The Future of Media Streaming Systems: Transferring Video over New IP

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

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

***Abstract*—Big Packet Protocol (BPP), which is part of New IP, was designed to transfer packets for future networking applications, and aims to overcome obstacles within current networks for high precision services. One of the most important advantages of New IP is that it allows changes to packets during transmission. The strategy of BPP is to reduce the packet size by eliminating specific chunks, cutting out segments from the transferred video, rather than dropping or retransmitting packets. This provides an effective mechanism to enhance the performance of video streaming applications, by obtaining continuous delivery and minimum guaranteed quality at the receiver. In order to make video transmission over BPP effective, we need to select a video codec that can do multiple encodings for the same region, such as scalable video coding (SVC). To support such functionality, we have augmented the BPP packet structure in order to transfer video data. This paper describes the use of BPP for carrying video from servers to clients, and defines the packet structure for this purpose, plus the extensions needed to support SVC encoded video. To evaluate the proposed approach, we use SDN to facilitate BPP operations, with results showing a successful implementation of a system using these combined techniques.**

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

I. Introduction

IP and current data plane technologies have been successfully deployed for many years. However, due to the limited flexibility in the data-plane, these traditional technologies may fail in meeting the requirements of high precision services, which is an important functionality of the emerging and next generation networking applications and technologies, such as 5G verticals and the FT-NET-2030 initiative. Recently, a new data packet framework called New Internet Protocol (New IP) was proposed to eliminate the limitations of traditional networks [1]. As a new data plane technology, New IP defines a new header and capabilities of network elements.

The best-effort service model aims to transfer packets to destinations with minimum delay and maximum possible throughput, but it does not guarantee any service level. One of the main objectives of New IP is to support advanced services and delivery guarantees, which are essential for high precision services. This is crucial for emerging and networking applications, especially those requiring stringient delay and high-bandwidth and for providing an application specific behaviour. Video streaming systems, which are continuously expanding and have increasing needs, in terms of throughput, latency, and percentage of traffic, are an example of such applications.

Big Packet Protocol (BPP) was originally developed to support such functionalities, in order to implement high precision services and application specific service agreements [2]. New IP is an enhancement and extension of BPP, where some functions such as 'action specification' or 'conditions' have been reevaluated and redefined as the 'Contract Clause' functionality of New IP. The meta-data carried in BPP packets allows network nodes to act based on the fields of the packet [3]. BPP provides a packet structure and functionality that fills the packets with data chunks, some of which can be removed regarding to the pre-defined conditions during transmission. With this approach, it is possible to shrink packets instead of dropping the whole packet, in case of congestion [4].

BPP functionalities can provide advantages for video streaming applications, if the video is encoded on the server side so that the clients can still achieve an acceptable quality and seamless playout, even if some parts of the video packets are removed. For this purpose, we use Scalable Video Coding (SVC), which encodes the video files into several layers called base layer and enhancement layers, where the base layer is essential for playout, and each enhancement layer increases the quality of the video. The characteristics of SVC coded video are well suited to the packet modification mechanism of BPP [5]. The server puts the layers into the BPP packets, and those layers can be removed during transmission by the network nodes, when there is not enough available bandwidth. The concept has been succesfully implemented and evaluated [6], and the technique is able transfer video packets with no loss, but with an acceptable quality reduction during transmission.

In this work we show how the BPP packets can be constructed to support such video delivery, and how video transmission performance can be enhanced by using BPP's packet modification capabilities. We define the necessary fields that hold the meta-data specific to the scalable video. The header fields include the carried chunk properties within the packet, and the conditions signaling the priorities of the chunks. We utilise a Software Defined Networking (SDN) controller to facilitate BPP operations, where network nodes send BPP packets to the controller for possible modification.

Dr S. Clayman is partially supported by Huawei Technologies Co., Ltd.

978-1-6654-4005-9/21/$31.00 ©2021 IEEE

The contributions of this paper are threefold: *(i)* we defined the necessary BPP fields to carry video traffic; *(ii)* we define the conditions and some BPP header parameters to enhance the performance; and *(iii)* we measure the effects of BPP usage on QoE. We first show the structure of BPP packets, plus the main functions that the SDN controller needs to support. We then show the extensions and enhacements that are needed for BPP, in order to transmit SVC video effectively.

