Dynamic Packet Content Construction and Processing for End-to-End Streaming in 6G

Stuart Clayman†, Emre Karakış*, Mustafa Tüker*, Elif Ak‡, Berk Canberk§, Müge Sayıt*
†Dept. of Electronic Engineering, University College London, London, UK
*International Computer Institute, Ege University, Izmir, Turkey
‡Department of Engineering, Istanbul Technical University, Turkey
§School of Computing, Engineering and The Built Environment Edinburgh Napier University, UK

***Abstract*—In the context of 6G, the use of drones / UAVs and satellite is a high priority. One of the main issues is that there is limited and varying bandwidth in these environments, so the question arises: how do we provide high Quality of Experience (QoE) to the users. BPP is a recent protocol which is effective when used with Scalable Video Coding (SVC) streams and limited bandwidth environments. We present an end-to-end architecture, with a drone sending video, utilizing functions for dynamically constructing the content of packets, and then dynamically processing those packets during their transmission across a network, all managed by a multi-domain orchestrator. These functions are implemented as virtualized network elements, as in our previous work. In this current work, we investigate how different packing strategies for filling packets impact different QoE parameters, when evaluated using a number of different bandwidths. These insights can be utilized for choosing the best QoE, and will be especially useful in 5G / 6G environments.**

***Index Terms*—6G, In-Network Packet Processors, Programmable Protocols, Packet Trimming, Traffic Management**

I. Introduction

As 5G is now being deployed, future enhancements in networks in the form of 6G look at performance improvements through various approaches. These include the integration of previously separate technologies, the use of edge computing to ensure overall throughput and for low latency and reliable communications, further support for machine-to-machine communication necessary for IoT usage, the extended use of virtualized elements across a number of different cloud providers, the integration of high-performance computing, and networks spanning UAV / drone network and satellites [1]. The addition of drones and satellites to a network will increase the scope of devices, including sources of data and the number of links. However, these links often have limited, and sometimes variable, bandwidth. Given such conditions, certain applications can be made more effective by taking into account these bandwidth issues.

In this paper we present an end-to-end scenario with a drone sending video, utilizing techniques for dynamically constructing the content of packets and then dynamically processing those packets during their transmission across a network, within the context of an end-to-end video streaming application. The goal is to have improved QoE attributes, where there are bandwidth limitations, compared to a number of existing techniques. The use of virtualized functions at the sender side and in the network is facilitated as part of the mechanisms to achieve this goal. These concepts are part of a vision for the operations and control of services in Network 2030 [2]. For the packet construction we utilize the Big Packet Protocol (BPP) protocol [3], which is a protocol designed to support programmable functions in the network, and in particular enables the trimming of packets as they pass through the network, from the source to the destination. We have previously considered the efficiency and effectiveness of using the BPP protocol for transmitting video, and we have observed that H264 SVC video is well matched with BPP. Furthermore, we devised a number of strategies for multiplexing the layers of the SVC video into the sequence of BPP packets [4]. In support of this approach, we presented a system that utilizes BPP for sending SVC video, and have shown how the *Packet Wash* process of BPP can be used for trimming video packets during transmission, when the available bandwidth is limited, but still having good QoE attributes [5]. Here we extend our previous work, and present new work, showing a more detailed view of the effects on QoE of these different packing strategies.

We propose an architecture that encompasses a drone network, a 5G / 6G core network, and an edge network with the clients. There is a multi-domain orchestrator to collect data and pass on control messages between the participating domains. This enables the signaling of the sender side, to select the most appropriate packing strategy from these different packing strategies, considering the expected performance at the receiver. To facilitate the packet creation at the sender, and the packet trimming during transmission, we utilize virtualized in-network functions. The architecture together with an evaluation of the packing strategies are presented, in the context of different bandwidths available at the receiver.

We observe various QoE parameters, as well as overall QoE values under many different bandwidth conditions. The main contribution of this paper is the use of these different packing strategies at the sender, and how they have an impact on the different QoE parameters, given that there can be processing of the packets as they pass through the network. These results are not obvious, just by considering the approach taken by each strategy. Furthermore, none of these strategies can be done with UDP or TCP, as UDP is too lossy and TCP is too reliable. We focus on evaluating the different strategies for packet filling, in the context of one drone, to determine the impact of these various strategies on the different QoE parameters of the received video.

