Article

A New Efficient Architecture for Adaptive Bit-Rate Video Streaming

Muhammad Hamza Bin Waheed ^{1,†} , Faisal Jamil ^{2,*} , Amir Qayyum ¹, Harun Jamil ^{3,†}, Omar Cheikhrouhou ^{4,*} , Muhammad Ibrahim ², Bharat Bhushan ⁵ and Habib Hmam ⁶

¹ Department of Computer Science, Capital University of Science and Technology, Islamabad 44000, Pakistan; mhamza@cust.edu.pk (M.H.B.W.); aq@cust.edu.pk (A.Q.)

² Department of Computer Engineering, Jeju National University, Jeju 63243, Korea; ibrahimislamian@gmail.com

³ Department of Electronic Engineering, Jeju National University, Jeju 63243, Korea; harunjamil@hotmail.com

⁴ College of CIT, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia

⁵ School of Engineering and Technology (SET), Sharda University, Uttar Pradesh 201310, India; bharat_bhushan1989@yahoo.com

⁶ Faculty of Engineering, Moncton University, Moncton, NB E1A3E9, Canada; Habib.Hmam@umoncton.ca

* Correspondence: faisal@jeju.ac.kr (F.J.); o.cheikhrouhou@tu.edu.sa (O.C.)

† These authors contributed equally to this work.

Citation: Bin Waheed, M.H.; Jamil, F.; Qayyum, A.; Jamil, H.; Cheikhrouhou, O.; Ibrahim, M.; Bhushan, B.; Hmam, H. A New Efficient Architecture for Adaptive Bit-Rate Video Streaming. *Sustainability* **2021**, *13*, 4541. <https://doi.org/10.3390/su13084541>

Academic Editors: Amir Mosavi, Jaime Lloret and Manuel Fernandez-Veiga

Received: 7 March 2021

Accepted: 14 April 2021

Published: 19 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: The demand for multimedia content over the Internet protocol network is growing exponentially with Internet users' growth. Despite high reliability and well-defined infrastructure for Internet protocol communication, Quality of Experience (QoE) is the primary focus of multimedia users while getting multimedia contents with flawless or smooth video streaming in less time with high availability. Failure to provide satisfactory QoE results in the churning of the viewers. QoE depends on various factors, such as those related to the network infrastructure that significantly affects perceived quality. Furthermore, the video delivery's impact also plays an essential role in the overall QoE that can be made efficient by delivering content through specialized content delivery architectures called Content Delivery Networks (CDNs). This article proposes a design that enables effective and efficient streaming, distribution, and caching multimedia content. Moreover, experiments are carried out for the factors impacting QoE, and their behavior is evaluated. The statistical data is taken from real architecture and analysis. Likewise, we have compared the response time and throughput with the varying segment size in adaptive bitrate video streaming. Moreover, resource usage is also analyzed by incorporating the effect of CPU consumption and energy consumption over segment size, which will be counted as effective efforts for sustainable development of multimedia systems. The proposed architecture is validated and indulged as a core component for video streaming based on the use case of a Mobile IPTV solution for 4G/LTE Users.

Keywords: multimedia; variable bit rate; HAS architecture; video streaming; quality of experience

1. Introduction

Multimedia streaming happens very frequently in this age of the Internet. Video calling and sharing is the most happening activity in this era to have interaction between people in different regions of World, that is most effectively done through HTTP protocol. Such protocols provide a mechanism for adaptive streaming known as Hypertext Transfer Protocol Adaptive Streaming (HAS) [1,2]. This mechanism became famous for using HTTP as its primary method of transportation in Internet applications. Moreover, this protocol is much easier to configure while allowing Network Address Translation [3,4]. Quality of Experience (QoE) is the main goal of service providers in current networks due to similar available consumer options that generate customers hopping from one service to another to get their services with high quality in less time. This trend is considerably frequent in videos, as the consumer gets irritated when the video game's ping is high. Over time,

multimedia content is revolutionizing. Improvements in video quality and resolutions are observed to a greater degree, which increases bandwidth usage. Still, better video quality requires more space to store content. Moreover, much higher bandwidth should be required to produce these videos. Caching and commercially available distribution technologies are costly and offer service providers less control over content management. IPTV services have the same form of effect as discussed above. In addition to the Video on Demand (VoD) service, IPTV services provide the live streaming capability. The content management and delivery of such services should be structured in such a way as to enhance the quality of customer experience and provide cost-effectiveness to service providers. Several global vendors such as Netflix, YouTube, and Hulu use HAS to deliver the up-to-mark QoE to end-users while keeping in view network quality or QoS [5,6]. The QoS includes delay, packet size, packet loss ratio, jitter, response time, and throughput [7]. In HAS, the original video content is converted into several bitrates; each bitrate is further converted into several segmenter segments, which may be called chunks. Each video segment or chunk has a fixed duration video in seconds [8,9]. The chunks' path or address is stored in a high-level language file called a media presentation description or a manifest file. Once the user has a manifest file, it is elementary to stream the video content from the original server or the content delivery networks (CDNs) [10]. The concept of a smart city is mainly based on information and communication technologies (ICT). The smart city is one of the main pillars for developing, deploying, and promoting sustainable development practices to meet the growing challenges of urbanization. Smart cities should include smart networks as well as efficient video streaming. Our contribution is embedded in this spirit where special attention is given to Content Delivery Networks. Content delivery networks enhance the quality of service and response time of the web static and dynamic pages. The HAS's main feature is to allow a multimedia player to adapt the video stream or demand the optimal video stream while keeping in view the network's condition and switch between different representations of the same video content seamlessly and effectively. Several giant company holders implemented their proprietary protocols for adaptive multimedia streaming, for example, Apple [11], Microsoft (MSS) [12], and MPEG-DASH [13,14]. Most of them have similar features while using different formats and mechanisms for HAS. Modern smartphones and mobile devices have much greater computing powers and capabilities and can perform multiple tasks effectively and quickly. Moreover, these devices can use different ports and interfaces to connect to the global Internet [15,16]. Mobility is the most fascinating and demanding feature of these devices. Everyone wants to connect with the Internet from everywhere, whether from their homes, offices, and picnic places, which allows users to stream multimedia applications in all these places in an effective way. Furthermore, multimedia services are generating a very high percentage of downstream traffic. The bandwidth usage of Netflix, a video content provider, has jumped to 40 percent of the Internet traffic and consumes more bandwidth than YouTube and Amazon. As mentioned by Cisco [17,18], global mobile traffic will increase by 8 percent of the current mobile traffic between 2015 and 2020. After considering all the factors, this research suggests a robust, effective, and flexible architecture for video streaming by integrating market available components for ensuring the effectiveness in terms of cost and streaming of videos by incorporating load balancers, web, and caching servers. A trans-coding server is also included in the proposed architecture, which can efficiently encode and decode raw videos into multiple short-duration segments, having bitrates for low, medium, and high-resolution quality. This trans-coding server is responsible for increasing QoE for the users. They can stream video segments with low bitrates if they do not have enough capacity to stream high-resolution video segments. The protocol used in trans-coding videos is HLS (HTTP Live Streaming), which Apple developed and is widely supported in multimedia servers. Moreover, it is based on standard HTTP transactions and is easily configured with every multimedia service. HLS converts raw video files into multiple short time chunks with an extension of the .ts file. The .ts file is directed by a playlist with a path for all these chunks, and the playlist has an extension of