The paper is organized as follows: in the related work section, studies related to New IP and BPP are presented as well as video streaming aspects. The use of BPP and its enhancements for video are discussed in section III. The details about the SDN functionality are presented in Section IV and the performance results are given in section V. Finally, in section VI, the conclusions of this work are summarized.

II. Background and Related Work

BPP was proposed as a part of New IP in 2018 [7]. There has been several use cases and studies proposed using BPP in the literature since it is introduced [2]. The positive impact of BPP on the performance was shown for Mobile Edge Networks and Time Sensitive Networking, in [8]. In [9], it was shown that when chunks in packets are dropped, the latency is minimized. The authors emhasized the importance of high precision services for emerging architectures, and discussed how BPP can provide the needs for such services in [10]. Although BPP and New IP concepts are discussed in these studies in the literature, none of them focus on multimedia transmission over BPP, nor presented any results related to that. In our previous work, we presented the initial experimental results of using BPP for video transmission, that we had succesfully developed and tested [6].

Many video streaming systems use the HAS (HTTP Adaptive Streaming) approach, which utilizes the advantages of TCP as the underlying transport layer protocol. MPEG standardized Dynamic Adaptive Streaming over HTTP (DASH) [11] to provide interoperability. While TCP's reliable mechanism provides no packet loss at the application layer, the lost packets at the network layer cause re-transmission, and hence latency, due to this mechanism. On the other hand, UDP provides faster communication since it does not use any re-transmission mechanism or algorithm adjusting the transmission rate, such as congestion control or slow start algorithms of TCP. But UDP suffers from packet losses.

The QUIC protocol was introduced to improve the QoE of web services, by providing HTTP over UDP instead of over TCP. QUIC has been suggested for use in video transmission, but in [12], the authors observe that there is no evidence for any QoE improvement of QUIC, over TCP, in the context of YouTube streaming. In [13], it was found that QUIC is still too reliable for real-time video traffic, with QUIC sometimes performing even worse than TCP for video streaming. Also, the ABR schemes ported to QUIC operate poorly compared to TCP. The authors conclude that reliable transports are ill-suited for video streaming, and suggest an unreliable version of QUIC. In our recent work [6], we demonstrated that BPP is better than TCP and UDP for latency and outages. In this context, New IP and BPP are a good alternative for video streaming systems, especially considering the high precision requirements of emerging and future networking applications.

The emergence of SDN leads researchers to also utilize this technology in video streaming systems. There are many studies in the literature related to scalable video transmission utilizing SDN. These works focus on either routing of the packets [14], [15] or getting assistance from SDN elements [16]. In the studies related to video streaming, SDN and scalable video codec, there is no similar work which focuses on no packet drop at the network layer. The next section presents how BPP should be extended to support video transmission.

III. Enhancements to BPP for Video Transmission

For video streaming applications to get the benefit of using BPP, the video content should be prepared and encoded in a suitable way. The packet preparation method on the sender side should be aware of the BPP transmission process, and packet construction should be done so that some of the BPP chunks can be removed during transmission. In SVC layered coding, each video frame has a base and several enhancement layers, and each layer is multiplexed into a number of packets. In Fig. 1, we present the scheme we devised that maps the 3 layers into the packets created on the server side, and show how each of the packets contain some data from those 3 layers.

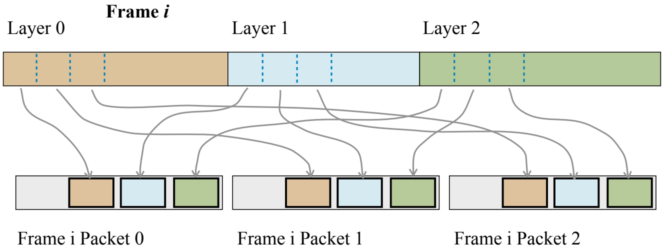


Fig. 1: Mapping Layers of a Frame to Packets

In our implementation, each BBP enabled network device always keeps the base layer (L0), whereas it can drop chunks belonging to enhancement layers (L1 and L2). This ensures seamless video play on the receiver side, as the receiver always gets a packet with some video data (L0), but allowing the decoder to apply the higher encodings onto the base layer.