II. Background

1) Video Streaming by Using Drones: Streaming video from drones is currently of interest. Video transmission between drones was proposed in [6], where a system was designed where users across the network could use drone, cellular, or a WiFi connection depending on the throughput value. In [7], the clients can signal their bandwidth and delay values to the drone, which then transfers video to the clients, and the drones can use this data to adapt quality. In [8], the QoE metrics such as latency and duration of pauses from the videos captured by drones are tracked by a MEC server to which the drones are connected. In [9], drones capture different areas in a sport field, where they collect the video on the basis of the signals sent by the motion sensors on the ball or on the players. In that work, the drone cameras move and capture the video depending on the information they receive from the motion sensors. No broadcasting was not undertaken in that study. The use of cameras on drones for streaming video can be used in many scenarios, as seen in the literature. However, except our paper [10], none of these studies consider the transmission of end-to-end video.

2) Streaming SVC Video: Scalable video coding (SVC), sometimes called Layered Video Coding, as the video is encoded as layers. It is a beneficial approach as it takes advantage of similarities between the different versions of the same frame, as well as being able to reassemble successive frames. There are different types of quality layers in this video encoding, including temporal quality layers and spatial quality layers. A detailed view of the mechanisms and techniques used for transmitting SVC video across the network, using the multi-chunk facilities and the packet trimming capabilities of BPP, is presented in our paper [4]. In that paper, we present details of mapping SVC video to BPP packets using different packing strategies, and then show an implementation of streaming video using BPP, where some QoE parameters are observed in very limited and fixed bandwidth conditions for the different packing strategies. Also, in [10], we utilise SVC video from drones combined with BPP transmission.

To deal with varying network behaviour and loss changes over time, in our work with BPP we utilise *packet trimming* [4]. Alternative approaches include Dynamic Adaptive Streaming over HTTP (DASH) [11], which is a standard developed by MPEG. It provides interoperability between the participants of HTTP Adaptive Streaming (HAS) deployments, as it runs over HTTP. With DASH, the client can dynamically adapt the quality of the video by requesting segments of a different quality. To deal with changing bandwidths, the original video is encoded multiple times with a set of qualities. Higher qualities have higher data rates, and bigger files, and use more bandwidth. There are also implementations that evaluate the behaviour of DASH when using QUIC as a transport. In [12], the authors evaluated sending SVC video over DASH.

3) Using BPP for SVC Video: In many streaming applications, the packets are constructed in a linear sequence, taking a number of consecutive bytes directly from the content of a video file, and then transmitted. Depending on the protocol being used, either the packet is lost (using UDP), or it is retransmitted (using TCP). In our system, we utilize SVC video which has multiple layers [13]. BPP allows packets to have a number of *chunks*, which can be *trimmed* by a network node during transmission. Although we could have a simple packet filling technique, the use of layered video gives us a number of options for multiplexing parts of these layers and packing that data into BPP chunks. In [4], we describe these different packing strategies, and has some initial results for those strategies, when using very limited, fixed bandwidths.

Recently we evaluated using BPP and video for an end-to-end transmission scheme [10], from multiple drones / UAVs to clients in various environments. That approach only used one strategy for filling packets, but all 4 approaches are used here.

4) Orchestration: Orchestrators can have a large number of functions, and more advanced ones allow services that are deployed across distributed domains, where they can be configured in a north-south way creating a hierarchy of service provision capabilities, rather than configured in the more common peer-to-peer approach [14]. This approach works particularly well where each domain, from the mobile edge, to the core DC, can be managed independently of the others. They allow virtualized network services, fully integrated with a service platform, using a modular and flexible architecture [15]. These setups become even more complicated when the multi Point-of-Presence (PoP) situation occurs [16].

Using our previous work with BPP, plus virtualized functions, acts as a foundation for extending the implementation of the architecture to include video data sources, and specifically drones, to evaluate a full end-to-end system with orchestration. We previously presented some parts of the architecture used here. We have also described the effects of running virtualized in-network quality adaption by trimming the packets of layered video streams at the edge [5].