.m3u8, which is further directed by the master playlist. The master playlist serves as a bank for each existing video quality. Both the media playlist and the master playlist are UTF-8 formats and have URI and a descriptive tag concerning every index file in the case of a media playlist and a playlist file in the case of a master playlist. The main contributions of this paper are as follows:

The elements of the originality of our paper may be summarized as:

- We proposed a new and efficient architecture for the adaptive bitrate video distribution and its streaming over HLS protocol.
- We assessed that the proposed architecture achieved high throughput and less response time by using a Content delivery network for Adaptive bitrate videos.
- We studied the effect of segment size variation over throughput, the response time of videos, the number of stalling events, and the frequency of bitrate.
- We analyzed QoE through subjective and qualitative measurements.
- We also analyzed the resource usage for all segment sizes of the video and suggested the most appropriate segment size that results in the effective delivery of videos in terms of resource usage.
- The segment size from 0–20 s is analyzed after 2 s, and its effect is discussed in the results and simulation section.
- The effect of RAM, throughput, response time, the impact of transactions, effect of segment size, and real deployment over the same architecture incorporating the content delivery network while deploying in the real application are analyzed in this research; such a study had not been conducted before in such an extensive way.

The remainder of this paper is as follows. Section 2 reviews related work and existing works are discussed and compared. Section 3 presents the proposed system of the architecture with all its components. Section 4 details the system implementation with all the criteria concerned. Section 5 presents the simulations and results of throughput and response time. Finally, Section 6 concludes the paper and gives future research perspectives.

2. Literature Review

Over the last decades, the Internet has spread like a nuclear reaction worldwide, leading to increased network traffic and data, causing network congestion and low response time of services over the Internet. The increased number of requests over the same servers can create a hotspot on web servers [19]. Web servers are overwhelmed by many users' simultaneous demands, leaving the web servers in a temporarily unreachable state. These issues may easily be mitigated by using Content Delivery Networks (CDNs). This type of network brings data near the users and caches data in a ready to use form; therefore, throughput and uptime of the website being used more concerning CDN technology [20,21]. More often, users even configured their CDNs on technologies like WordPress for their websites. Most of the CDN providers generate much revenue because of various undeniable features of CDN services that increase the quality of experience for the users. These functionality and revenue trends attract researchers to work in this field and generate feasible and easily manageable content providers' solutions. Effective delivery of resources by using network elements can be called a Content Delivery Network (CDN) [22]. It may be in many forms, as it may be centralized or decentralized, in an administrative domain, or more than one. Features that are important in CDNs are a redirection of the user's request, distribution among various servers, content retrieval, and management of the content and servers [23]. The performance of the CDNs can be enhanced by making mirror servers of the original one and redirecting users to the most suitable one based on some criterion, which reduces the sudden load on the main servers. CDN can significantly reduce the response time and enhance the service quality to which it is used [24]. Digital data is used in CDNs, in which static and dynamic data can be classified differently and dealt with in this way to boost service performance [25]. CDN architecture is focused on three primary roles: Resource provider, CDN provider, and end-users. Most service provider companies use mainly third-party CDNs because they are unable to handle their

CDNs. Caching in CDN plays an essential role as the content is replicated on various servers located in different locations, providing this cached content to the nearest users in the region called surrogates or edge servers.

This section includes both the related work of content delivery networks and adaptive bitrate video streaming architectures. Content distributors are also efficiently developing techniques and tools to deliver their content and generating revenue from their infrastructure. White papers addressed the need and main components that should be included in the content delivery system to automate the distribution process and improve service quality by enhancing the quality of experience (QoE). The open-connect is the architecture that Netflix has deployed for video distribution over VOD services, not only used by its services, but they also market their product to generate revenue. Another contribution in [26] describes the architecture and provides an overview of the content delivery networks for mobile streaming. This paper essentially tells a brief description of the MSM-CDN system and presents the testbed design based on those architectural concepts. This work provides a new medium for media content delivery. Pedro Luis deployed a virtual architecture in the University of Quindio and presented two architectures, one for video on demand (VOD) and the other for live streaming of video over HTTP for the broadcast of educational material. It uses the DASH protocol to test the capacity of its server by using a stress tool by Apache and only provides the results of the response time of the server over different bitrate and number of users. Dynamic Adaptive Streaming over HTTP (DASH), additionally regarded as MPEG-DASH, is an adaptive bitrate streaming method that permits excessive-quality streaming of media content material over the Internet introduced from conventional HTTP net servers. Similar to Apple's HTTP Live Streaming (HLS) solution, MPEG-DASH works with the aid of use breaking the content material into a series of small segments, which might be served over HTTP. Each section incorporates a brief c language of playback time of content material; this is doubtlessly many hours in duration, including a film or the stay broadcast of a sports activities event. The content material is made to be had at quite a few exceptional bit prices, for example, opportunity segments encoded at exceptional bit prices protecting brief aligned periods of playback time. The content material is performed lower back with the aid of a MPEG-DASH consumer. The consumer uses a chunk fee adaptation (ABR) algorithm to routinely choose the section with the very best bit fee viable that may be downloaded in time for playback without inflicting stalls re-buffering occasions inside the playback. Moreover, it did not include the support of any cache server or load balancer and did not provide the effect of segment sizes. The author created a virtualized environment and did not use the simulator that actually can download the chunks created by DASH and provide optimal response time [27]. Another architecture proposed by Miran Taha, which is also a virtualized network environment and developed over the VNX program over the single machine including cache serves to support, uses DASH protocol to transcode the video into its segments and evaluate QoE metrics in the architecture by doing subjective analysis of many stalling events. The number of times the video changes its resolution during playback time is determined by conducting a video session. Moreover, accumulative time and CPU usage are also evaluated. The author suggested that the videos' preferred segment size is 6–8 s after evaluating the mean opinion score (MOS) [28]. Zbrovskiy Anatoliy et al. present simulations with simple architecture over mininet and use the bitcodin service to transcode videos while evaluating the research's switching representation parameter by performing experiments on the mininet environment [29]. Yomna et al. present research on streaming over WiMAX networks by using an op-net simulator upon one physical node, which can easily be moved and evaluate the effect of varying segment sizes of the video on throughput and CPU computations; they found that small segment sizes are useful in quality video streaming. This architecture was quite simple, without any cache and load balancing servers [30]. Yae et al. deploy a real environment testbed for investigating initial delay, stall, throughput, and CPU consumption in DASH video streaming. The author uses five nodes for the deployment in simple network topology and analyzes that small segments give better performance than

