

# Managing Video Processing and Delivery using Big Packet Protocol with SDN Controllers

Stuart Clayman  
Dept. of Electronic Engineering,  
University College London,  
London, UK  
Email: s.clayman@ucl.ac.uk

Mustafa Tüker, Halil Arasan, Müge Sayıt  
International Computer Institute,  
Ege University,  
Izmir, Turkey  
Email: {tukermustafa, arasanhalil}@gmail.com, muge.sayit@ege.edu.tr

**Abstract**—This paper describes the use of the Big Packet Protocol (BPP) for carrying video from servers to clients, and how SDN controllers can effectively manage the flow-rate and QoE, based on the available bandwidth. BPP relies on meta-data being injected into packets in order to provide information for network nodes on how to process those packets. Given specific commands, the network node can drop parts of the payload, called chunks in BPP. When using BPP, the strategy is not to drop whole packets, but to reduce the packet size by eliminating specific chunks. The approach allows for reducing the load on the network, when there is a limited bandwidth, by having a flow of packets regularly arriving at the receiver, so there is continuous delivery and minimum guaranteed quality. To make video transmission over BPP effective, a video encoder and decoder that can do multiple encodings for the same region is selected – namely scalable video coding (SVC). The results show the successful implementation of a system using these combined techniques.

**Index Terms**—SDN, Future networks, IP, BPP, SVC

## I. INTRODUCTION

Big Packet Protocol (BPP) is a new network protocol designed to address the needs and requirements of high throughput applications and network architectures [1], as part of future networks in 2030. One of the main objectives of BPP is to address the network requirements of high precision services, especially applications such as high-bandwidth video, AR, or VR. BPP allows a network node, such as the forwarding elements, to define and implement an application specific behavior at the level of the flow and the individual packet, to achieve these objectives [2]. To support such functionality, each BPP packet is partitioned into a header and a number of data chunks. Some of these chunks can be dropped during transmission by the network node, based on the contents meta-data in the header injected into the packet, and the load of the network [3]. The BPP approach is to remove parts of a packet, namely the chunks, but not to drop the whole packet. This adaptation of packet size is undertaken by considering the network conditions and any application feedback.

The BPP process for eliminating chunks provides advantages to streaming applications, such as video, where the data can be placed into packets, and sent over the network to the receiver. Such a process of only removing chunks, not whole

packets, provides a flow of packets regularly arriving at the receiver, so there is continuous delivery. In general, video players require all the video to arrive, however many network video receivers are designed to be adaptive so that they will continue even if some video frames are lost in transmission. These missing frames manifest to the user as different kinds of quality reduction, depending on which kind of frame is lost.

In this work we show that an SDN controller can take advantage of the BPP packet structure and packet modification capabilities to provide an enhanced end-to-end video transmission. An enhanced SDN controller was devised to maintain the highest quality of video for the receiver, but also to be dynamically adaptable to the network conditions, especially bandwidth usage. This SDN controller can effectively manage the flow-rate and QoE, based on the available bandwidth. When using BPP and an SDN controller, it is possible to modify packets by removing specific chunks by considering both the current network conditions and the knowledge of how video players can maintain the highest perceived qualities [4].

To utilize BPP effectively for video, to allow the SDN controller to make suitable packet modifications, and to ensure that the receiver gets the highest quality stream, the selection of video encoder and decoder is important. The choice is one that has multiple encodings for each frame, providing multiple layers whereby the video stream is encoded such that the video file contains one base layer and a number of enhancement layers. While the base layer provides the lowest video quality, it does not require any other layer to be decodable; the enhancement layers are dependent to the layers below to be able to be decodable. Such layer dependencies are compatible with the packet modification of BPP, and it allows the SDN controller to delete some chunks during transmission, while being aware of the layered structure of the video. If the network bandwidth becomes limited, the SDN controller can delete some chunks of the enhancement layers, but always keep the base layer, allowing the client to play the video [4].

In this paper we use Scalable Video Coding (SVC) [5] to encode video files at various bitrates. A video file encoded with SVC includes all quality alternatives due to its layered structure, resulting in one file with all the layers. The server has a video file which is encoded with SVC, and includes one base layer (L0) and two enhancement layers (L1 and L2). For

the implementation of the system, the first step is to construct BPP packets on the server side, by reading data from the video stream, and placing the base layers and the enhancement layers, into the BPP packets, with some priority labelling for L0 chunks. The second step is to potentially modify the BPP packets during transmission, if the network conditions demand it. Finally the packets are delivered to the client.