III. End-to-End Video Streaming Architecture

This section presents the architecture for end-to-end delivery of streaming video from a drone network to the clients. The architecture of the framework is presented in Fig. 1. Video is sent from a drone to a Drone Control Unit, and then onto a virtualized *Packet Creation Function* which takes the SVC video and creates BPP packets. These are forwarded over the provider's network towards the client. At the client edge network, a virtualized *Packet Trimming Function* intercepts the packets during their transmission over the network, and these video packets can be modified by using the packet trimming capabilities of the BPP protocol [5]. In the system, the sender always transfers the video at the highest quality, and we rely on the *Packet Trimming Function* to address the client aspects of network behaviour and adaptiveness.

A multi-domain orchestrator interacts with the Controllers and Servers in the domains. The orchestrator manages the end-to-end video transmission by considering the network resources, the video parameters, and the client characteristics. All the system components are in communication with the orchestrator. In this paper, the main functional elements are

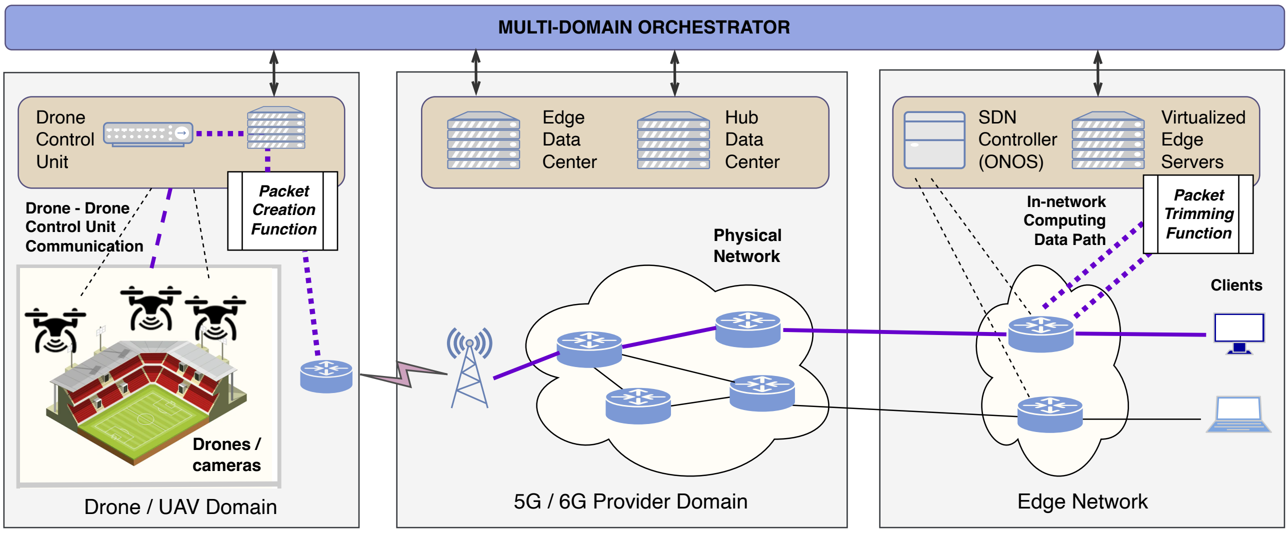


Fig. 1: The Architectural Framework

presented, and just a single path from one drone to one client is evaluated, shown by the purple line.

A. The Virtualized Packet Creation Function

The *Packet Creation Function* is a virtualized element which runs in the UAV Domain. It takes the SVC video stream and uses one of the 4 different packing strategies to create BPP packets. The JNSM paper [4] has all the details of this operation. Here is a brief overview of the 4 packing strategies devised. Each one takes data from the 3 layers of the SVC video stream, and creates 3 chunks in each BPP packet, as shown in Fig. 2.

