Towards High Precision End-to-End Video Streaming from Drones using Packet Trimming

Emre Karakış*, Stuart Clayman†, Mustafa Tüker*, Elif Bozkaya‡, Müge Sayıt*
*International Computer Institute, Ege University, Izmir, Turkey
†Dept. of Electronic Engineering, University College London, London, UK
‡Dept. of Computer Engineering, National Defence University Naval Academy, Istanbul, Turkey

***Abstract*—The emergence of a number of network communication facilities such as Network Function Virtualization (NFV), Software Defined Networking (SDN), the Internet of Things (IoT), Unmanned Aerial Vehicles (UAV), and in-network packet processing, holds a potential to meet the low latency, high precision requirements of various future multimedia applications. However, this raises the corresponding issues of how all of these elements can be used together in future networking environments, including newly developed protocols and techniques. This paper describes the architecture of an end-to-end video streaming platform for video surveillance, consisting of a UAV network domain, an edge server implementing in-network packet trimming operations with the use of Big Packet Protocol (BPP), utilization of Scalable Video Coding (SVC) and multiple video clients which connect to a network managed by an SDN controller. A Virtualized Edge Function at the drone edge utilizes SVC and in communication with the Drone Control Unit to manage the transmitted video quality. Experimental results show the potential that future multimedia applications can achieve the required high precision with the use of future network components and the consideration of their interactions.**

***Index Terms*—Future Networks, High Precision, In-Network Programmable Protocols, Packet Trimming**

I. INTRODUCTION

As one of the most popular Internet applications, video streaming applications are used for a wide range of scenarios, from entertainment to surveillance to public safety or remote surgery. The requirements of these applications including high precision, low latency and low loss might become challenging with the use of traditional network technologies and protocols. Emerging network network communication facilities such as Network Function Virtualization (NFV), Software Defined Networking (SDN) and Internet of Things (IoT) can help provide solutions that will meet such requirements and improve the Quality of Experience (QoE) of such applications. More recently the facility to do in-network packet processing gives rise to the development of new applications for various future network scenarios. The combination of all of these elements raises a number of corresponding issues as to how all of these elements can be used together in future networking environments, including the use of cross-layer design techniques and newly developed protocols.

Over the last years, the application scenarios of Unmanned Aerial Vehicles (UAVs) / drones have drawn considerable attention from both industrial and military fields. Due to their advantages of low cost, highly mobile, and flexible deployment, these drones can: capture images and video by monitoring a surveillance area and transmit that data to a ground control station, which is very significant for future IoT. Consequently, aerial platforms / drones can be used in various fields, such as environmental monitoring, disaster assistance, aerial photography, and so on [1] [2].

Recent developments in network technologies and increasing user demands have led to the introduction of various future network services, such as applications containing haptic or holographic communication, and these applications need advanced network services that provide high precision, low loss, and low latency. Big Packet Protocol (BPP), which has been proposed for this [3], is a programmable network protocol which enables the trimming of packets on their journey through the network, from source to destination [4]. In this paper, we present and evaluate a potential system design which has future network components, such as softwarized and virtualized networks, with the use of BPP, plus transmission and processing of video streams from UAV devices. We focus on enhancing the Quality of Experience (QoE) and providing low latency and low loss for video streaming applications for drones which are utilizing future networks.

In this paper, we consider a typical scenario of drone monitoring for a surveillance area, whereby the surveillance area is covered by multiple drones, and the sequence of videos are captured by the drone cameras and transmitted to a virtualized MEC server for processing. Although video streaming from drones has gained great attention recently, there has been no study that considers end-to-end video transmission, where the videos captured by the drones are streamed to a set of clients. We believe this is the first paper that uses emerging network components and techniques, combined with IoT drones and programmable network protocols, to provide low latency and high reliability for end-to-end video streaming. There is a particular focus on the efficient transfer of video streams, utilizing Scalable Video Coding (SVC), while applying traffic adaptation and Packet Trimming during transmission. SVC video allows the encoding and extraction of video sequences at different qualities from one encoded video file. The encoded video file consists of a number of layers, where the first layer provides the lowest quality, each additional layer provides quality improvements. In [5], we showed that the use of SVC is highly compatible with BPP, and using such an approach can be an effective way to utilize in-network computing.

