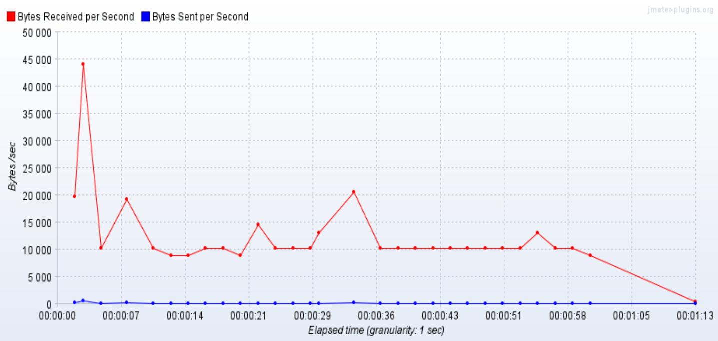
G: Throughput, 1280x720 1.7Mbps



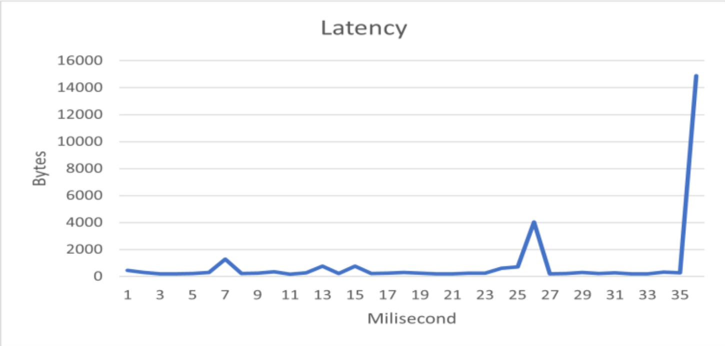
H: Latency, 1280x720 1.7Mbps

Figure 13. Throughput and Latency of 1280x720, (808kbps, 1.1Mbps, 1.3Mbps, 1.7Mbps)

Next level we have a higher level of throughput and low latency due to the high range of bitrate, rather than previous levels on the adaptation set, see figure 14.



A: Throughput, 1920x1080 2.2Mbps



B: Latency, 1920x1080 2.2Mbps
Figure 14. Throughput and Latency values of 1920x1080 2.2Mbps

V. RESULTS AND DISCUSSIONS

Achieving the goal by investigating the video load changes on the parameters of performance, such as Low Latency, that have a high impact on multimedia quality and control congestion by maximizing the average bit rates and reducing buffering for each level of quality. Controlling the congestion and lowering the latency can be analyzed for the highest and lowest value of each data rate in resolution level to reduce these levels by joining the closest results to the closest level, as organized in Table (2). Making the streaming video more seamless we can trade-off between low latency and quality as seen in figure 15. This figure illustrates the adaptive framework flowchart procedures. In our previous paper (Wafa, et. al., 2020), we show the bitrates differences before and after the study of minimizing levels of quality to reduce the transferring between switching levels. Dividing the adaptation set into three levels with a group of quality and data rates which have closed results on performance parameters such as (CPU and memory usages). After studying the results of CPU usage and memory usage, it is found that there are only three levels on the adaptation set, namely high level (HL), middle level (ML), and low level (LL), as organized in Table (3). Now in this study, we divided the adaptation set into three levels only; each group of quality and data rates has results on (CPU usage, Memory Usage, and Low Latency) parameters.

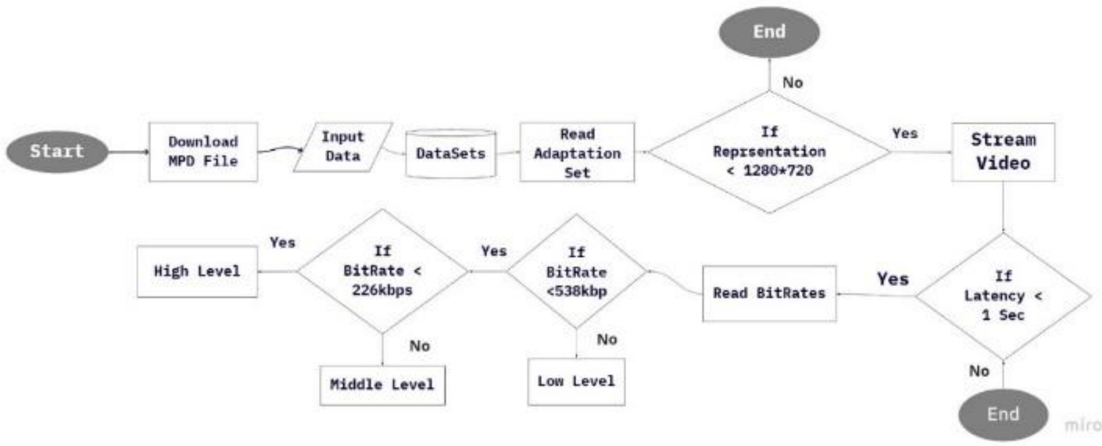


Figure 15. Adaptive Framework.