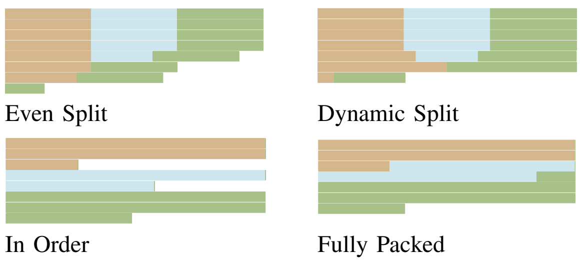


Fig. 2: Different packing strategies have different chunks

The data presented in the figure above shows a sequence of packets for each of the 4 strategies, producing a different number of packets and varying packet sizes, all from a single frame of video which has the following content: Layer 0: 3232 bytes, Layer 1: 2232 bytes, and Layer 2: 3527 bytes. We use a 1500 byte packet, providing a content size of 1472 bytes. This allows for the BPP header and BPP meta-data, before allocating data to the chunks of video. So, depending on the size of each frame, the size of each layer in the frame, and the selected packing strategy, there will be different sized packets. The created packets are forwarded onwards.

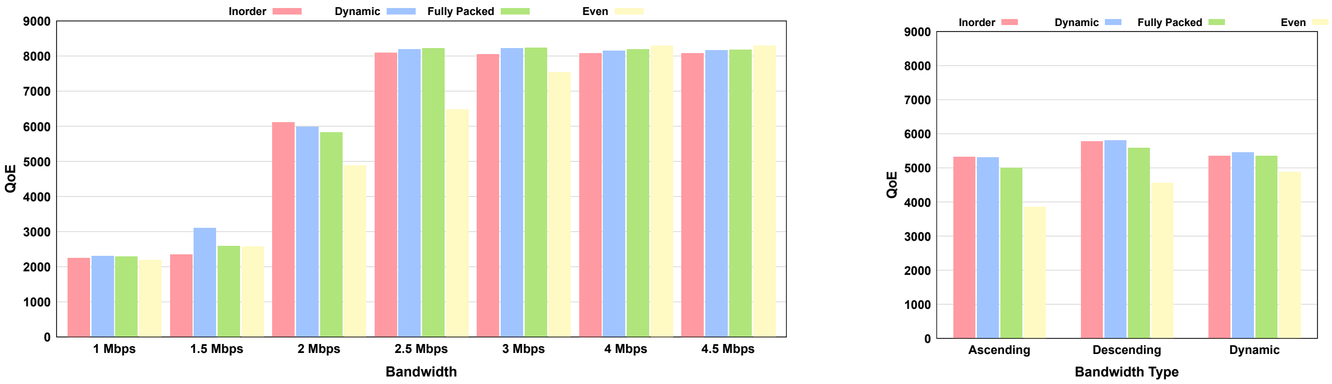
B. The Virtualized Packet Trimming Function

The SDN controller sets a rule so that after the BPP packets have crossed over the provider's network and the edge network, they are sent to a virtualized *Packet Trimming Function*. This function uses an algorithm to decide if, and how many, chunks should be removed from a packet, based on the available bandwidth. For each packet received, the algorithm checks the amount of traffic sent in the current time period. If the number of bytes sent is greater than the available bandwidth, it determines that bytes should be trimmed. The trim operation is done by removing some chunks based on their significance to the content [4]. As only whole chunks can be removed, not an arbitrary number of bytes, and as the size of the chunks that can be trimmed can be different to the number of bytes that need to be trimmed, it can take a number of packets of trimming before it is possible to match the bandwidth level. However, if the number of bytes sent are less than or equal to the available bandwidth, then the received packets are forwarded without implementing packet trimming. All packets of content must be passed through the trimming process to ensure that the video is sent in a quality suitable for the bandwidth between the clients and the edge network.

C. The Role of the Multi-Domain Orchestrator

The Multi-Domain Orchestrator has different roles in each of the different domains. In the Edge Domain, the orchestrator manages the in-network computing process for the virtualized Packet Trimming Function during setup and during transmission, by considering the bandwidth values of the paths and the client capabilities. The SDN controller sets up the paths and also periodically sends the network conditions and available bandwidth information of the paths to the orchestrator. This information will be utilised to manage the Packet Trimming Function, as described in [5].