bigger segment sizes [31]. Jaehyun Hwang et al. investigate the different segment length for adaptive bitrate video streaming and conclude that the smaller the segments the faster their reaction when network bandwidth oscillates, or problems like network fluctuation network congestion occur [32]. Another study [33] benefits from the CDNs and proposes the streaming service along with incorporating content delivery network distribution and its management, presents research in the comparison of P2P video streaming and with CDN streaming, and assesses that better performance is archived with CDN in contrast to P2P delivery. This work was done in simulations; its real scenario is still missing yet. Some studies were made on the streaming of adaptive bitrate video streaming in an educational context. The author proposes a solution for streaming and video playback using periscope to globalize the education of pathology [34]. In [35], streaming is done for the knowledge of online courses and teaching, in which good quality of experience matters for which a solution is proposed for the delivery of instructional notes over HTTP. A concept of Massive open online courses is presented in [36], which uses the video streaming application as the leading resource in the delivery of instructional notes. Another research [37] shows and validates over emulated scenarios concerning the control of the video selection, while the switching off of the video is controlled by the controller. This video's adaption occurs automatically by using controllers at the user's end, application, and actuators at the server-side. Another control is the switch control loop, which performs the role of controlling the switching between streams. The study [38] incorporates the architecture WebRTC technology that transforms communication as it enables the real-time interaction of the web clients. It includes incorporating API's power in building video and audio codec, web interfaces, and plugins. These were offered by some big companies like Telestax [39], and Bistro provides intelligent platforms for video conferences.

Considering the above-discussed issues regarding Quality of Experience (QoE), many testbeds have been proposed or deployed to estimate and rate multimedia application performance and provide a platform for fulfilling testing requirements. Many types of research suggested real as well as virtualized network testbeds for carrying out experiments. The execution of virtualized and real scenarios was carried out in [40–42]. They compare different network scenarios and observe in the test results that packet loss and delay of real-time systems consume a higher rate than simulated scenarios. Performance evaluation and guarantee HTTP adaptive live streaming over heterogeneous network conditions such as wired, wireless, and on different devices such as mobile in real-time is challenging. This research's most important aspect is to carry out an accurate, cost-effective, scalable, and implementable testbed design and, moreover, to develop its content delivery network. As the content delivery network available globally is very costly, it can be a possible solution for websites containing text content and small video content. Still, the multimedia content provider cannot become affordable because most of the CDNs are charged on data stored by the content provider and bandwidth used. Therefore, the quality of a video that enhances the bandwidth usage will also increase, and this will automatically increase the rates or charges. Furthermore, in a third-party solution, there is less control in video streaming as they provide services worldwide for various applications. Therefore, in this research, we gathered information from real users, our application system and also conduct result by using a simulator to test the testbed while increasing the number of users, fetching all the bitrates of the video to see the actual throughput and response time difference in between them while keeping a fixed number of users. Moreover, the throughput and response time are also analyzed and evaluated by changing the videos' segment sizes using HLS trans-coding. This research testbed is used to evaluate three important key performance parameters in HAS while changing the segment size that affects overall QoE: Initial delay, video stalls, and switching quality representation in a video session. The correlation between QoE measured from the proposed model, and QoE from the subjective test is evaluated.

3. Proposed System Architecture and Performance Parameters

This section details the system and necessary components for real environment deployment. Conventionally, multimedia content transportation over the Internet can be considered using vast networks, including content delivery networks (CDNs), where users connect to ISP using wired or wireless media such as Wi-Fi. The video is delivered to users using different network access environments through cache servers nearest to the users. When the content is not available in cache servers or replica servers, then the cache server fetches the user's request from the central server, serves the user with the desired request, and saves the copy of the future content requests of the users. Other replica servers can fulfill the request and main servers [13]. The principle system design contains both virtualized and real nodes. The system components which are the mainstreaming server, cache servers, and load balancers are deployed on a real physical machine. Every component is a separate virtual machine (VM) deployed on a single physical server. These machines run over the Ubuntu 16.04 operating system. The overall performance is shared between the virtual machines, and all machines run on a single server. Therefore results will improve more if you use separate machines rather than virtual machines. The impact of performance on throughput and delay are also discussed in the results section as well as simulations based on Ram variations in machines. The trans-coding server is deployed on a separate physical machine equipped with high processing power for the videos' fast trans-coding. Moreover, DNS load balancing is achieved through an application named Dyna to provide maximum uptime of the service [43]. The trans-coding server is configured with Trans-coding Service for Live and Video on Demand content. This service takes live channels or recorded media content in any format as input. It converts it to HLS format for streaming to clients and has to convert video into adaptive bitrate video. The origin server is referred to as the mainstreaming server where the content providers store trans-coding media. Cache servers are the replica servers that serve the user with media content. If the cache server does not have the desired content, it requests the content through the central server, serves the user, and stores a copy of its content. Load balancer/DNS load balancer is the front-end application configured to balance the load between all the cache servers and DNS load balancing servers to save the single point of failure and provide optimal service uptime. As shown in Figure 1, the proposed research design can connect the external networks: The Internet or intranet. Clients can be connected through any heterogeneous device. For the objective as well as subjective analysis, the proposed system uses both ways to request the video content for its performance analysis. First through simulated nodes or users by using the J.meter simulator that can request the adaptive bitrate videos and generate the results of throughput, initial delay, and latency by increasing the number of threads simultaneously [44]. The second way is to generate a request by real users through their mobile phones and their laptops. An app is developed for mobile users for the request of videos, in which EXO player is deployed and configured, through which adaptive bitrate video content is delivered to users without any hindrance; this app is configured to provide all the real-time statistics of the current video being viewed and store the information in a text file. After reading these statistics and watching the current bandwidth available for the user, the EXO player decides the video's quality that should be delivered to the user. A script is written in the open-source file EXO player to retrieve all the real-time video streaming statistics. When the bandwidth and bitrate of the user are less, the optimal threshold concerning the EXO player switches the quality of the video in a seamless manner, which reduces the number of stalls and increases the quality of experience of the user.

Figure 1. System Design of proposed adaptive Bit-rate video architecture.