The main contributions of this work include: (i) a working implementation of BPP to transmit SVC video streams structure for SVC video data; (ii) the use of an SDN controller, to process, modify, and transfer the BPP packets; and (iii) the measurement of BPP's effect on QoE. The paper is organized as follows: in the background section, video streaming aspects are presented. The use of BPP and its enhancements for video are discussed in section III. The experimental setup and performance results are provided in the section IV, and the last section, the contributions of this work are summarized.

## II. BACKGROUND AND RELATED WORK

Video transmission can be done using a packet by packet approach with UDP, which is unreliable, or using a data stream with TCP, which is reliable. When using UDP, the network presents as a pipe that displays loss at the receiver. The application at the receiver has to deal with the loss of packets from the network. This loss can be shown missing data as grey squares in the video, or the application can choose to make a request for the missing data, but this adds some extra time for display. UDP works well when *low latency* is required, such as interactive video. When using TCP, the network presents as a byte stream which has no loss, rather, there is delay or latency. In this case, the application at the receiver has to deal with this latency. This is usually done by having buffer containing a few seconds of received video. The application is isolated from the network by the TCP stack in the kernel, and has no control over any packet resending. TCP works well for video players, which have no interactivity.

The HTTP protocol, uses TCP as a transport, and is often used to transmit video streams by CDNs. HTTP Adaptive Streaming (HAS) has become a de-facto delivery standard. To address network loss and network characteristics over time, the Dynamic Adaptive Streaming over HTTP (DASH) [6] standard has been devised. DASH was developed by MPEG to enable the inter-operation of elements that participate in HAS systems. In DASH, the video is split into a number of segments of varying qualities, usually of 2 or 3 seconds in length. Each encoding quality will be in different files, and the higher the video quality, the larger the file will be, the higher the bitrate, and the more bandwidth will be used. A client requests these segments from a server, and can make dynamic adjustments based on the network conditions. At any time, the client can dynamically adapt its requests for qualities, based on its internal parameters and observation of delay. The received segments are queued up for the decoder, for display.

The advantages provided by SDN leads to enhanced video streaming systems, for improving QoE. In these systems, SDN is utilized for selecting the routes for the video flows [7],

reserving network resources [8], [9], giving network related information to the clients [10], or guiding them using its network related information [11]. There are also a number of proposals to use layered video streaming over SDN. The general approach is to transfer video layers over different paths, where the routes are selected by considering the characteristics of the layers [12]. In none of the previous studies has SDN been used for adjusting the quality during transmission for providing an acceptable quality at the receiver side.

Scalable video coding (SVC), or layered video coding, enables video files to contain more than one quality within the same stream [13]. The frames of the video are encoded in such a way that the layers, from the base layer upwards, provide increasing qualities, with each layer improving on the previous one. Using layered video has been demonstrated to be beneficial for video transmission [5]. The layered approach takes advantage of the similarities between each encoding of the same frame, and the similarities between the frames. However, this type of coding, with its dependencies between video layers, can affect clients when there is lossy transmission.

BPP was introduced in 2018 [1], and has been used for Time Sensitive Networking [14], and in Mobile Edge Networks [15]. BPP is suggested as a good mechanism for multimedia transmission, however none of the previous studies have addressed the the transmission video using BPP, or the effects of using BPP for video. In this paper, we use video encoded with SVC, and place that video into BPP packets for transmission. BPP has a potential for video streaming, and it could become one of the main protocols for video streaming applications in the future as it can deal effectively with the dependencies between layers and between frames. In the next section, we present more details of using BPP for video transmission.

## III. USING BPP FOR VIDEO

In this section we present a more detailed view of using BPP for video transmission, which is an alternative transport layer protocol. There is little behavioural data for BPP, as it has not been evaluated in many live scenarios. The existing MPEG-DASH standard uses HTTP Adaptive Streaming (HAS), which has HTTP running over a TCP transport. DASH represents an application layer mechanism, with client intelligence. HAS systems are the most successful and dominant video streaming systems in today's network thanks to the ABR (Adaptive Bit Rate) algorithms used, and the advantages of HTTP, such as web-caches and easy firewall traversal. DASH systems need these additional components and mechanisms, and cannot just rely on the built-in TCP behaviour only. TCP's congestion control algorithm, slow start mechanism, competing TCP flows, enforced delay & RTT for reliability, all cause negative impact on the QoE of HAS. Hence, it can be foreseen that if another transport layer protocol is used, which eliminates the negative effects of TCP in streaming applications, the performance of these applications can be enhanced, and some of those additional mechanisms may no longer be needed.