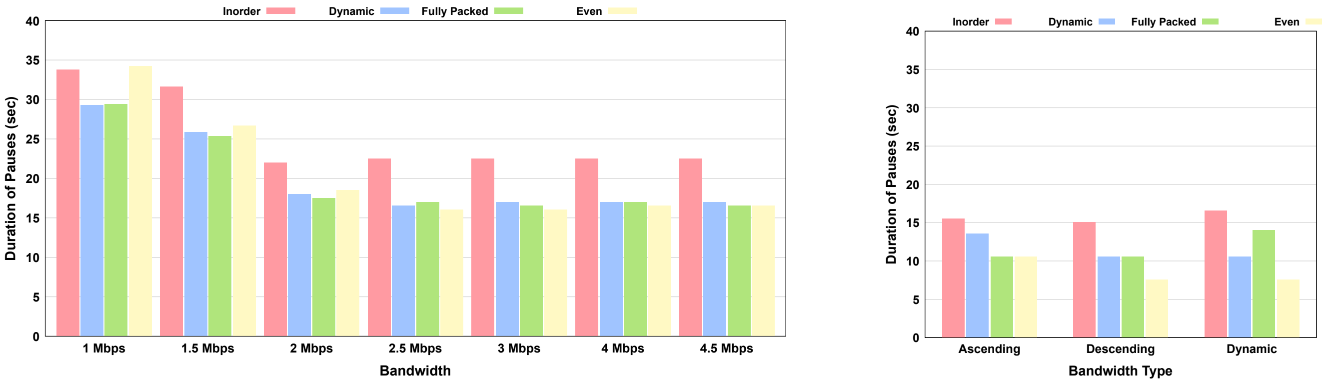
In the Drone Domain, the orchestrator has the responsibility for defining the parameters for some components, namely the Drone Control Unit and the virtualized function. It signals the Packet Creation Function about the most appropriate packing strategy based on the conditions when the streaming starts. In a fully integrated system the orchestrator would interact with the management of the Provider Domain, but in this setup no functionality is used.



(a) *Fixed Bandwidth Configuration*

(b) *Varying Bandwidth Configuration*

Fig. 3: Overall QoE Results by Packing Strategies with Fixed and Dynamic Configuration



(a) *Fixed Bandwidth Configuration*

(b) *Varying Bandwidth Configuration*

Fig. 4: Total Duration of Pauses by Packing Strategies with Fixed and Dynamic Configuration

IV. Performance Evaluations

A. *Experiment Environment and Settings*

To evaluated the performance of transmission with BPP and the different packing strategies, we conducted several experiments using Mininet emulation environment, combined with the SDN controller in our experimental setup. We used virtualized Packet Creation and Packet Trimming functions.

To observe the performance under different network conditions, we use two sets of bandwidth values in our experiments. In the *fixed bandwidth* setting, the values between 1 and 4.5 Mbps are used, whereas in the *varying bandwidth* settings, the bandwidth value fluctuates between 1 and 3 Mbps during the video session. We use three different scenario for the varying bandwidth settings, namely *ascending*, *descending*, and *dynamic*. In the ascending scenario, the bandwidth value is 1 Mbps at the beginning of the streaming session, increases up to 2 Mbps, and then to 3 Mbps. The dynamic scenario begins at 3 Mbps, decreases to 1 Mbps, and then increases to 2 Mbps. In the descending scenario, the bandwidth initiates at 3 Mbps, decreases to 2 Mbps, and then down to 1 Mbps. Each bandwidth change occurs with a 14 second interval. The total video duration is 42 seconds. All the details of the SVC encoded video, that is used in the experiments, are shown in Table I.

Video type	H.264	**Frame rate**	24 fps
Resolution	480 x 360	**L0 bitrate**	897.66 Kbps
Frame No	1000	**L0+L1 bitrate**	1927.29 Kbps
Duration	42 sec	**L0+L1+L2 bitrate**	4384.19 Kbps