The main contributions of this paper are as follows: (i) we introduce an end-to-end video streaming platform which shows how to combine various emerging network technologies to provide high precision and to enhance QoE; (ii) we propose a new usage technique for utilising SVC video, and (iii) we determine the minimum number of drones to cover a surveillance area and deploy them for the end-to-end video streaming. The rest of this paper is organized as follows: in section II, we present background work related to video streaming; the details of this study is given section III; a performance evaluation of the proposed system is given in section IV; which is followed by the conclusions in section V.

II. BACKGROUND

A. Video Streaming by Using Drones

Systems that stream video using drone cameras are one of the recent hot topics. Communication between drones was proposed for video transmission in [6], where a system was designed in which users in the network would use drone, cellular or WiFi connection according to the throughput value. The channel rental and energy consumption have been studied in [7] for the video sent from multiple drones to the base station, which then later sent to the clients connected to the base station. The authors in [8] proposed a system based on signaling the drone according to the client's observations such as bandwidth and delay, and adapting the quality of the video captured by the drone camera according to this signal, in the transmission of the video captured by the drone to a client. In [9], different areas in a sport field are captured by drone cameras and transferred to a server for broadcasting. However, in that study, motion sensors were installed on all players and users, and drone cameras moved and captured the video depending on the data they received from the motion sensors. The broadcasting part was not carried out in that study.

The use of drone cameras to stream video can be realized in wide-ranging scenarios, as given in the literature. None of the studies related to this topic considers end-to-end video transmission. In this work, which is different from the literature, the videos captured by the drones are transmitted to the clients, with the utilization of SDN and NFV concepts, while the available bandwidth at the client is considered in order to determine the video quality and in-network computing parameters for packet trimming, to ensure enhanced QoE.

B. Video Streaming with Different Transport Layer Protocols

BPP has been designed as one of the protocols which can be used for low latency / high reliability applications in future networks. It was introduced in 2018 [3], and there are a number of papers related to BPP, which show how it can be used. BPP is a protocol that can provide *packet trimming* capabilities in its design – called Packet Wash in BPP. The use of the BPP Packet Wash process, where chunks in BPP packets are dropped, has been shown to reduce latency [10]. Although BPP was designed for communicating video, AR, and VR, our previous work [5] was the first real implementation of streaming video using BPP, and that work also determined the effects of video transmission using BPP.

RTP is used for multimedia communications over UDP. Its header allows the server to put information such as media timestamps or codec type, but the standard does not cover how to use this information. Another protocol that runs over UDP is QUIC. It implements HTTP to overcome some TCP issues [11]. While its advantages about reliability are reported in many works, there are studies showing it has no advantage or worse performance than TCP for video [12], [13].

In our recent paper [14], we compare video transmission over BPP, with HTTP adaptive streaming, and with TCP. The comparisons showed that BPP adapts the quality in a more efficient way, and BPP is a promising approach that can provide high precision requirements of the future networking applications. In [14], we also proposed an architecture with an SDN controller and a Virtualized Edge Server to provide low latency with high QoE for video streaming applications, based on Packet Trimming and the BPP protocol.

III. END-TO-END VIDEO STREAMING

In this study, we consider an architecture and framework for end-to-end streaming of SVC enabled video from UAVs / drones on the edge drone network, to a set of clients that are connected to an SDN-based edge network. In this system, sequences of video are captured by the drones and transmitted to a virtualized MEC server, where it does not transmit all of the video sequences, it only transmits one of the videos to the clients. The Edge Network that the clients are connected to is managed by an SDN controller. The controller determines the maximum bitrate value for the streamed video and sends this information to the MEC server. This section gives the details of this process and an explanation of the system components. The framework is illustrated in Fig. 1.

A. Management of Drones

At the IoT edge network, the drones capture the video from the surveillance area. The Drone Control Unit is responsible for managing the drones, where it determines: (i) the number of drones to use; (ii) the location of each drone, by taking into account the size of the surveillance area; and (iii) signalling the video parameters to each of the drones which are used for producing scalable encoded video.