When BPP is used for the transmission of video, it has a large impact on the applications that send and receive video.

Just by using BPP, does not guarantee improved behaviour or performance. Using BPP in the most simple way, some BPP chunks can be dropped from a packet, which is obviously different to having the whole packet being dropped. If video blocks are placed in BPP chunks, and some of those chunks are dropped during transmission, then at the receiver there will be loss of video data. This loss will still have to be dealt with by the receiver, and this has a very similar impact to loss of a packet. Used in this way, BPP provides very little benefit.

If we utilise some concepts from MPEG-DASH and map different video segments into the chunks of BPP packets, then each segment could contain data for the different bitrates. Unfortunately, DASH segments are usually of length 2 seconds, and this is way too much data to fit into a packet. Even if we were to reduce the DASH segment length to milliseconds, so that several encodings at different bitrates could be mapped into a packet, this would not help us. The video decoder does not expect to receive different bitrate encodings for each video frame. It is designed to play consecutive multi-second sequences, with bitrate changes happening more slowly.

Considering the use of BPP combined with DASH, which uses HTTP over TCP, or indeed any TCP delivery mechanism, we observe a contradiction. BPP is explicitly designed to remove some chunks from the packets during transmission. Conversely, all TCP based schemes are designed to guarantee the delivery of data, but with some delay and retransmission.

Operationally, BPP packets with video data are processed by an SDN controller, during their transmission across the network. Fig. 1 shows the general outline of this process, starting from the packets at the server, through the SDN controllers, to the client. The first two network hops are high bandwidth, and so no chunks are dropped, but the last hop to the client is a lower bandwidth, and we see that a Layer 2 chunk has been removed. On the server side, we use a packet filling approach which enables chunk removal during the transmission of the packets. The chunks are put into the packets so that each packet carries chunks from each layer. As there will be some chunks that can be removed from the packet, if the SDN controller decides to shrink the packet, this will be possible most of the time. For this purpose, the server partitions each layer of each frame and puts these partitions as the chunks. The SDN controller can decide to modify the packets by using its information about the available bandwidth.

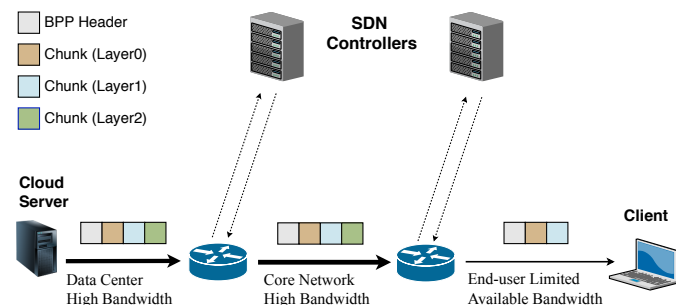


Fig. 1: Scenario with SDN Controllers updating BPP packets

## IV. PERFORMANCE EVALUATIONS

In this section we show the testbed setup and some comparative performance measurements and analysis.

### A. Evaluation Settings

For the performance evaluation, we used the Mininet platform, which is an open source emulator software that allows the creation and testing of Software Defined Networks, and we used ONOS as the SDN Controller in the study. ONOS is an open source Network Operating System developed by service providers, device manufacturers, internet operators, and ONF (Open Network Foundation). By utilizing the modules included in ONOS, current bandwidth measurements, transmission of packet flows, and traffic statistics are collected.

BPP packets are created to fit into standard 1500 byte Ethernet packets here, although BPP can carry Jumbo frames. For video, the "foreman" video sequence is used, which is 10 seconds long and includes 300 frames. The encoded video file has one base layer (L0) and two enhancement layers (L1 and L2). The base layer (L0) has 204 Kbps, L1 layer has 488 Kbps, L2 layer has 1094 Kbps. In a real implementation of BPP, it is possible to transmit video files with higher bitrates.

### B. Comparative Performance Measurements

The performance results are obtained by measuring the raw data in packets received by the clients. We also conducted tests by using UDP and TCP for transferring the layered video packets, and compared the obtained performance results of BPP, UDP, and TCP. In the first set of tests, the available bandwidth of the paths were fixed. In the second set of set, we changed the available bandwidth dynamically with cross traffic. While constructing UDP and TCP packets, the packets are filled with the maximum number of layers that the packet can carry. The layers are put into the packets sequentially, starting from the base layer. The structure and the payload of the BPP packets was described in previous sections. As well as the raw data that is received, we measured two important QoE parameters: PSNR (in dB) and outage duration (in msec).