3.1. Quality of Experience Metrics

This section explains the QoE metrics used in our research. We aim to calculate and analyze these metrics' effects over the QoE and get information about how their variations affect overall QoE.

3.1.1. Initial Delay

This metric defines the period between starting time of loading a video and start time of playing it [45]. The user end application forms a buffer length of T duration in seconds. This T depends upon the user's video segment size and the time when the buffer is available for the maximum size related to QoS factors such as availability of high bandwidth and variations in the bandwidth user end, and packet loss or packet drop rate. B_{ssi} is denoted as a buffer size of segment length concerning B_{ssi} , $B_{ssi} + 1$, where $I = 1$ to M , and M is the maximum degree of segment length, as shown in Figure 2. The user end application, which in our case is EXO-player or HLS player, will start to receive data at time T_0 and accumulate them in the queue maintained by the application before its first frame delivered at T_n , which would have some delay or more considerable response time of the video.

Figure 2. Timeline during video playback.

3.1.2. Quality Switch Frequency

The number of switches describes the number of times the quality changes from one bitrate to another in the entire video session; it depends upon two factors: The number of bits that flowed through the channel in one second and the instability of the throughput at the user end; when both these parameters are not optimized, the number of oscillations or the number of switches increases, which reduces the QoE of the user as the higher the number of switches in the entire session the more irritated the user will get and this may lead to switching the channel and service you are providing because when the switch

occurs a stall will also occur for a much shorter time and also jitters and glitches are seen on the screen of the device user who is watching the video; hence with the increase in the number of switches the QoE will be hindered; therefore, we need to minimize the number of switches. The following formula can describe the change of the number of switches:

$$\sum_{p=1}^n g(cp) = \begin{cases} 1 & \text{if } g(cp) \end{cases} \quad (1)$$

$$\sum_{p=1}^n g(cp) = \begin{cases} 1 & \text{if } g(cp-1) \neq g(cp) \end{cases} \quad (2)$$

$$\sum_{p=1}^n g(cp) = \begin{cases} 0 & \text{else} \end{cases} \quad (3)$$

where p can be from 1 to n , n is maximum time bitrate change, and Cp stands for the chunk current bitrate; there the value of the above function can be represented as 1 if $g(cp-1) \neq g(cp)$, which means if the current bitrate and the previous bitrate are not equal, then the value of the function will be 1, and the value of the quality switch will be incremented.

3.1.3. Stalling Events

Several stalling events may also be referred to several pauses, stops, or several times the video tends to buffer in the whole video session; this is a significant parameter in QoE's prediction. Many stalling events occur during video playback; the mean score for rating reduces because users get irritated. The frequency of pauses increases, and the average time of the video in the complete video session increases.

$$\text{average buffer time} = \beta_0 + \beta_1 + \beta_2 + \dots + \beta_n \quad (4)$$

where β is the number of times the video stalls; therefore, the *averagebuffer time* is the sum of all the pauses in the video session.

3.1.4. Mean Opinion Score

The mean opinion score is also calculated to see the effect of the above discussed parameter that results in the variation of QoE of the end-users. Moreover, to check and analyse the behavior of the metrics with subjective analysis and objective analysis, subjective analysis is the way to rate the quality of experience from people who know about the video quality and have an interest in generating some meaningful reviews; in this research case, we have used as defined criteria, for all around the world, the rating of QoE, which can result in telling whether the user is satisfied or annoyed. The criteria of rating can range from 1 to 5, where a result from 3 to 5 indicates the user is satisfied and from 1 to 2 means that the user is annoyed.

4. Implementation Methodology for Adaptive Bit-Rate Video Architecture

This article proposed a well-defined architecture for an IPTV service, which can perform the effective distribution of HTTP video content. QoE can be easily estimated by evaluating and finding valuable metrics. The architecture depicted in the figure below consists of a trans-coding server, origin server, caching servers, load balancer, and DNS load balancer, ensuring seamless and efficient video streaming to the end clients. The trans-coding server further comprises two servers, the encoding server and the segmentation server. The entire implementation scenario is presented in Figure 3.

Figure 3. Adaptive bitrate Video Streaming Architecture.

The raw video is given as an input to the trans-coder. The raw or pre-encoded video content is de-multiplexed into separate audio or video files then encoded using an H.264 encoder and AAC audio encoder into several adaptive bitrate video files. In this research, only three representations are evaluated, i.e., high bitrate, medium bitrate, and low bitrate, and then given to a segmenter by using the HLS multiplexer, where the segmenter converts each representation into several .ts video chunks or segments. Afterward, the results are stored in the origin server or the video streaming server of the content provider. The available video content in the adaptive bitrate pattern is then pushed into several web servers (in our case, only two web-servers) located in the appropriate places. The traffic congestion can be reduced by separating them into different regions. Therefore, whenever the user requests content, the request is handled by a DNS load balancer, which is used to end the effect of a single point of failure. Afterward, the request is dealt over an HTTP load balancer using a round-robin algorithm; this load balancer chooses the caching server. For testing purposes, this round-robin algorithm is used; it will select the region-wise caching server using the IP-hash method of load balancing. Subsequently, the cache server checks whether it has the desired content or not. If not, it will pull the desired content, save a copy of the content in itself, and immediately copy the copy to the user. Next time any other user from the same region requests the same content, it will be delivered to the user from the caching server in no time. It saves the copy of the content in a ready to deliver format—the strategy used by the user called “the pull strategy”. Moreover, in this research, we have tried to predefine used components that are already available in the market and integrate them into suitable and sustainable video streaming architecture. The proposed solution is easily manageable and configurable according to the content provider’s needs. Moreover, a monitoring system is added into this architecture for easy configuration of the cache servers from the web manager. The number of requests handled by the server can be easily monitored. Finally, we used an EXO player at the user end to demand the appropriate video resolution according to the network condition at the user end.

The flow chart shown in Figure 4 describes the complete flow of the proposed adaptive bitrate video architecture. It describes the system’s flow that includes all the main servers, cache servers, network connection, and the load balancer and cache servers’ request. First, all the servers are turned on. The central server has the encoded video chunks in its master manifest file and uploads them onto the HTML video portal. The HTML video portal user can navigate through and can request any of the video files. When the user requests some media, the initial request is handled by load balancers; after the initial request, the load balancer decides which of the cache servers will provide the desired multimedia content. The cache server is responsible for providing the desired content in the most efficient way. These cache servers check whether the desired content is available in its cache or not. If it is not available, it fetches the main server’s content by establishing the HTTP connection and providing the desired content to the end-user and saves the copy in its cache in complete form for the next request.

Figure 4. Flow Chart of the proposed adaptive bit-rate video architecture.