In our model, a minimum number of drones needs to be placed in a way to cover the entire surveillance area. We first describe the deployment of multiple drones in a 3D space to cover a given surveillance area (i.e. a square or rectangular area) and manage the drone operations. In this work, we consider the issue of drone coverage as a circle packing problem which is defined as covering a given region with equal circles. We plan to utilize a solution for circle packing as part of contribution (iii). We note that the coverage region of a drone is a circular disk and overlapping areas are included in the problem definition. We assume that each drone is placed at the same height so that they offer an equal amount of coverage, and that the number of drones can vary based on the size of

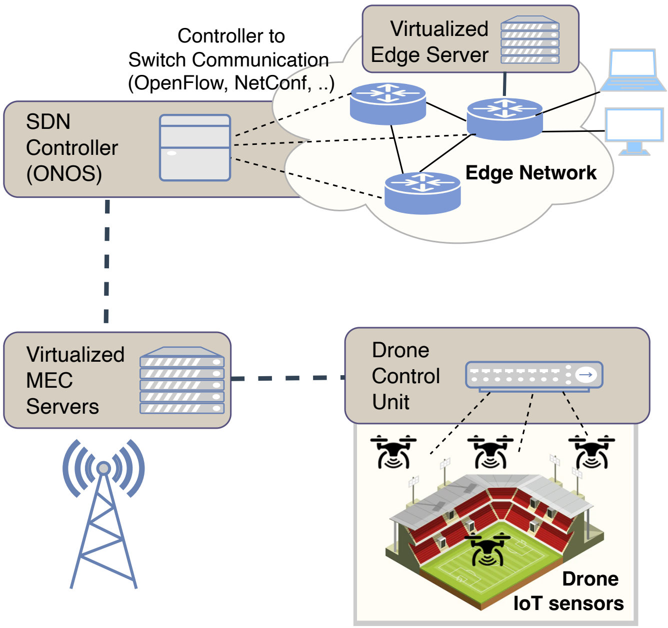


Fig. 1: The Architectural Framework

surveillance area. We use a solution to circle packing presented in [15], where the minimum number of circles is determined according to the size of a rectangular area.

B. The Role of Virtualized MEC Server

The videos that are captured by the drones are transmitted to the MEC server, which resides on the base station as a virtualized function. In our study, the MEC server runs the function to select which video sequence to send over to the clients, chosen from the video streams sent by the drones. The selection of the video sequence is done on the basis of the Region of Interest (RoI). For example, in a basketball or football match, it could be the video sequence that shows the ball. In order to determine which video sequence includes the RoI, the MEC server runs an object search application over the video sequences and signals the Drone Control Unit about the selected video sequence. The application Yolo v3 is used to detect the object of interest because it can detect that object in the order of milliseconds [16].

In this paper, we propose to use SVC to encode the videos from drones, not only is SVC video compatible with BPP, but also it has an amenable coding structure which is different and more dynamic than an unlayered codec, therefore we utilize this. While the selected drone sends the video with the highest quality, other drones send the video with the lowest quality. If the MEC server detects that the RoI has moved to another video stream, then the MEC server signals the newly selected drone to switch to the highest quality, via the Drone Control Unit. At the same time, it signals the original drone to switch down to the lowest quality. Having just one drone send at the highest bitrate, reduces the total video traffic from all of the drones, thereby using less bandwidth and avoiding congestion. This approach of having SVC enabled drones and dynamic video quality selection helps to provide the low latency and high precision needed for video streaming applications.

C. The Role of the SDN Controller

In our framework, the clients are connected to the network which is managed by an SDN controller. We assume that there is a video server which works as a 3rd party application server on the controller. The video server keeps track of the information regarding the client device characteristics, whether it is a mobile phone or TV, via a subscription mechanism.