The graphs given in Fig. 2 present the received layers at the client side, as a function of frame number, for three different fixed bandwidth settings: 0.5 Mbps, 0.8 Mbps and 1.5 Mbps. The layers received by the BPP clients are given in the first three graphs, and by the UDP clients, in the last three graphs. Due to its reliable mechanism, TCP clients always get all the layers, so we do not present the graphs related to TCP here. When we examine the results of BPP and UDP comparatively, for the bandwidth settings of 0.5 Mbps and 0.8 Mbps, we see that BPP always transmits the base layer packets, whereas UDP starts the streaming by transmitting the video with the highest quality for a limited time, and then drops the quality, so that even the base layer for many frames does not arrive at the client. Both protocols transfer the enhancement layer packets occasionally even if the bandwidth is limited, because there is some available bandwidth, and the size of the some enhancement layers are not so big. When the bandwidth equals 1.5 Mbps, both protocols transfer all layers of all frames, since

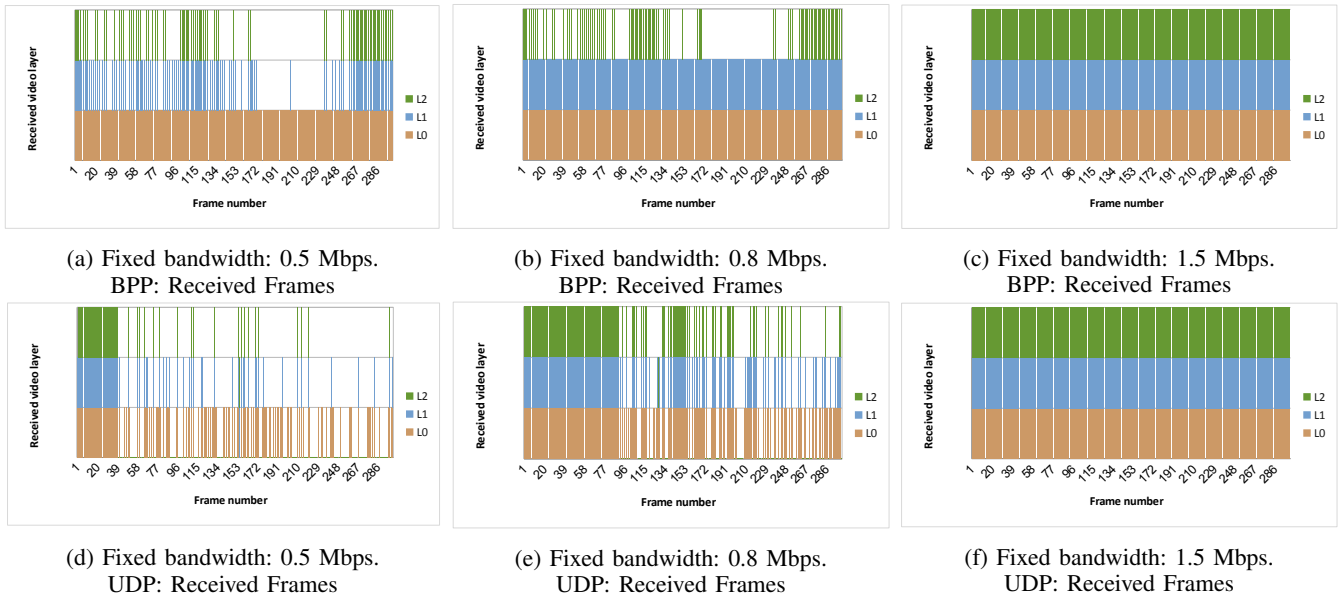


Fig. 2: Received layers for different bandwidth values

Bandwidth	BPP	UDP	TCP
0.5 Mbps	173	5470	14396
0.8 Mbps	205	2634	5697
1.5 Mbps	0	0	584
Ascending	0	916	2554
Descending	0	920	4691

(a) Duration of outages (in msec)  
*Smaller is better*

Bandwidth	BPP	UDP	TCP
0.5 Mbps	40	29	44
0.8 Mbps	40	36	44
1.5 Mbps	44	44	44
Ascending	39	18	44
Descending	40	37	44

(b) Average PSNR values (in dB)  
*Larger is better*

Bandwidth	BPP	UDP	TCP
0.5 Mbps	10	2.4	10
0.8 Mbps	10	4.2	10
1.5 Mbps	10	10	10
Ascending	10	1.7	10
Descending	10	6.7	10

(c) Video playout on the client side (in seconds)  
*10 secs is full video*

TABLE I: Comparing BPP vs UDP vs TCP QoE Metrics

the capacity of the network is good enough to transfer all the video with the highest quality. The results for ascending and descending bandwidths are not given here due to the space limits, however, table I has the relevant QoE metrics obtained.