A. BPP Packet Structure

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 in itself guarantee improved behaviour or performance. We now present a more detailed view of using BPP for video transmission, and show the structure of the BPP packets, plus the processes that are used in network nodes to shrink certain packets, by removing chunks, rather than throwing them away.

The structure of a BPP packet, with the main blocks, is defined in [2], and presented in Fig. 2. We discuss the fields of the BPP packet structure to show how they can be used for carrying layered video, with the existing fields as well as the necessary additional fields for such transmission.

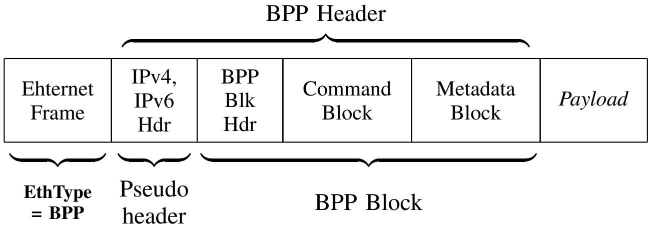


Fig. 2: BPP packet format with BPP block

1) BPP Block: A *BPP Block* is the set of fields which are added to the regular IP packet structure for constructing a BBP packet. The fields of the *BPP Block* provide application related information so that network nodes can use this information as a guidance for packet processing. There are three types of blocks residing within a BPP Block: the *BPP Block Header*, the *Command Block*, and the *Metadata Block*. It is up to the application provider to determine when and how they are used, regarding to the requirements of the applications [2]. Basically, a *Command Block* holds commands and their conditions and parameters, while a *Metadata Block* carries additional metadata. These blocks are optional, as the applications may need to use one, and not the other. A brief explanation of the fields in the BPP block, shown in Fig. 2, are given below.

2) Command and Metadata Block Fields: The Command and Metadata Field blocks hold the most important part of the BPP Block, which provides necessary information to the BPP enabled network nodes, to decide how to act when a BPP packet is received. *Command*, *Condition*, *Threshold value*, and the *Actions* are among the Command and Metadata Field block fields. These blocks also carry additional metadata related to the chunks within the current packet. The *Threshold* is used as the boundary which determines which chunks cannot be dropped. In the Command and Metadata Field blocks, there are also fields which hold the information for each chunk carried within the packet. Off_i and SIG_i represent the offset location in the payload and the *significance value* of $chunk_i$, respectively, while OF_i shows whether $chunk_i$ is dropped. BPP supports various commands, with *Packet Wash* being the one which enables dropping some chunks during the transmission of the packet [4]. This approach helps to prevent dropping the whole packet when the bandwidth is limited, by reducing the size of the packets. Chunks having low-priority can be dropped, and the data arriving at the receiver side provides some usable information. Even though the data is not the same as the data sent by the server, it is better than no data or delayed data [4]. In this work, we directly use this command in the BPP packet transmission, as Packet Wash provides an efficient technique for managing streaming video.

To support the transmission of layered video, the *Command* and *Metadata Block* fields of a BPP frame have been extended to include new fields, whose details will be given next.

B. Additional BPP Block Fields for Video Streaming

In addition to the fields of the BPP Block Header, we added new information showing how many chunks are within the packet. This field, which is named *Chunk Count*, is needed because when the packet is modified during transmission, due to the deletion of some chunks, the SDN controller and the client should be able to understand the ending address of the chunk offsets and the starting address of the payload. The *Chunk Count* is written in the Metadata offset field. The size of the field is 5 bits. In addition to *Chunk Count*, some fields have been added to the fields proposed in [2] and [4], in order to have some video related information about chunks in *Command and Metadata Block.*

The additional fields introduced add a total of 42 extra bits to the BPP Block for each chunk:

$\mathbf{OFF_i}$ (22 bits)			
ChunkOffset (5 bits)	*SourceFrameNo* (12 bits)	*FragmentationNo* (5 bit)	
$\mathbf{CS_i}$ (14 bits)	**$\mathbf{SIG_i}$** (4 bits)	**$\mathbf{OF_i}$** (1 bit)	**$\mathbf{FF_i}$** (1 bit)

Fig. 3: New Chunk Offset Structure

Offset (22 bits): Although the *Offset* field was introduced in [2], it is enhanced here by adding a Source Frame number and a Fragmentation number to the chunk Offset for supporting the video streaming applications. The details of the new chunk *Offset* structure, **$\mathbf{OFF_i}$**, used in this study is shown in Fig. 3, and presents the fields used for each chunk. Some frames might be fragmented into the more than one packets due to their size. The Fragmentation number is used to combine fragments on the client's side.