One role of the SDN controller is to determine the maximum video quality level for each of the clients, based on the available bandwidth. Let the rendering and decoding buffer storage capability of i^{th} client's device be represented as c_i, which is then transformed by the video streaming server into the bitrate unit that corresponds to br_i by using a mapping function. The br_i values determined by each client are used by the SDN controller. The controller has information about the available bandwidth of the paths. It defines the available bandwidth value for the path of i^{th} client, which is represented as abw_i. Hence, $min(br_i, abw_i)$ can be used to define the quality of the video that can be received by i^{th} client. In our system, a set of clients are connected to a first hop router, where there is a virtualized edge server running on it. The virtualized edge servers are managed by the controller, and it determines the maximum video quality level, Q_v, for v^{th} virtualized edge server by using the equation $max_{\forall v}(min(br_i, abw_i))$. The Q_v value is then used for defining the bitrate of the highest layer of the video that will be sent to the clients. That value is signaled to the virtualized MEC server by the controller.

D. Packet Trimming by the Virtualized Edge Functions

In our previous paper [14], we showed the advantages of using both a virtualized edge server and virtualized BPP functions, in order to provide the managed delivery of video to clients, based on packet trimming and the BPP protocol. The experimental results of [14] have shown that providing traffic engineering by implementing in-network quality adaption using packet trimming at the edge, and by using a virtualized BPP function, provides scalability and high quality at the receiver. These results provided a foundation for extending the implementation of the architecture to include video data sources, and drones in particular, in order to evaluate the streaming of SVC video using the full end-to-end resources.

In this work, a virtualized BPP function v trims the video packets regarding the value Q_v, determined by the SDN controller. According to this, the virtualized function trims the chunks by considering their significance values until the packet size is reduced under the limit with respect to the available bandwidth. The determination of the significance values is omitted from this paper due to the space limitation. We refer to our previous work for the details of trimming process [5].

IV. PERFORMANCE EVALUATIONS

In this section, we give the experimental setup and performance evaluation of the end-to-end streaming of the videos captured by the drones.

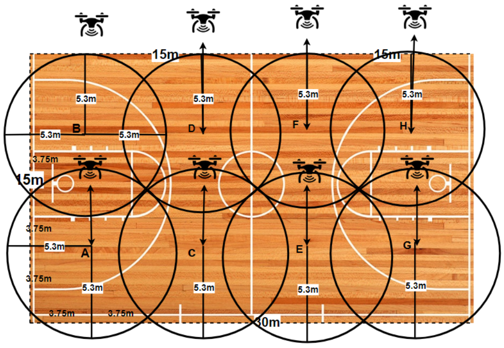


Fig. 2: Positioning & Coverage of Drones on Basketball Court

A. *The Setup of Drone Network and Video Capturing*

Within the scope of the experiments, a sports scenario on a basketball court is simulated on the Unreal Engine 4.25 environment to capture realistic video sequences from the environment. AirSim simulator [17], which is a simulation platform for drones, cars, and more, is used to simulate the drones and it is provided for use with Unreal Engine. We evaluate the effectiveness of the our drone management model and propose solutions for the required number of drones and their locations in the surveillance area so that a video of the sports match can be recorded via stabilized gimbal cameras.

In the experiments, drone deployment is performed for a basketball court. The number of and the coordinates of central points of circles are determined by using the circle packing formula defined in [15]. The drone placement is depicted in Fig. 2. The Drone Control Unit, as the result of this calculation, determines that 8 drones are necessary for video sequences to cover the whole area. 720 frames are recorded during the flight time by each drone and the videos containing are encoded with H.264 SVC. The bitrates of different layers of the videos are 476 Kbps, 864 Kbps, and 2 Mbps for L0, L1, and L2 layers, respectively. The resolution of the video is 480x360. The video is encoded with limited bitrates because MTU size is limited and some data from each layer should be put into the BPP packets to operate efficiently packet trimming process. In these experiments, we assume that there is enough bandwidth to transmit all of the packets reliably.