The graphs presented in Fig. 2 represent the raw layers received by the client, but not processed to feed the video decoder. SVC video is encoded by using the similarities between consecutive frames, as well as the dependency between the layers of the same frames, thus there is a dependency among the same layers of different frames. Most frames make reference to other frames, and for proper decoding, all reference frames of a frame must be received by the clients. Therefore, these raw layers do not reflect playout capability.

In table Ia, the observed outage durations for different bandwidth settings are presented for the 3 protocols. *Outage duration* is one of the most important QoE metrics affecting the perceptual quality of video. If the outage values in the table are examined, it can be seen that the clients using BPP did not experience any significant outages, the highest outage is 0.2 seconds (205 msec). On the other hand, we observed up to 5.47 seconds of outages with UDP, and up to 14.4

seconds of outages with TCP when bandwidth is 0.5 Mbps. Considering that the duration of the video is only 10 seconds, having such long outage durations has a high negative impact on the perceptual quality. UDP clients have also experienced 2.6 seconds of outage with 0.8 Mbps, and 0.9 seconds with dynamic bandwidth scenarios. However, the highest outage values are observed in the TCP tests. There are two main reasons for this: all layers are sent to the clients, and all lost packets are retransmitted. Even when the bandwidth is 1.5 Mbps, an outage of 0.5 seconds is observed in the TCP test.

The PSNR of the original encoded video file is 44 dB, and the average received PSNR values are listed in table Ib. When the bandwidth is 1.5 Mbps, we see that using all protocols, the clients play the video with the highest possible quality, since no data loss occurs. For the other bandwidth settings, the clients with BPP play the video with a PSNR value always higher than 39 dB, which shows the quality of the video is always high. UDP can not provide consistent and good quality in all scenarios. In the ascending bandwidth scenario, we observe the average PSNR value is 18 dB, which shows the video quality is not acceptable. TCP clients always play

the video with the highest PSNR value, but with big outages.

Table Ic shows the duration of the video played at the client side. BPP and TCP clients could play the whole video sequence. When we examine the graphs for Fig. 2a and 2d, we see that the clients both continue to receive video layers until the end of the streaming session. However, for UDP, it shows that the client cannot play the video after a few seconds. The reason that it cannot be played, even if the client receives some frames, is that all the main frames providing references to other frames are lost after a certain point, due to bandwidth limitations. In essence, the network resources used for the transmission of most of the layers arriving at the clients are wasted. We conclude that managing the network by jointly using BPP and SDN provides higher effective bandwidth utilization, by adjusting the quality during transmission, and sending the higher layers only if there is enough capacity.

## V. CONCLUSIONS

In this study, we presented a video transmission system for layered video using the Big Packet Protocol (BPP), in a network under the management of an SDN controller. For this purpose, a BPP enhancement, to support layered video transmission, was proposed and presented in detail. An SDN controller which utilizes both real-time network condition information and video coding characteristics, was devised, and this controller manages the video transmission by considering the QoE of the video and the BPP chunks.

On the server side, BPP packets are constructed in such a way that they can be shrunk during transmission, with a small cost of quality degradation. During transmission, the SDN controller can modify the packets, and can remove some chunks by using its knowledge of the video characteristics and the current network conditions. While a HAS client adapts the video quality using its observation about the network and individual bandwidth estimations, our approach directly adapts the quality inside the network. This mechanism provides an *in-network quality adaptation*, which is obtained by combining the features of SDN and BPP. It is clear that quality adaptation in the network, by using the real bandwidth values, provides better performance than quality estimations on the client side, by using estimations on historical data.

By using both scalable video coding (SVC) and BPP, it was clear that they complement each other. With BPP, chunks can be eliminated from video packets, in a way that is far more adaptable and refined than the UDP method of dropping the whole packet. For the receiver, this approach radically improved the QoE, as demonstrated by our results. By using SDN, there is a level of flexibility and control, that no packet-by-packet network node can successfully provide. However, we see this is essential for in-network video management. SDN controllers provide the best results by using packet meta-data, network conditions, and flexible processing.