Table 2. Adaptation Set Levels Before Proposed Method.

Resolution	Bit Rates (Before)	Bit Rates (After)
320 x 240	47 kbps	92 to 135 kbps 480x360, 182 kbps
	92 kbps	
	135 kbps	
480 x 360	182 kbps	480x360, 226 to 425 kbps
	226 kbps	
	270 kbps	
	353 kbps	
	425 kbps	
854 x 480	538 kbps	854x480, 538 to 621 kbps
	621 kbps	

Table 3. Adaptation Levels After proposed method.

Levels	Adaptation Set
High Level (HL)	320 x 240, 92 to 135 kbps 480 x 360, 182 kbps
Middle Level (ML)	480 x 360, 226 to 425 kbps
Low Level (LL)	854 x 480, 538 to 621 kbps

The dynamic Adaptive Streaming over HTTP (DASH) model has two parts, first, the DASH server hosts media segments, and each segment is encoded at several bitrate levels listed in the MPD file. Second, the DASH client estimates available bandwidth in the network and uses the information in the playback buffer to choose the highest bitrate for the next segment fetched to avoid video stalls and provide a good trade-off of Qo E trade-off, this method is called bitrate switching. The authors in (Ito, et. al. 2014) analyze the network traffic behavior of two popular adaptive streaming environments, namely Netflix and YouTube, under different varying network conditions and different constraints of bandwidth and packet loss. These players depended on DASH stream control and worked at the network level to compute some metrics such as throughput, packet inter-arrival time, packet size, and the like. In this paper, the authors discussed the behavior of the Netflix and YouTube players and the extent to which network metrics affected their effectiveness and quality. It became clear that both players did not operate well under a small bandwidth and the behavior of these applications under network standards and loss restrictions became poor, see table (4) & (5). Such applications need to improve their adaptation algorithm to obtain Qo E of the customer.

Table 4: Netflix Summary (Ito, et. al. 2014)

Netflix	No cons.	BW	Loss	Delay
Throughput (kbps)	3417	407	752	2293
Pps	343	61	95	238
IAT (s)	0.005	0.053	0.026	0.038
Packet Size (bytes)	1249	794	924	254
RTT (ms)	298	278	277	647

Table 5: YouTube Summary (Ito, et. al. 2014)

YouTube	No cons.	BW	Loss	Delay
Throughput (kbps)	1195	221	821	1204
Pps	122	34	107	130
IAT (s)	0.280	0.527	0.145	0.084
Packet Size (bytes)	1073	529	859	1082
RTT (ms)	64	378	47	443

In general, a decrease in the throughput and an increase in the delay due to the limitations of the network measurements and the low bandwidth. To plan control algorithms that contribute to improving adaptive video streaming. In this paper, we have studied the same performance parameters (Craig, et. al., 2020) on Libdash players by using a control algorithm to improve adaptive video streaming. We used control algorithms based on studying many performance parameters to reduce the switching between quality levels by increasing the bitrates.

VI. CONCLUSION AND FUTURE WORK:

In this paper, we are studying Adaptive video streaming by a control congestion algorithm to stream video seamlessly without interruption and increasing Qo E. We design a Low Latency system by controlling performance parameters (Latency, Buffering, and Throughput) to make the system more efficient by increasing bitrates and minimizing quality switching between levels. We used a simulation program to analyze video load data and control performance parameters, after that we are minimizing quality switching levels to three levels and increased Bitrate value to adapt a video in low latency. In future work, the security feature is added to the proposed framework to make a secure network for video streaming.

Conflicts of Interest: The authors whose names listed at the top of the article certify that they have no affiliation with or in any organization or entity with any funding interest and that, there is no conflict of interest, and there were no funders for the design of the study, the topic of the materials discussed in this manuscript.

References

[1] Suzen Saju Kallungal., “A Survey on Adaptive Video Streaming Technologies,” IJARCET, Advanced Research in Computer Engineering & Technology, vol. 6, issue 3, pp. 350–352, March 2017.

[2] CAST, Inc., Understanding - and Reducing - Latency in Video Compression Systems, https://www.design-reuse.com/articles/33005/understanding-latency-in-video-compression-systems.html.

[3] Jeroen Van, Cedric De, Stefano Pet., Tim Wauters, “Performance Characterization of Low-Latency Adaptive Streaming From Video Portals”, Digital Object Identifier Journal, VOLUME 6, pp. 43039-43055, August 2018.

[4] Yixiang Mao, Liyang Sun, Yong Liu and Yao Wang, “Low-latency FoV-adaptive Coding and Streaming for Interactive 360◦ Video Streaming”, Proceedings of the 28th ACM International Conference on Multimedia, October 2020.

[5] Mariem Ben Yahia, “Low latency video streaming solutions based on HTTP/2”, Ecole nationale supérieure Mines-Télécom Atlantique, 10 Jul 2019.