B. *End To End Video Transmission Experiments*

The experiments were conducted on the Mininet environment for the SDN-based access network. An ONOS controller is used as the SDN controller to manage the SDN domain. The Mininet topology contains a server and 6 clients, connected to OpenFlow switches. The server acts as the virtualized MEC server. There is a virtualized edge server running on the edge switch, which the client are connected to. The bottleneck link between the MEC server and a client is the connection between the client and the edge switch. In some experiments, the bandwidth of this last hop is set to *fixed* values, equal to 0.6 Mbps and 2 Mbps. However, even if the bandwidth equals 2 Mbps, which is the highest video bitrate, the bandwidth required to transmit the video with the highest quality is higher than 2 Mbps due to the overhead of the video packetization. The bandwidth of the links are limited to make the available bandwidth aligned with the bitrates of the videos. In a real-life scenario, videos encoded with higher bitrates and networks connection with higher available bandwidth can be used.

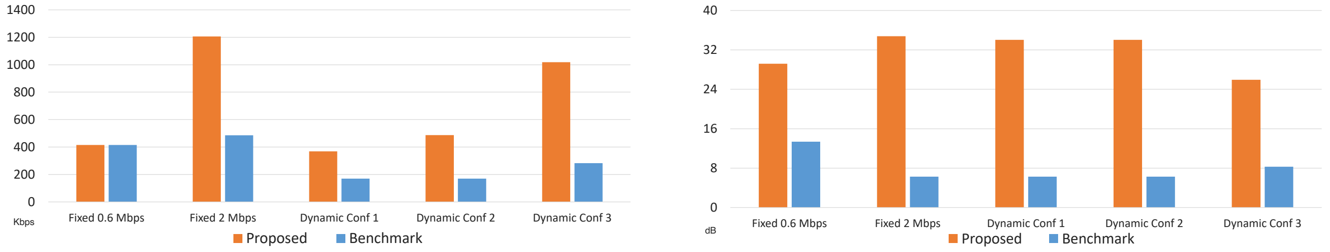
In other experiments, the bandwidth values *dynamically change* due to cross traffic during the streaming session. In the first dynamic bandwidth configuration, which we labeled ***Dynamic Conf 1*** in the graphs, the bandwidth value starts from 0.6 Mbps, it increases to 0.9 Mbps, and then up to 1 Mbps. In the second dynamic bandwidth configuration, labeled ***Dynamic Conf 2***, the bandwidth value starts at 0.7 Mbps, it increases to 2.1 Mbps, and then it decreases to 0.6 Mbps. And in the final dynamic configuration, labeled ***Dynamic Conf 3***, the bandwidth starts with the value of 1.5 Mbps, it decreases to 0.9 Mbps, and then to 0.6 Mbps. In the dynamic tests, the available bandwidth changes every 12 seconds.

On the drone network, the drones are positioned and capture the video with the requested quality based on the commands sent by the Drone Control Unit, as previously described. The MEC server forwards the video towards the clients with the highest quality, but it is possible for the packets to be trimmed by the virtualized edge server during transmission.

We implemented a comparative approach to evaluate the performance. The same Drone Control Unit and drone placement strategy is used in both the proposed approach and the benchmark approach. UDP is used as the transport protocol for the benchmark approach, and BPP is used as the transport protocol for the proposed approach. We selected UDP as the comparison transport layer protocol to provide fair comparison because the clients used in the benchmark and the proposed studies have no capability to adapt quality or to give feedback to the server for retransmission. In the benchmark approach, drones always send the video with the highest quality and thus the bitrate of this quality is not changed on the basis of the network conditions, also, in-network computing and virtualized edge server functionalities were not implemented. The benchmark approach shows the raw performance of a system without using any MEC server functionalities or using a programmable network protocol. The initial waiting time at the client is set to 0.6 sec in order to provide low latency. Therefore, in both approaches, the clients start playing the video 0.6 sec after they get the first packet.