The results presented here demonstrate that using BPP for layered video transmission presents the client with a continuous stream of playable video and low levels of outage, even with limited bandwidths. With video streaming applications,

frames will be rejected by the decoder if they arrive at the client later than the playout time. In the experiments with BPP, all of the frames arrived in a timely manner, and only some enhancement layer chunks were dropped. Using UDP, we saw some frames were lost and others arrived late, by as much as 5 seconds. When TCP is used, there were no packets lost, but the latency increased to a number of seconds. It is clear that BPP provides benefits for the high precision timing of video, and that our selected techniques, using an SDN controller, worked well for transferring video.

In this work, we have demonstrated that the basic concepts of chunk elimination work, and we have built a foundation for using video over BPP, not yet a whole system. In future we can evaluate many more aspects like scalability and throughput. Our experiments highlighted that there is scope to further improve the network bandwidth utilization, when using BPP and SDN. As the layers of the SVC video have dependencies between each other, if earlier enhancement layers are removed, the following dependents in that layer need not be transmitted.

## REFERENCES

- [1] R. Li, A. Clemm, U. Chunduri, L. Dong, and K. Makhijani, "A new framework and protocol for future networking applications," in *NEAT 2018 - Proc. of ACM Workshop on Networking for Emerging Applications and Technologies*, 2018.
- [2] L. Dong and R. Li, "Big Packet Protocol: Advances the Internet with In-Network Services and Functions," *MMTC Communications - Frontiers*, vol. 14, no. 5, pp. 23–29, 2019.
- [3] R. Li *et al.*, "A Framework for Qualitative Communications Using Big Packet Protocol," in *NEAT 2019: Proc. of ACM Workshop on Networking for Emerging Applications and Technologies*, 2019, pp. 22–28.
- [4] S. Clayman, "The Inter-Dependence of Network Transport and Application Behaviour," in *ITU Network 2030*, Geneva, October 2019.
- [5] S. Yang, Z. Wei, and L. Ling, "Multimedia Communication and Scalable Video Coding," in *Fourth Intl Conf on Intelligent Computation Technology and Automation*, Shenzhen, China, 2011, pp. 616–619.
- [6] "MPEG-DASH: Dynamic Adaptive Streaming over HTTP." [Online]. Available: <https://mpeg.chiariglione.org/standards/mpeg-dash>
- [7] M. Sayit, C. Cetinkaya, H. U. Yildiz, and B. Tavli, "DASH-QoS: A Scalable Network Layer Service Differentiation Architecture for DASH over SDN," *Computer Networks*, vol. 154, pp. 12–25, 2019.
- [8] S. Ramakrishnan, X. Zhu, F. Chan, and K. Kambhatla, "SDN Based QoE Optimization for HTTP-Based Adaptive Video Streaming," *2015 IEEE Intl. Symposium on Multimedia (ISM)*, pp. 120–123, Dec 2015.
- [9] A. Bentalab *et al.*, "SDNHAS: An SDN-Enabled Architecture to Optimize QoE in HTTP Adaptive Streaming," *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2136–2151, 2017.
- [10] K. T. Bagci, K. E. Sahin, and A. M. Tekalp, "Compete or Collaborate: Architectures for Collaborative DASH Video Over Future Networks," *IEEE Trans. on Multimedia*, vol. 19, no. 10, pp. 2152–2165, 2017.
- [11] J. W. Kleinrouweler, S. Cabrero, and P. Cesar, "Delivering Stable High-quality Video: An SDN Architecture with DASH Assisting Network Elements," in *Proceedings of the 7th International Conference on Multimedia Systems*, ser. *MMSys '16*, 2016, pp. 4:1–4:10.
- [12] R. S. Kalan, C. Cetinkaya, and M. Sayit, "Design of a layer-based video streaming system over software-defined networks," in *8th International Conference on the Network of the Future (NOF)*, 2017, pp. 8–13.
- [13] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, October 2007.
- [14] K. Makhijani, R. Li, and H. E. Boukary, "Using Big Packet Protocol Framework to Support Low Latency based Large Scale Networks," in *ICNS 2019*, Athens, 2019.
- [15] L. Dong and R. Li, "Distributed Mechanism for Computation Offloading Task Routing in Mobile Edge Cloud Network," in *International Conference on Computing, Networking and Communications (ICNC)*, Honolulu, HI, USA, 2019, pp. 630–636.