The Offset field is divided into three subfields: *Chunk Offset*, *SourceFrameNo*, and *FragmentationNo*. The field *Chunk Offset* (5 bits) shows the chunk number within a packet, *SourceFrameNo* (12 bits) shows the frame number carried in a chunk, for complete or fragmented frames, and *FragmentationNo* (5 bits) indicates the fragmentation number that chunk represents, in case the frame is fragmented.

The extended BPP packet structure also uses the following fields to keep information about layered video:
$\mathbf{CS_i}$ (14 bits): Shows the size of the current chunk. A similar field having the same purpose was used in [3].
$\mathbf{SIG_i}$ (4 bits): Shows the significance value of the chunk.
$\mathbf{OF_i}$ (1 bit): This field is set to 1 if the chunk is dropped.
$\mathbf{FF_i}$ (1 bit): Shows if the current fragment is the last one.
The overhead introduced by the proposed BPP header structure stays stable even if the jumbo frames are used.

C. Commands used in the BPP Block for Video Streaming

The *Packet Wash* directive is carried as part of the Command and Metadata Blocks for video transmission. The *Packet Wash* Command is defined as a scrubbing operation that reduces the size of a packet while retaining as much information as possible in qualitative communications. It operates by dropping lower-priority chunks from the payload according to the information carried in the packet header, helping the forwarder to understand the significance of (or the relationship between) the chunks. The lost chunks are not recovered, but some information is usable at the receiver [4]. With the *Packet Wash* operation, some chunks may be dropped from the packet. *Packet Wash* performs selective trimming of the payload, from less significant to higher significant chunks. Accordingly, the forwarding node makes a decision as to which packet to trim, and also which chunks to trim. In case of congestion, a forwarding node can intentionally remove as many chunks containing enhancement layers as necessary [4].

Now that BPP has been presented, plus our enhancements to BPP to be used for video transmission, we now describe how video using BPP is processed by an SDN controller.

IV. The SDN View of Video over BPP

In this section we present how the server constructs BPP packets with the video data, considering the characteristics of the video codec and the different types of video frame, by utilizing the extended BPP packet structure described in the previous sections. We then present how these packets are processed by the SDN controller, during their transmission across the network. The general outline of this process is illustrated in Fig. 4, starting from the packets at the server, through the SDN controllers, to the client. In Fig. 4, the first two network hops are high bandwidth, and so no chunks are dropped, but the last hop the the client is a lower bandwidth, and we see that a Layer 2 chunk has been removed.

A. Video Streaming System Implementation with the Enhanced BPP Packet Structure

The encoded video file information is used for deciding which data will be carried within the chunks, and how the significance values of the chunks (SIG_i) will be determined. Since layered video is transmitted, the importance of each layer and frame type also plays an important role for determining the significance values. Some chunks carry crucial elements, such as I frames, which should *never be deleted*. As such, the most important chunks are those chunks carrying the base layer of I frames. On the other hand, a chunk carrying the highest layer (L2) of a P frame can be discarded without affecting the received quality significantly. These characteristics also used for determining a *Threshold* value, which represents the chunks that should not be removed from the packet.

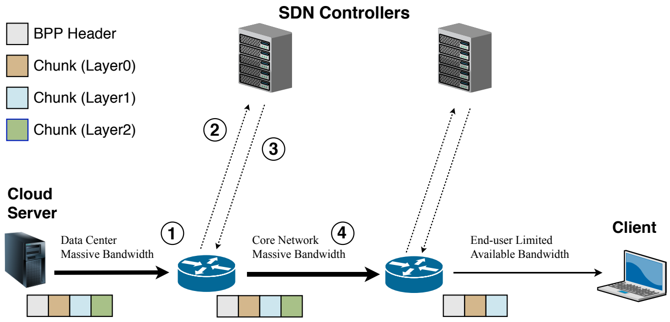

BPP Header
Chunk (Layer0)
Chunk (Layer1)
Chunk (Layer2)
SDN Controllers
②
③
Cloud Server
Data Center Massive Bandwidth
①
Core Network Massive Bandwidth
④
End-user Limited Available Bandwidth
Client