TABLE I: Encoded Video Details

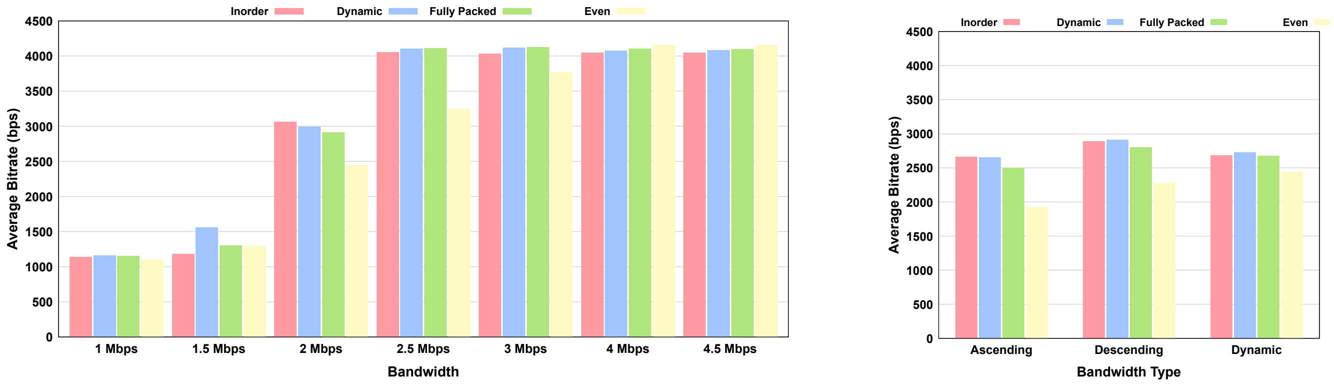
Packing Strategy	Packets	Packing Strategy	Packets
Even Split	28560	Dynamic Split	17638
In Order	18626	Fully Packed	17624

TABLE II: Number of Packets for Packing Strategies

The Packet Creation Function generates a different number of packets for each of the 4 packing strategies, as shown in Table II. As BPP has a meta-data and chunk structure, the volume of traffic transferred is greater than the content being read. This has an impact in limited bandwidth conditions.

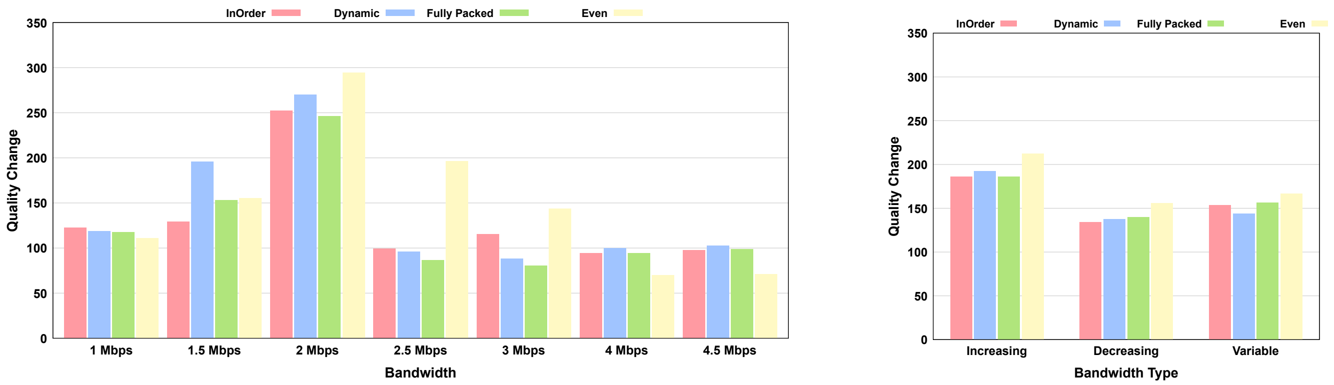
B. *Comparison of Transmission with the Packing Strategies*

Fig. 3 represents the overall QoE values observed in the experiments with different bandwidth settings. These values are calculated by using a weighted linear function, where the weight of each QoE parameter is determined according to the impact of that parameter to the perceived quality [5]. It is observed that the Dynamic Split strategy provides higher QoE compared to other packing strategies in the experiments conducted with low bandwidth (1-1.5 Mbps) as shown in



(a) *Fixed Bandwidth Configuration* (b) *Varying Bandwidth Configuration*

Fig. 5: Total Average Bitrate Results by Packing Strategies with Fixed and Dynamic Configuration



(a) *Fixed Bandwidth Configuration* (b) *Varying Bandwidth Configuration*

Fig. 6: Total Quality Change Results by Packing Strategies with Fixed and Dynamic Configuration

Fig. 3a. This strategy is always among the packing strategies providing the best QoE in all scenarios, including fixed and varying conditions. Although overall QoE values present a generalized QoE value, the perceived quality differs for each user and users might be affected by different QoE parameters with different level. Therefore, as well as overall QoE values, we also show each QoE parameters in the following graphs.