An encoding server flow diagram is also provided in Figure 5 that shows how a raw video is de-multiplexed into audio and video and then encoded by an H.264 codec and AAC audio codec, multiplexed with an HLS encoder; it generates multiple *.ts* video segments and chunks. In this flow diagram, stream readers read the video stream from any raw video source like a video camera. A de-multiplexer separates both audio and video from the raw file by using audio and video decoders. After this, it provides both streams to the raw packet dispatcher. The raw packet dispatcher raw video, and raw audio are encoded through the H.264 video encoder and AAC audio encoder. Both the streams are then multiplexed and combined through an HLS multiplexer. The master indexed file is generated by the master index file generated at the HLS multiplexer's output.

Figure 5. Encoding server flow diagram.

5. Implementation Methodology for IPTV Service Case Study

The proposed architecture is also assembled for the case study of the Mobile IPTV service for 4G users and validated for live streaming and VOD. This case study is developed to mark the architecture for IPTV services for telecom and Internet service organizations. Due to the use of the IMS, it can be connected to any running network of any organization. The IMS is referred to as the “IP multimedia subsystem”; in this system, a client registers itself through the IMS system to become the service client. Then its billing and charges automatically charge through the IMS-based billing and charging system. This service runs over 4G and LTE networks and becomes part of the 5G architecture to enhance its future capability. The architecture, including the architecture proposed in this research, uses standard ports and tools that can be easily adaptable and flexible. The complete architecture diagram of the IPTV service is shown in Figure 6.

Figure 6. Proposed use case architecture of IPTV service.

5.1. IPTV Performance Metrics

This section contains the real-time measurements from the IPTV service; these measurements constitute QoE metrics, including initial delay, stalling frequency, and quality switching frequency. The measurements are listed in Table 1. It has been observed that, with an increase of segment size, the initial delay also increases, but the stalling events and quality frequency reduce.

Table 1. Real-Time Measurement from IPTV Service.

Segment Size	Initial Delay	Number of Stalling Events	Number of Quality Switches
2 (s)	1 (ms)	6 times	6 times
4 (s)	1 (ms)	2 times	4 times
6 (s)	2 (ms)	4 times	4 times
8 (s)	2 (ms)	4 times	4 times
10 (s)	4 (ms)	3 times	2 times
12 (s)	4 (ms)	2 times	1 times
15 (s)	5 (ms)	1 times	1 times
20 (s)	5 (ms)	1 times	1 times

5.2. Subjective Analysis

Another test is run over the architecture; this is for subjective analysis, which aims to evaluate and estimate the quality of experience to an accurate degree. The subjective analysis is done by selecting 15 users, who had knowledge about the video perception and its quality parameters. This test was aimed to estimate the quality of experience (QoE) to an accurate degree with real-time users. The test was conducted for every segment size on a scale of 1–5. A score of 1–2 indicates that the user is dissatisfied, and a score of 3–5 means the user is satisfied, where 1 is the poorest and five is excellent. The results achieved during this experiment are given in Table 2.

Table 2. Subjective Analysis of Real Users.

Segment Size	Initial Delay	Number of Times Quality Switches	Number of Stalling Events	QoE
2 (s)	1 (ms)	3 times	5 times	3
4 (s)	2 (ms)	4 times	4 times	2
6 (s)	2 (ms)	4 times	3 times	3
8 (s)	1 (ms)	5 times	4 times	3
10 (s)	3 (ms)	4 times	6 times	3
12 (s)	3 (ms)	2 times	2 times	4
15 (s)	4 (ms)	6 times	3 times	2
20 (s)	5 (ms)	4 times	3 times	2

After analysis of this experiment, it was seen that the results of the initial delay are much worse in the case of large segment sizes like in the range of 12–20 s. Moreover, other important factors were evaluated in this experiment, including a smooth playing region in segment sizes, in which a region between 6 and 12 s was found to be much smoother than other segments that have comparatively less initial delays. The bar chart of the people's reviews is given in Figure 7. The bar chart has several viewers on its vertical axis and its segment size on its x-axis, which shows that the size of 12 s has the most smooth playing region, and the size of 6 s is the second-best option. The smooth-playing region is good at 12 s because of less HTTP connection established with the main server at the cache server to fetch the video chunks. Moreover, power consumption is much less. Indeed, it is a real-time analysis with real users, which may differ in different conditions.

Figure 7. Region of smooth streaming.

Another important finding brought to attention that the mean opinion score given by the reviewers or the subjective analysis and the influence factors, initial delay, segment sizes and stalling frequency (objective analysis), have an inverse relationship with each other, as the influence factor number increases the quality of experience at the user end decreases exponentially, the trajectory is also given in Figure 8.

Figure 8. Relation of QoE with Influence Metrics.

6. Simulations and Results

In this research, we used the Big Buck Bunny video [46] to evaluate QoE parameters and the effects on quality of experience by changing different parameters of videos. For the production of HLS content, as described earlier, a trans-coding server was used that further uses FFmpeg and Lib-X264 library. This HLS content contains 24 representations that are constituents of three levels of video quality: Low, medium, and high quality. These three levels of video quality are encoded in different bitrates, which are gradually increased from low rate; in this research case, a low-quality video has a resolution of 320×240 with a bitrate of 936,000 bps. A medium-quality video has a resolution of 640×480 with a bitrate of 128,000 bps. In the same way, a high-quality video with a resolution of 1280×720 with a bitrate of 180,800 bps, with separate audio bitrates for each quality representation gradually increased from low quality to high quality—64,000 vps for low quality and 192,000 bps for high quality with a sample rate of 44,100 or 44.1 kHz.

Furthermore, this research provides eight versions of the same video with different video segment lengths or chunks sizes starting from 2 s. These segment sizes are 2 s, 6 s, 8 s, 10 s, 12 s, 15 s, and 20 s. By generating these different segment sizes, the effect over throughput, initial, and QoE will be observed by changing essential parameters.

In the first test, we observe the initial delay by varying the number of users. In this case users simultaneously try to fetch the same content. To perform this test, we used J.meter which can retrieve HLS streams by increasing the number of threads or users simultaneously and has the capability to retrieve relevant information, e.g., initial delay, response time, and throughput. Therefore, J.meter was allowed to run over a separate machine in Linux distribution platform and users were varied from 1 to 150; the response time was observed, for each quality representation, from low to high. While considering this response, it was observed that there is no significant difference in up to almost 50 users while fetching high quality or low-quality video. Still, as users are increased from 50, one can easily observe that the three video qualities' gap increases significantly. Response time for each quality increases uniformly with the increase of users. Therefore, 50 users can be declared as the point of fluctuation or the fluctuating point, as observed in Figure 9. There is a huge difference between the low, medium, and high-resolution video files in accumulative time. Our proposed architecture's basic purpose is to reduce response time by using CDN. Figure 9, referred in this simulation result, achieved a much slower response time than the actual one without CDN.