Fig. 4: Scenario with SDN Controllers updating BPP packets

On the server side, we use the described packet filling approach, with a chunk from each of layers 0, 1, and 2, as in Fig. 1, which enables chunk removal during the transmission of the packets. For this purpose, the server partitions each layer of each frame and puts these partitions as the chunks. Each packet carries partitions from each layer, and as some layers can be larger than the others, the chunks within the packets might only carry partitions of subset of layers. Hence in our implementation we have 3 chunks in each packet sent from the server. As there will be some chunks that can be removed from the packet, if the SDN controller decides to shrink the packet, then this will be possible most of the time.

The server constructs the BPP packets based on the semantics of the outlined BPP packet fields. The *Condition* and *Threshold* values are put in the related fields, Although the *Condition* and packet length values are defined in the packet header, the packet processing steps and the semantic of this field are slightly different from work in [4], due to the difference in the requirements of the layered video transmission in this work. The *Condition* field represents the number of packets that will be transferred in one second. In order to signal the importance of each chunk to the SDN controller, the significance values are set based on their importance, determined by the characteristics of the transmitted video. For the i^{th} chunk, the significance value is put into the SIG_i field, and the CS_i, OF_i, and FF_i fields are also filled for each chunk, as well as the OFF_i values.

B. SDN for Layered Video Transmission with BPP

In order to execute the BPP packet processing, BPP enabled switches have been suggested [2]. We used SDN controllers and OpenFlow enabled switches which jointly facilitate BPP enabled switches. When video packets are received by the switches ①, they are sent to the SDN controller to be processed ②. The SDN controller decides *(i)* whether the packet should be chopped and, *(ii)* which chunk(s) should be removed from the packet. The modified packet is sent back to the switch ③ and then forwarded to the output link ④.

The SDN controller has several modules, including measuring the active traffic amount on the links, and calculating the available bandwidth of the paths. The available bandwidth values are calculated by smoothing the amount of traffic values which are periodically obtained from the switches.

The SDN controller decides to modify the packets by using its information about the available bandwidth. The SDN controller drops some chunks within a packet, if the total amount of data transferred in a specific time period exceeds the available bandwidth. This time period is selected as 1 second, to be aligned with the information provided by the server, in the *Condition* field. The controller checks the condition for each packet, given the formula (1), where L represents the packet length, and $AvailableBW$ is the available bandwidth that is estimated by the controller. If the condition occurs, then packet wash processing is performed by the controller.

$$L > \frac{AvailableBW}{Condition} \quad (1)$$

The next step is to determine how many chunks, and which chunks, should be removed from the packet. The number of chunks that will be removed from the packet depends on the size of and the significance value of each chunk. The controller calculates an upper limit for the size of the current packet and it also checks each packet to determine which chunks will be dropped. This upper limit equals L, the value in equation (1). The chunks are listed in descending order, using their significance values. The controller deletes the chunks starting from the beginning of the list, until the packet length reduces below the limit, with chunks being dropped. Note that, the chunks whose significance values are lower than the threshold, are not deleted, in any case.

V. Performance Evaluations

Now the SDN controller process has been described, we now utilise all of the techniques, showing the testbed setup, and some comparative performance measurements.

A. Evaluation Settings

For the evaluation of the study, the Mininet emulation platform was used. Mininet is a widely used emulation platform for testing SDN-based approaches and evaluating their performance. For the SDN controller, the ONOS framework is used. With the built-in modules of ONOS, it is possible to collect traffic statistics, which helps to measure available bandwidth and to reactively forward the flow of packets, as well as communicating with the OpenFlow enabled switches.

In this evaluation, the BPP packets are constructed within Ethernet packets whose MTU size is 1500 bytes, although BPP enables the use of jumbo frames. The video used in the evaluation is encoded so that the length and the bitrate of each video layer are smaller than usual, for experimental purposes. The video file is of 10 seconds duration, and consists of 300 frames, each of which are encoded by using one base and two enhancement layers. The bitrate of the base (L0), the first and the second enhancement layers (L1 and L2) equals to 204 Kbps, 488 Kbps, and 1094 Kbps, respectively. It is possible to send a video with higher bitrate in a real implementation.