As seen in Fig. 4a the Dynamic Split strategy is among the strategies having low total Duration of Pauses in fixed bandwidth settings, whereas it is one of the packing strategies that has the highest total bitrate as seen in Fig. 5a and 5b. The In Order strategy is less good for Duration of Pauses. The Dynamic Split strategy has similar bitrate and QoE values with In Order and Fully Packed strategies in all experiments.

As seen in the Packet Count table above, the high number of packets produced by the Even Split strategy leads to an increase in packet header overhead, which becomes a disadvantage over limited bandwidth conditions. Consider the **L0+L1 bitrate** value used for the experiments, which is a bit under 2Mbps. When examining the results of the 2Mbps experiment, we observe from Fig. 6a that the Quality Change values of all packaging strategies are high. The reason for this is that the bandwidth becomes insufficient to carry all L1 frames because the headers are a significant overhead added to the packets. Therefore, only a limited number of L1 packets can be transferred, which causes a large number of quality changes due to the number of transitions between L0 and L1 frames when the client plays the video.

Fig. 6a shows that the Quality Change value decreases for all packing strategies when a bandwidth of 2.5 Mbps is used, which verifies if the bandwidth is enough for carrying video data and the headers. If so, the number of Quality Changes decrease. The minimum decrease is observed with the Even Split strategy, due to the large number of packets. Especially, for the tests performed using 4 and 4.5 Mbps bandwidth, the number of quality changes for all packing strategies is minimized. Despite the large number of packets created in the Even Split strategy, it has the lowest value with regard to total Duration of Pauses for the experiments having the fixed bandwidth 2 Mbps and above. When the bandwidth increases to greater than or equal to 4 Mbps, the Even Split strategy reaches the highest values in terms of QoE as seen in Fig. 3a. This packing strategy gives better results at high bandwidths.

As seen in Fig. 4a and Fig. 4b, the In Order strategy has the highest total Duration of Pauses in the experiments for all bandwidth values, and this situation leads to a significant negative effect on overall QoE value. Since the In Order strategy has the highest bitrate value in the experiment with 2 Mbps, the best QoE is obtained in these network conditions. Furthermore, In Order reaches its best performance in terms of overall QoE for the bandwidth values equal to 2 Mbps and above, however, the other strategies have higher QoE values in the experiments conducted using high bandwidth.

The Fully Packed strategy, having a low total Duration of Pauses value and a high bitrate value, performs a moderate performance in all types of experiments. It has never had the minimum QoE value in any type of experiment. It does not outperform other strategies by a large margin, but good results can be obtained in suitable situations. In most experiments, Fully Packed is one of the strategies having the highest overall QoE result as it can be seen from Fig. 3a. However, the QoE parameters causing this outcome are different from the other strategies. Fully Packed strategy provides high QoE thanks to its low duration of pauses values as seen in Fig. 4a and its low Quality Change values as seen in Fig. 6a.

In the experiments with *varying network conditions*, the highest QoE values are obtained by In Order, Dynamic Split, and Fully Packed strategies whereas we observe that Even Split strategy can not perform as good as in fixed bandwidth scenarios. However, Even Split has the minimum Duration of Pauses values in ascending, descending, and dynamic bandwidth experiments. The number of quality changes is the lowest with Even Split when the bandwidth is greater than or equal to 4 Mbps. These results show that, if the bandwidth capacity is enough to carry the video data plus header or highly dynamic, the Even Split strategy can be used for the users preferring lower duration of pauses and quality changes.

V. Conclusions

In this paper we have presented an architecture that undertakes the orchestration of end-to-end network resources, and contains an Orchestrator which interacts with a Drone Control Unit, a Virtualized Packet Creation Function, an SDN Controller, and a Virtualized Packet Trimming Function at the Edge, in order to provide the control needed to manage all of the disparate elements. This multi-domain orchestrator collects data about network conditions and manages the packet creation strategy on the other end of the network. With BPP and SVC, it is possible to use different packing strategies where the performance of each strategy might differ under different network conditions.