[6] Liyang Sun, Tongyu Zong, Yong Liu, Yao Wang, and Haihong Zhu, “Optimal Strategies for Live Video Streaming in the Low-latency Regime”, IEEE 27th International Conference on Network Protocols (ICNP), October 2019.

[7] Wafa A. Alqhtani, Ashraf A. Taha and Maazen S. Alsabaan, “An Adaptive Quality Switch-aware Framework for Optimal Bitrate Video Streaming Delivery” International Journal of Advanced Computer Science and Applications(IJACSA), 11(8), 2020. http://dx.doi.org/10.14569/IJACSA.2020.0110871

[8] Adnan Toutounji Alkallas, “Building Distributed Systems for Fresh and Low-latency Data Delivery for Internet of Things”, degree project in computer sciences and engineering, second cycle, Stockholm, Sweden 2019.

[9] Craig Gutterman, Brayn Fridman, Trey Gilliland, Yusheng Hu, Gil Zussman, “STALLION: Video Adaptation Algorithm for Low-Latency Video Streaming”, MMSys '20: Proceedings of the 11th ACM Multimedia Systems Conference, May 2020 Pages 327–332.

[10] WOWZA Media Stream, “2019 Video Streaming Latency Report”, WOWZA.com.

[11] Abdelhak Bentaleb, Christian Timmerer, Ali C. Begen, Roger Zimmermann, “Bandwidth Prediction in Low-Latency Chunked Streaming”, Proceedings of the 29th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, June 2019, Pages 7–13.

[12] Mary Riad, Hatem Abu-Zeid, Hossam S. Hassanein, Mazhar Tayel, Ashraf A.Taha, “A Channel Variation-aware Algorithm for Enhanced Video Streaming Quality”, 40th Annual

IEEE Conference on Local Computer Networks, LCN 2015, Clearwater Beach, Florida, USA, 2015, Pages 893-898.

[13] M. Bin Waheed, F. Jamil, A. Qayyum, H. Jamil, O. Cheikhrouhou, M. Ibrahim, B. Bhushan, and H. Hmam, “A New Efficient Architecture for Adaptive Bit-Rate Video Streaming”, Published on MDPI Journal, Switzerland, 19 April 2021, 13(8), 4541

[14] C. Gutterman, B. Fridman, T. Gilliland, Y. Hu, G. Zussman. “STALLION: Video Adaptation Algorithm for Low-Latency Video Streaming”, Proceedings of the 11th ACM Multimedia Systems Conference, May 2020, Pages 327–332.

[15] M. Lim, N. Akcay, A. Bentaleb, Ali C. Begen and R. Zimmermann, “When They Go High, We Go Low: Low-Latency Live Streaming in dash.js with LoL”, Proceedings on MMSys 11th ACM Multimedia Systems Conference, June 8-11 2020, Istanbul Turkey.

[16] M. Ito, S. Fernands, R. Antonello, D Sadok, "Network Level Characterization of Adaptive Streaming over HTTP Applications", Proceedings - International Symposium on Computers and Communications, June 2014.

Authors Profile:

Wafa Aoudh Alqhtani received a B.Sc. degree in the Networking and Communications system from Princess Nourah University, Saudi Arabia, in 2015. M.Sc. degree in Computer Engineering in the field of Networks from King Saud University, Saudi Arabia, in 2021. She was an Assistant researcher for 3 months at Imam Mohammad bin Saud University. Shae was a teacher corporative at King Khaled University. Her current research interests include Video Streaming.

Ashraf Abdelaziz Taha received the B.Sc. degree in Electronics Engineering from Menoufia University, Egypt, in 1990, the M.Sc. degree in Computer Science and Engineering, from Menoufia University, Egypt, in 2000, and the Ph.D. degree in Electrical Engineering from Alexandria University, Egypt, in 2010, respectively. He is currently a Researcher in the Department of Computer Networks, the City of Scientific Research and Technological Applications (SRTA-City), Egypt. He was an Assistant Professor in the Department of Computer Engineering, King Saud University (KSU) from 2013 to 2019. He had earned (STDF Visiting Grants) in the Speed School of Engineering, Department of Computer Engineering and Computer Science (CECS), Louisville University, Kentucky, USA. His current research interests include Video Streaming over Networks, the Internet of Things, and Wireless Communications and Networking communication.

Maazen Alsabaan received a B.Sc. degree in electrical engineering from King Saud University, Saudi Arabia, in 2004, the M.Sc. and Ph.D. degrees in Electrical and Computer Engineering from the University of Waterloo, Canada, in 2007 and 2013, respectively. He is currently an Assistant Professor in the Department of Computer Engineering, King Saud University (KSU), Saudi Arabia. He was the Chairman of the department from 2015 to 2018. Dr. Alsabaan serves as a consultant in different agencies and awarded many grants from KSU and King Abdulaziz City for Science and Technology (KACST). His current research interests include Wireless Communications and Networking communication, vehicular networks, green communications, intelligent transportation systems, and cybersecurity.