B. Comparative Performance Measurements

In order to provide comparative performance measurements, we also test the streaming of the layered video over UDP. The performance of BPP and UDP is tested in different networks, where in the first set of tests, the bandwidth values of the path between the server and the client are fixed, and for the second set of tests, the available bandwidth changes over time. While BPP packets are constructed as described previously, with UDP, the server puts the maximum number of video layers to fit in the packet size sequentially, starting from the base layer for each frame. We measure two important video QoE metrics: the outage duration (in msec) and PSNR (in dB).

SVC video is encoded by using the similarities between consecutive frames, as well as the dependency between the layers of the same frames, so 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. In Fig. 5a, the PSNR values versus frame number, are given for both BPP and UDP, where the bandwidth is 0.5 Mbps. As can be seen, while the BPP client plays the video with a stable quality for all frames, the UDP client could only decode and play the video until its 81^{st} frame. Furthermore, the PSNR values of the previous frames changes a lot, which shows there was not a seamless playback. In images Fig. 5b and 5c, the same frames that BPP and UDP clients play are displayed. The positions of these frames in the video stream are indicated in Fig. 5a with a yellow rectangle. The effect of missing reference frames can be seen from the screenshot of the UDP client. In this test, the UDP client showed video for 4 seconds, and the frame in Fig. 5c stayed on the screen until the end of the video.

In table Ia, the observed outage durations for different bandwidth settings are presented for both protocols. Outage duration is one of the most important QoE metrics affecting the perceptual quality. If the outage values in the table are examined, it is seen that the clients using BPP did not experience significant outages, the highest outage is 0.2 seconds. On the other hand, we observed up to 5.47 seconds of outages with UDP when bandwidth is 0.5 Mbps. Considering that the duration of the video is only 10 seconds, having 5 seconds of outage 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.

The PSNR of the original encoded video file is 44 dB, and the average received PSNR values are listed in table Ib. We see that using both protocols, the clients play the video with the highest possible quality when the bandwidth is 1.5 Mbps, 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. On the other hand, UDP can not provide consistent and good quality in all scenarios. In ascending bandwidth scenario, we observe the average PSNR value is 18 dB, which shows the video quality is not acceptable.

When we examine the tests results, we see that the clients both continue to receive video layers until the end of the streaming session. However, the PSNR values in Fig. 5a for

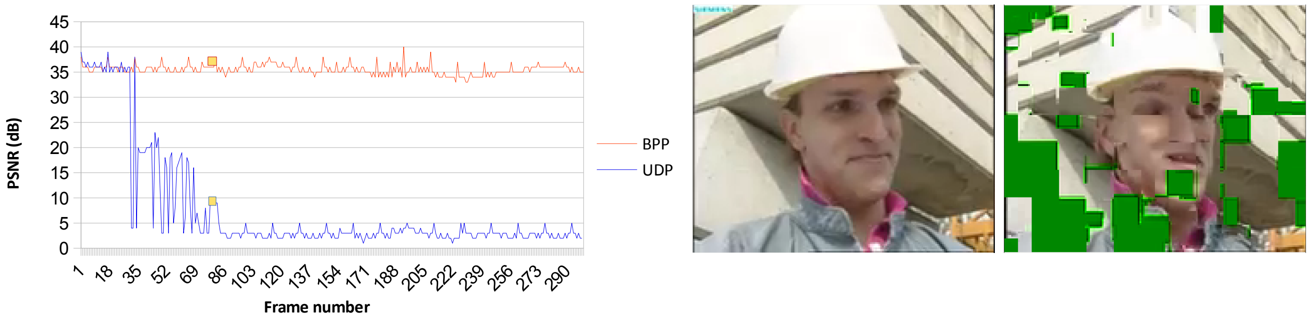
	BPP	**UDP**
0.5 Mbps	173	5470
0.8 Mbps	205	2634
1.5 Mbps	0	0
Ascending	0	916
Descending	0	920

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

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

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

TABLE I: Comparing BPP vs UDP QoE Metrics



(a) Comparative PSNR results (Fixed bandwidth: 0.5 Mbps)

(b) Frame 81: The screenshot from the BPP client's video

(c) Frame 81: The screenshot from the UDP client's video

Fig. 5: Perceptual Quality: BPP vs UDP

UDP shows that the client cannot play the video after a few seconds. The reason that the video cannnot 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 bandwdith limitations. Consequently, 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 SDN and BPP provides higher effective bandwidth utilization, by adjusting the quality during transmission and sending the higher layers only if there is enough capacity.