We have extended our previous work, and shown a more detailed view of the effects on QoE of these different packing strategies. The experimental results show that, some strategies might perform better in limited and fixed bandwidth conditions whereas the others might be preferable under varying network conditions. These insights can be utilized in determining the best packing strategy to use for QoE, and will be especially useful in 5G / 6G environments.

In future work we intend to enhance the multi-domain orchestrator functionality to provide a control loop and allow for dynamically signaling the Packet Creation Function, based on the real-time client information of the available bandwidth. This would provide a mechanism to deliver video with the *best* packet strategy for the observed conditions.

Acknowledgment

This work is funded by TUBITAK Electric, Electronic and Informatics Research Group (EEEAG) under grant 121E373. Stuart Clayman is funded by the TUDOR project (Dept. for Science, Innovation and Technology under the Future Open Networks Research Challenge).

References

[1] "Solutions for NR to support non-terrestrial networks (NTN)," 3GPP, Tech. Rep. TR38.821, 2019.

[2] 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.

[3] 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.

[4] S. Clayman and M. Sayıt, "Low Latency Low Loss Media Delivery Utilizing In-Network Packet Wash," *Journal of Network and Systems Management*, vol. 31, no. 1, p. 29, 2023. [Online]. Available: https://doi.org/10.1007/s10922-022-09712-1

[5] M. Tüker, E. Karakış, M. Sayıt, and S. Clayman, "Using Packet Trimming at the Edge for In-Network Video Quality Adaption," *Annals of Telecommunications*, 2023, Springer Nature.

[6] C. Singhal and B. N. Chandana, "Aerial-son: Uav-based self-organizing network for video streaming in dense urban scenario," in *2021 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 2021, pp. 7–12.

[7] J. Molina, D. Muelas, J. E. L. De Vergara, and J. J. García-Aranda, "Network quality-aware architecture for adaptive video streaming from drones," *IEEE Internet Computing*, vol. 24, no. 1, pp. 5–13, 2020.

[8] C. Qu, P. Calyam, J. Yu, A. Vandanapu, O. Opeoluwa, K. Gao, S. Wang, R. Chastain, and K. Palaniappan, "Dronecoconet: Learning-based edge computation offloading and control networking for drone video analytics," *Future Gener. Comput. Syst.*, vol. 125, no. C, p. 247–262, dec 2021.

[9] X. Wang, A. Chowdhery, and M. Chiang, "Networked drone cameras for sports streaming," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 308–318.

[10] E. Karakış, S. Clayman, M. Tüker, E. Bozkaya, and M. Sayıt, "Towards High Precision End-to-End Video Streaming from Drones using Packet Trimming," in *PVE-SDN, IEEE Conf. on Network Softwarization – Netsoft*, Madrid, Spain, 2023.

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

[12] C. Kreuzberger, D. Posch, and H. Hellwagner, "A Scalable Video Coding Dataset and Toolchain for Dynamic Adaptive Streaming over HTTP," in *Proceedings of the 6th ACM Multimedia Systems Conference*, ser. MMSys '15, T. O. Wei, Ed. New York, NY, USA: ACM, mar 2015, pp. 213–218. [Online]. Available: http://concert.itec.aau.at/SVCDataset/

[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] F. Tusa, S. Clayman, D. Valocchi, and A. Galis, "Multi-domain orchestration for the deployment and management of services on a slice enabled nfvi," in *2018 IEEE Conf. on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2018, pp. 1–5.

[15] S. Dräxler, H. Karl, M. Peuster, H. R. Kouchaksaraei, M. Bredel, J. Lessmann, T. Soenen, W. Tavernier, S. Mendel-Brin, and G. Xilouris, "Sonata: Service programming and orchestration for virtualized software networks," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2017, pp. 973–978.

[16] M. Peuster, S. Draxler, H. R. Kouchaksaraei, S. v. Rossem, W. Tavernier, and H. Karl, "A flexible multi-pop infrastructure emulator for carrier-grade mano systems," in *2017 IEEE Conference on Network Softwarization (NetSoft)*, 2017, pp. 1–3.