Figure 9. Response time and accumulative time graph with number of users with CDN.

Similarly, in Figure 10, we had also generated the result of the initial delay with a variation of the number of users while not using a load balancer and caching servers. It can be analyzed that the maximum peak of accumulative time with CDN was around 200,000 ms, while not using CDN results in a maximum peak of 3,200,000 ms.

Figure 10. Response time and accumulative time graph with number of users without CDN.

Moreover, the throughput is also measured and evaluated against the number of users by using J.meter. We also compared the system's throughput without using a content delivery network and the user content delivery network. It was observed that, using CDN, the throughput increases, and the difference is more interestingly seen in the case of fewer users or requests, as presented in Figure 11.

Figure 11. Throughput with varying number of users.

Similarly, another observation is made that response time and accumulative time for three qualities changes with the variation of segment size; until 10–12 s it increases with the increase of segment size, and then it becomes much more uniform at the duration of 15–20 s. However, response time is the lowest at the segment size of 2 s. The results of the above discussion are given in Figure 12.

Figure 12. Response time with varying number concerning segment size.

Resource Usage Evaluation

Resource usage evaluation is also carried out upon the proposed architecture, in which we captured the CPU load for 300 s for all segment sizes that are of 2 s, 4 s, 6 s, 8 s, 10 s,

15 s, and 20 s. From this one can observe the behavior of the CPU consumption and Energy consumption as shown in Figure 13. Important things that can be seen in this evaluation are:

- Segments with short sizes do more CPU consumption than the segments of greater sizes.
- Likewise, short size segments appear to consume more energy than the segments with greater sizes in seconds.

Figure 13. Effect of segment size over CPU consumption.

This type of behavior appears because, while fetching short size segments, the clients have to make multiple HTTP connections to the central server and then store them into the buffer of the application for smooth playback of the desired video; the client will establish several connections based on chunks of videos; the fewer the video chunks the fewer will be the connections established by the client or from the client end as shown in Figure 14.

Figure 14. Effect of segment size over energy consumption.

7. Comparative Analysis

Table 3 shows that the benchmark comparison of the proposed architecture with other existing architecture is provided in detail. All the criteria are compared for the comprehensive comparison to enlighten our proposed new and efficient architecture for video delivery. This comparison contains the comparison of simulators used, the number of real nodes tested in every environment. This involves the concept of virtualization, load balancer availability, availability of the content delivery network, performance metrics used to analyze architecture, scalability, realism, virtual machine existence, mobility, scalability, preferred video segment size, and last but not least current deployment in real scenarios or applications. This article also provides a comparison so that it can be used as a comparative study of video streaming architectures.

Table 3. Benchmark comparison with the existing architecture.

Criteria	Educational Testbed [47]	Testbed for QoE Analysis [48]	Simulated Testbed on Mininet [49]	Real Testbed [31]	Testbed over Wi-Max [30]	Our Proposed Testbed
Used Simulator	Apache stress test tool	Not discussed	Mininet	N/A	Opnet	J.meter and siege stress tester tool
Number of Nodes Tested	01	01	01	05	01	06
Mobility	Not discussed	Easy	Easy	Low	Easy	High
Scalability	Not discussed	High	High	Low	Medium	High
Performance Metrics	Response time	CPU consumption, segment size effect and MOS of staling events	Switching representations	Segment bitrate, video bitrate and initial delay	CPU consumption and throughput	Response time, throughput, effect of segmentation, initial delay, effect of RAM
Realism	Simulation	Emulation	Simulation and emulation	100% real	Simulations	Real and simulations
Network Topology	Simulation	Emulation	Simulation	Real	Simulation	Real
Cache Service	No	Support	No	No	No	Highly efficient
Load Balancing	No	No	No	No	No	Yes
Preferred Segment Sizes	No	4–6 s	Not discussed	Small segments	Not discussed	6–12 s
Language	Not discussed	Extensive language	Python	Not discussed	Not discussed	JSON'S, html and C++.
Virtual Machine	No	Linux	Linux	No	Not discussed	Yes
Network Topology	Simple	Complex	Simple	Simple	Simple	Complex
Wireless Tested Nodes	No	No	No	No	Yes	Yes
VM Chains Distribution	No	Support	No	No	Not discussed	Yes
Current Deployment	No	No	No	No	No	In production of IPTV service

8. Conclusions

In this research, an adaptive bitrate video delivery network was tested to enhance the response time and throughput by varying different factors influencing it. Moreover, fundamental parameter fluctuations are also measured and analyzed by varying the same video's segment size with multiple representations. The factors, e.g., accumulative time, stalling frequency, and quality switching representation, are also compared with experience quality. An inverse relationship was observed between all the factors and the QoE from the subjective users. Furthermore, it was observed that load balancing and caching servers have a positive effect on response time, throughput, and the number of transactions through the Siege load testing tool by increasing the number of concurrent users from 1 to 200 users at a time. The whole testbed is deployed on real machines, and the real-time measurements were taken through J.meter, which evaluates the performance of the proposed architecture in terms of response time and throughput. This research presents that segment size is an essential factor contributing effectively to the QoE at the end-users. By analyzing

throughput and response time by the users, a segment length of 6–12 s is proposed for the video's smooth streaming. The energy and CPU consumption are also evaluated. Small segments result in more energy and CPU consumption in slightly longer chunks. Multiple connections were generated by the end client application or server to fetch the smaller segments multiple times. The statement can be correlated with Figures 13 and 14. The contribution and relevance towards sustainability can be realized by analyzing the system's source usage and energy consumption parameters. Furthermore, the design of the architecture is a novel step toward intelligent and sustainable multimedia networks. In the future, the architecture design could be expanded by including an intelligent system for network monitoring and trans-coding of videos in real-time; a centralized monitoring system may be included for intelligent route optimization of the users to the concerned caching server. Moreover, by taking all the real-time statistics, the server can decide the encoding of the videos, to which time a specific coding scheme will optimize the QoE at the user end. Besides, for the QoE analysis, a prediction model may be implemented by applying machine learning algorithms like neural networks, which enables predicting the required QoE and current user's QoE, and then compares the subjective and objective QoE that may be made.

Author Contributions: Data curation, M.H.B.W., F.J., A.Q.; Formal analysis, M.H.B.W., F.J.; Funding acquisition, O.C., H.H., B.B.; Investigation, H.J., M.I.; Methodology, M.H.B.W., A.Q., H.J.; Software, M.H.B.W., A.Q.; Supervision, A.Q.; Validation, M.H.B.W., M.I., O.C., H.H., B.B.; Visualization, M.H.B.W., A.Q., F.J.; Writing—original draft, M.H.B.W., F.J., A.Q., H.J.; Writing—review & editing, O.C., M.I., H.H., B.B. and H.J. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funded by Taif University Researchers supporting project number (TURSP-2020/55), Taif University, Taif, Saudi Arabia.