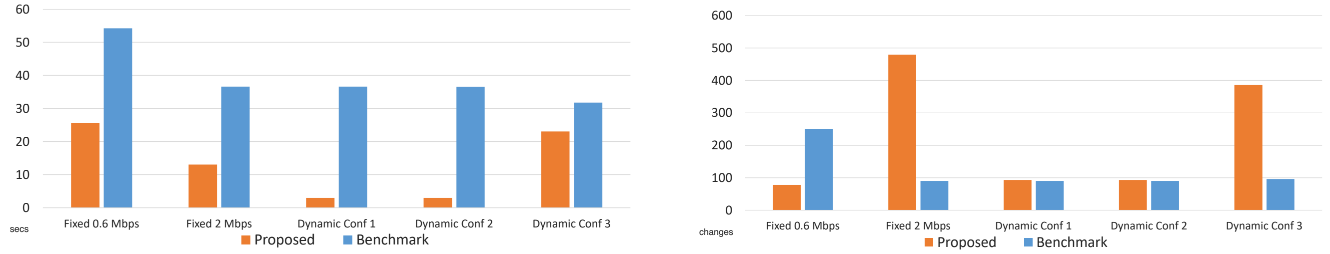
In Fig. 3a, the average received bitrates observed in the clients are given for the proposed and benchmark approaches. For the highly limited bandwidth value, where the bandwidth is 0.6 Mbps, similar results are observed in both approaches in terms of average bitrate. But the clients with the approach using BPP get higher average bitrate for other bandwidth values. We see that even if the bandwidth is high and fixed, equal to 2 Mbps, the client received the video with a low bitrate in the benchmark approach due to the high number of lost packets caused by (i) the lack of quality selection by means of the MEC and (ii) not using in-network computing.

The perceived quality is affected by various factors, not only by the received video bitrate. When we examine other



(a) Average Bitrate Varying Conditions (in Kbps)

(b) PSNR Varying Conditions (in dB)



(c) Duration of Pauses Varying Conditions (in seconds)

(d) Quality Changes Varying Conditions

Fig. 3: Quality of Experience Values across the Experiments

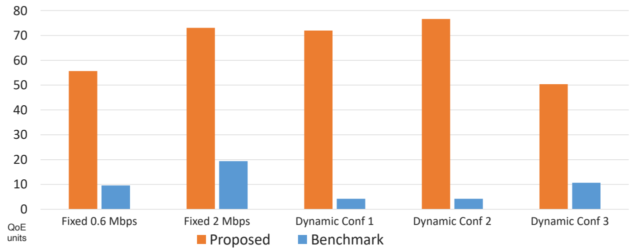


Fig. 4: Overall QoE Results

QoE parameters, we see the results obtained in the proposed approach are higher in terms of PSNR, than the results obtained in the benchmark approach, as shown in Fig. 3b. The low PSNR values observed in the benchmark approach is again due to lost packets. As seen in the graph, there is no direct correlation between the received bitrate and PSNR values. This is because of the frame / layer dependencies in the video, and the varying bitrate of the different parts of the video stream. In addition to that, in Fig. 3c, we see that higher duration of pauses is observed in the benchmark approach than in the proposed approach.

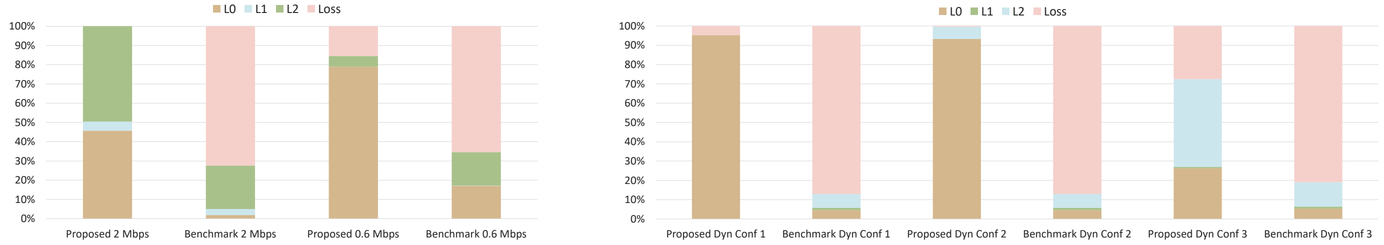
The number of quality changes are presented for both approaches in Fig. 3d. We observe that if the available bandwidth is approximately equal to the bitrate of a particular video layer, then the use of BPP and the in-network computing function in the proposed approach successfully trims the packet carrying the higher layers and provides a seamless transmission. However, if the bandwidth equals to a value that allows different number of layers can be transmitted over time, then we observe higher number of quality changes as we observe in the fixed 2 Mbps bandwidth experiments.