VI. Conclusions

We presented how BPP is utilized for video streaming systems and showed the necessary BPP packet enhancements in order to obtain an operational system. We provided additional fields to BPP, presenting the commands, as well as field sizes. During the transmission of the packets, the SDN controller removes chunks when the network is congested. Hence, all packets are transmitted to the client with a possible quality degredation, but without any re-transmission, delay, or loss.

In the paper, we have shown that scalable video is a good option that is highly compatible with BPP. In order to utilize BPP, the server should be aware that some parts of the packets can be removed during transmission, hence the video layers are partitioned into the packets in such a way that each packet carries each type of layer. This allows the SDN controller to be able to shrink the packet when it is needed.

The tests that were conducted, using various network conditions, showed that BPP significantly outperforms UDP in terms of PSNR and duration of pauses. Even when the network capacity is highly limited, the observed duration of pauses was under several hundreds of milliseconds, and the clients continue to play the video without any frame loss. The performance evaluation confirms that BPP has huge potential to provide low latency requirements and high precision services.

In future work, we plan to embed these BPP enhacement fields presented here into the New IP packet structure. We will also utilize the other advantages of New IP for video streaming systems, developed for multicast and broadcast transmission.

References

[1] R. Li, K. Makhijani, and L. Dong, "New IP: A Data Packet Framework to Evolve the Internet : Invited Paper," in *2020 IEEE 21st Intl. Conf. on High Performance Switching and Routing (HPSR)*, 2020, pp. 1–8.

[2] 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*, August 2018.

[3] 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.

[4] 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*, August 2019, pp. 22–28.

[5] S. Clayman, "The Inter-Dependence of Network Transport and Application Behaviour," in *ITU Network 2030*, Geneva, October 2019. [Online]. Available: https://www.itu.int/en/ITU-T/Workshops-and-Seminars/2019101416/Documents/Stuart_Clayman_Presentation.pdf

[6] S. Clayman, M. Tuker, H. Arasan, and M. Sayıt, "Managing Video Processing and Delivery using Big Packet Protocol with SDN Controllers," in *IEEE Conf. on Network Softwarization – Netsoft*, Tokyo, July 2021.

[7] A. Clemm, M. Zhani, and R. Boutaba, "Network management 2030: Operations and control of network 2030 services." *Journal Network System Management*, vol. 28, pp. 721–750, 2020.

[8] 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.

[9] L. Dong and R. Li, "In-packet network coding for effective packet wash and packet enrichment," in *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.

[10] A. Clemm and T. Eckert, "High-precision latency forwarding over packet-programmable networks," in *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, 2020, pp. 1–8.

[11] "MPEG-DASH: Dynamic Adaptive Streaming over HTTP." [Online]. Available: https://mpeg.chiariglione.org/standards/mpeg-dash

[12] M. Seufert, R. Schatz, N. Wehner, and P. Casas, "QUICker or not? -an Empirical Analysis of QUIC vs TCP for Video Streaming QoE Provisioning," in *22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 2019, pp. 7–12.

[13] M. Palmer, T. Krüger, B. Chandrasekaran, and A. Feldmann, "The QUIC Fix for Optimal Video Streaming," in *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, ser. EPIQ'18. ACM, 2018, p. 43–49.

[14] R. S. Kalan, C. Cetinkaya, and M. Sayit, "Design of a layer-based video streaming system over software-defined networks," in *2017 8th Intl. Conf. on the Network of the Future (NOF)*, 2017, pp. 8–13.

[15] T. Uzakgider, C. Cetinkaya, and M. Sayit, "Learning-based Approach for Layered Adaptive Video Streaming over SDN," *Comput. Netw.*, vol. 92, no. P2, pp. 357–368, Dec. 2015. [Online]. Available: https://doi.org/10.1016/j.comnet.2015.09.027

[16] F. Tashtarian, A. Erfanian, and A. Varasteh, "S2vc: An sdn-based framework for maximizing qoe in svc-based http adaptive streaming," *Computer Networks*, vol. 146, pp. 33–46, 2018.