Acknowledgments: Omar Cheikhrouhou thanks Taif university for its support under the project Taif University Researchers supporting project number (TURSP-2020/55), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABR	Adaptive Bitrate
CDN	Content Delivery Network
HLS	HTTP Live Streaming
HAS	HTTP Adaptive Streaming
IP	Internet Protocol
ISP	Internet Service Providers
IMS	IP Multimedia Subsystem
P2P	Point to Point
QoE	Quality of Experience
QoS	Quality of Service
STB	Set-top Box
URI	Universal Resource Identifier
VM	Virtual Machine
WebRTC	Web Real Time Communication

References

1. Karn, N.K.; Zhang, H.; Jiang, F.; Yadav, R.; Laghari, A.A. Measuring bandwidth and buffer occupancy to improve the QoE of HTTP adaptive streaming. *Signal Image Video Process.* **2019**, *13*, 1367–1375. [[CrossRef](#)]
2. Robitza, W.; Göring, S.; Raake, A.; Lindegren, D.; Heikkilä, G.; Gustafsson, J.; List, P.; Feiten, B.; Wüstenhagen, U.; Garcia, M.N.; et al. HTTP adaptive streaming QoE estimation with ITU-T rec. P. 1203: Open databases and software. In *Proceedings of the 9th ACM Multimedia Systems Conference, Amsterdam, The Netherlands, 18–21 June 2018*; pp. 466–471.

3. Lee, C.H.; Chen, I.F. Collaborative P2P streaming based on MPEG-DASH for smart devices. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, 15–18 April 2018; pp. 1–6.
4. Lee, C.H.; Chen, Y.C. Multipath mobile multimedia streaming based on content delivery network and peer-to-peer network. In Proceedings of the 2018 1st International Cognitive Cities Conference (IC3), Okinawa, Japan, 7–9 August 2018; pp. 211–214.
5. Viola, R.; Martin, A.; Zorrilla, M.; Montalbán, J. MEC proxy for efficient cache and reliable multi-CDN video distribution. In Proceedings of the 2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Valencia, Spain, 6–8 June 2018; pp. 1–7.
6. Yang, W.; Hu, Y.; Ding, L.; Tian, Y. Viewer-Oriented CDN Scheduling on Crowdsourced Live Video Stream. In Proceedings of the 2019 IEEE 2nd International Conference on Electronics and Communication Engineering (ICECE), Xi'an, China, 9–11 December 2019; pp. 112–117.
7. Seufert, M.; Egger, S.; Slanina, M.; Zinner, T.; Hoßfeld, T.; Tran-Gia, P. A survey on quality of experience of HTTP adaptive streaming. *IEEE Commun. Surv. Tutor.* **2014**, *17*, 469–492. [\[CrossRef\]](#)
8. Hu, S.; Xu, M.; Zhang, H.; Xiao, C.; Gui, C. Affective content-aware adaptation scheme on QoE optimization of adaptive streaming over HTTP. *ACM Trans. Multimed. Comput. Commun. Appl. TOMM* **2019**, *15*, 1–18. [\[CrossRef\]](#)
9. Nguyen, D.V.; Le, T.T.; Lee, S.; Ryu, E.S. SHVC tile-based 360-degree video streaming for mobile VR: PC offloading over mmWave. *Sensors* **2018**, *18*, 3728. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Adhikari, V.K.; Guo, Y.; Hao, F.; Varvello, M.; Hilt, V.; Steiner, M.; Zhang, Z.L. Unreeling netflix: Understanding and improving multi-cdn movie delivery. In Proceedings of the 2012 Proceedings IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012; pp. 1620–1628.
11. Pantos, R.; May, W. *HTTP Live Streaming*; Apple Inc.: Cupertino, CA, USA, 2012.
12. Hahm, S.I.; Kang, P.; Bang, H.; Yeon, H.J. Dynamic media buffer control scheme for seamless streaming in wireless local area networks. In Proceedings of the 2016 IEEE Wireless Communications and Networking Conference, Doha, Qatar, 3–6 April 2016; pp. 1–6.
13. Zambelli, A. IIS smooth streaming technical overview. *Microsoft Corp.* **2009**, *3*, 40.
14. García, L.; Lloret, J.; Turro, C.; Taha, M. QoE assesment of MPEG-DASH in polimedia e-learning system. In Proceedings of the 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India, 21–24 September 2016; pp. 1117–1123.
15. Lloret, J.; Garcia, M.; Atenas, M.; Canovas, A. A QoE management system to improve the IPTV network. *Int. J. Commun. Syst.* **2011**, *24*, 118–138. [\[CrossRef\]](#)
16. Garcia, M.; Canovas, A.; Edo, M.; Lloret, J. A QoE management system for ubiquitous IPTV devices. In Proceedings of the 2009 Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Sliema, Malta, 11–16 October 2009; pp. 147–152.
17. Shi, W.; Dustdar, S. The promise of edge computing. *Computer* **2016**, *49*, 78–81. [\[CrossRef\]](#)
18. Waheed, M.H.B.; Rais, R.N.B.; Khan, H.; Bano, M.; Gilani, S.S.A. Indoor Mobile Localization Using Wireless Sensor Networks (WSNs). In Proceedings of the Workshops of the International Conference on Advanced Information Networking and Applications, Caserta, Italy, 15–17 April 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 341–351.
19. Adler, S. The Slashdot Effect: An Analysis of Three Internet Publications (1999). Available online: <http://ssadler.phy.bnl.gov/adler/SDE/SlashDotEffect.html> (accessed on 20 January 2021).
20. Bartolini, N.; Casalicchio, E.; Tucci, S. A walk through content delivery networks. In Proceedings of the International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Orlando, FL, USA, 12–15 October 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 1–25.
21. Bin Waheed, M.H.; Qayyum, A.; Gilani, S.S.A.; Khosa, I.U.; Bano, M. MMCDN: A Novel Architecture for Multimedia Content Delivery Networks. In Proceedings of the 2019 4th International Conference on Multimedia Systems and Signal Processing, Guangzhou, China, 10–12 May 2019; pp. 99–103.
22. Pathan, A.M.K.; Buyya, R. A taxonomy and survey of content delivery networks. In *Grid Computing and Distributed Systems Laboratory*; Technical Report; University of Melbourne: Melbourne, Australia, 2007; Volume 4, p. 70.
23. Abad-Segura, E.; González-Zamar, M.D.; Rosa, A.L.D.L.; Cevallos, M.B.M. Sustainability of educational technologies: An approach to augmented reality research. *Sustainability* **2020**, *12*, 4091. [\[CrossRef\]](#)
24. Aguilar, M. Aprendizaje y Tecnologías de Información y Comunicación: Hacia nuevos escenarios educativos. *Rev. Latinoam. Cienc. Soc. Niñez Juv.* **2012**, *10*.
25. Zhang, D.; Zhou, L.; Briggs, R.O.; Nunamaker, J.F., Jr. Instructional video in e-learning: Assessing the impact of interactive video on learning effectiveness. *Inf. Manag.* **2006**, *43*, 15–27. [\[CrossRef\]](#)
26. Wee, S.; Apostolopoulos, J.; Tan, W.T.; Roy, S. Research and design of a mobile streaming media content delivery network. In Proceedings of the 2003 International Conference on Multimedia and Expo, ICME'03, (Cat. No. 03TH8698), Baltimore, MD, USA, 6–9 July 2003; Volume 1, p. 1-5.
27. Wang, X.; Kwon, T.; Choi, Y.; Wang, H.; Liu, J. Cloud-assisted adaptive video streaming and social-aware video prefetching for mobile users. *IEEE Wirel. Commun.* **2013**, *20*, 72–79. [\[CrossRef\]](#)
28. Taha, M.; Lloret, J.; Ali, A.; Garcia, L. Adaptive video streaming testbed design for performance study and assessment of QoE. *Int. J. Commun. Syst.* **2018**, *31*, e3551. [\[CrossRef\]](#)
29. Zabrovskiy, A.; Kuzmin, E.; Petrov, E.; Fomichev, M. Emulation of dynamic adaptive streaming over HTTP with Mininet. In Proceedings of the 2016 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT), St. Petersburg, Russia, 18–22 April 2016; pp. 391–396.

30. Hassan, Y.M.; Helmy, A.; Rehan, M.M. Effect of varying segment size on DASH streaming quality for mobile user. In Proceedings of the 2014 International Conference on Engineering and Technology (ICET), Cairo, Egypt, 19–20 April 2014; pp. 1–4.
31. Liu, Y.; Dey, S.; Gillies, D.; Ulupinar, F.; Luby, M. User experience modeling for DASH video. In Proceedings of the 2013 20th International Packet Video Workshop, San Jose, CA, USA, 12–13 December 2013; pp. 1–8.
32. Hwang, J.; Lee, J.; Yoo, C. Eliminating bandwidth estimation from adaptive video streaming in wireless networks. *Signal Process. Image Commun.* **2016**, *47*, 242–251. [[CrossRef](#)]
33. Yin, H.; Liu, X.; Zhan, T.; Sekar, V.; Qiu, F.; Lin, C.; Zhang, H.; Li, B. Design and deployment of a hybrid CDN-P2P system for live video streaming: Experiences with LiveSky. In Proceedings of the 17th ACM International Conference on Multimedia, Beijing, China, 19–23 October 2009; pp. 25–34.
34. Fuller, M.Y.; Mukhopadhyay, S.; Gardner, J.M. Using the Periscope live video-streaming application for global pathology education: A brief introduction. *Arch. Pathol. Lab. Med.* **2016**, *140*, 1273–1280. [[CrossRef](#)]
35. Hartsell, T.; Yuen, S.C.Y. Video streaming in online learning. *AACE J.* **2006**, *14*, 31–43.
36. Sigama, K.; Kalema, B.M. Conceptualizing moocs implementation for higher education in developing countries. In Proceedings of the 2018 IEEE 6th International Conference on MOOCs, Innovation and Technology in Education (MITE), Hyderabad, India, 29–30 November 2018; pp. 14–18.
37. De Cicco, L.; Mascolo, S. An adaptive video streaming control system: Modeling, validation, and performance evaluation. *IEEE/ACM Trans. Netw.* **2013**, *22*, 526–539. [[CrossRef](#)]
38. Johnston, A.B.; Burnett, D.C. *WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web*; Digital Codex LLC.: Lilburn, GA, USA, 2012.
39. Stockhammer, T. Dynamic adaptive streaming over HTTP—Standards and design principles. In Proceedings of the Second Annual ACM Conference on Multimedia Systems, San Jose, CA, USA, 23–25 February 2011; pp. 133–144.
40. Taha, M. A novel CDN testbed for fast deploying HTTP adaptive video streaming. In Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications, Xi'an, China, 18–20 June 2016; pp. 65–71.
41. Garcia-Pineda, M.; Felici-Castell, S.; Segura-Garcia, J. Using Factor Analysis Techniques to Find Out Objective Video Quality Metrics for Live Video Streaming over Cloud Mobile Media Services. *Netw. Protoc. Algorithms* **2016**, *8*, 126–147. [[CrossRef](#)]
42. Jimenez, J.M.; Romero Martínez, J.O.; Rego Máñez, A.; Lloret, J. Analyzing the performance of software defined networks vs. real networks. *Int. J. Adv. Netw. Serv.* **2016**, *9*, 107–116.
43. Jiang, J.; Sekar, V.; Zhang, H. Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. In Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies, Nice, France, 10–13 December 2012; pp. 97–108.
44. Zabrovskiy, A.; Feldmann, C.; Timmerer, C. Multi-codec DASH dataset. In Proceedings of the 9th ACM Multimedia Systems Conference, Amsterdam, The Netherlands, 12–15 June 2018; pp. 438–443.
45. Xu, S.; Sen, S.; Mao, Z.M. CSI: inferring mobile ABR video adaptation behavior under HTTPS and QUIC. In Proceedings of the Fifteenth European Conference on Computer Systems, Heraklion, Greece, 27–30 April 2020; pp. 1–16.
46. Guzmán, P.; Arce, P.; Guerri, J.C. Automatic QoE evaluation for asymmetric encoding of 3D videos for DASH streaming service. *Ad Hoc Netw.* **2020**, *106*, 102184. [[CrossRef](#)]
47. Agudelo, P.L.; Campo, W.Y.; Ruíz, A.; Arciniegas, J.L.; Giraldo, W.J. Architectonic Proposal for the Video Streaming Service Deployment Within the Educational Context. In Proceedings of the 12th Colombian Conference, CCC 2017, Cali, Colombia, 19–22 September 2017.
48. Solera, M.; Toril, M.; Palomo, I.; Gómez, G.; Poncela, J. A testbed for evaluating video streaming services in LTE. *Wirel. Pers. Commun.* **2018**, *98*, 2753–2773. [[CrossRef](#)]
49. Sanabria, N.F.W.; Bustos, J.G.; Hernández, W.E.C. Adaptive video transmission over software defined networks. *Visión Electrónica* **2019**, *13*, 152–161. [[CrossRef](#)]