Although various QoE parameters have been presented for the experiments, it is possible to construct an overall QoE value to get a better idea of the overall perceived quality. To calculate the overall QoE value, a linear function proposed in [18] is used. In the formula, the importance of different QoE parameters is indicated by assigning different weights to different terms based on their effects on perceived quality and an overall QoE value is calculated. The averaged overall QoE values for each client in each experiment is given in Fig. 4. As seen from the graph, the performance results show that the proposed system which includes a virtualized edge function, a programmable network protocol, and drone management functionalities provides higher QoE in various network conditions thanks to its cross-layer design capabilities, when compared to the benchmark approach.

The distribution of the received layers shows the played layers on the receiver side, plus the lost frames, and is presented for the *fixed* test in Fig. 5a, and *dynamic bandwidth* test in Fig. 5b. In the graphs for the proposed approach, the distribution of the received layers changes with respect to the available bandwidth, while the distribution is almost the same in all experiments for the benchmark approach. The graphs also show that the number of lost frames is limited with the proposed approach, while there are many lost frames which can be observed with the benchmark approach. These results again show that each client in the proposed approach plays the most suitable video quality with respect to the given network conditions thanks to the quality determination and adaption by the virtualized MEC and edge servers.

V. CONCLUSIONS

In this paper we have presented an architecture that allows drones to stream video from the drone camera, via a number on network nodes using in-network computing, to trim video packets on their way to the clients for end-to-end video streaming. The architecture contains a Drone Control Unit, a



(a) The Distribution of the Received Layers *With Fixed Bandwidth Configuration*

(b) The Distribution of the Received Layers *With Dynamic Bandwidth Configuration*

Fig. 5: Layers Received by the Clients + Loss in the Network

Virtualized MEC Server, an SDN Controller, and Virtualized Edge Servers, in order to provide low latency and low loss.

To evaluate the effectiveness of our proposal, we decomposed the problem into the aspects: (i) the architecture, (ii) the SVC encoded video streaming problem through the drone cameras, and (iii) the 3D drone deployment problem. We evaluated a basketball court scenario where a minimum number of drones were deployed in the surveillance area, using the circle packing approach. Then, sequences of SVC video were captured by the drones and transmitted to a virtualized MEC server for processing. We successfully determined the minimum number of drones to cover the region of interest and deployed them for end-to-end video streaming.

The architecture and the implementation provides software control of the these elements, and as importantly provide a level of integration of the subsystems. This is a different approach to many drone and IoT systems, where they run in isolation, with their own independent management and control. Here we have overcome the problem of separate silos from other systems, and have connected all the elements together. The performance results show that when these different network components are combined in a way to enhance QoE by considering the available network conditions, it provides a good level of improvement in various QoE parameters.

The observations in this study shows that while the management of different emerging network components is promising to meet the future network applications, there is still some room to enhance QoE. In future work, we plan to enhance the QoE by providing fewer quality changes. We will focus on the improvement of the functionality of the Virtualized Edge Function for this purpose. We also plan to consider battery lifetime of the drones and manage the drones so that they provide seamless streaming for a long video sessions, consider other drone scenarios including mobility, and a model to calculate the minimum cover for a surveillance area.

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 (UK DCMS).

REFERENCES

[1] E. Bozkaya, K.-T. Foerster, S. Schmid, and B. Canberk, "Airnet: Energy-aware deployment and scheduling of aerial networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12 252–12 263, 2020.

[2] L. Bertizzolo, T. X. Tran, J. Buczek, B. Balasubramanian, R. Jana, Y. Zhou, and T. Melodia, "Streaming from the air: Enabling drone-sourced video streaming applications on 5g open-ran architectures," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.

[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] 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.

[5] S. Clayman and M. Sayıt, "Low Latency Low Loss Media Delivery Utilizing In-Network Packet Wash," *Journal of Network and Systems Management*, no. 1, p. 29.

[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] C. He, Z. Xie, and C. Tian, "A qoe-oriented uplink allocation for multi-uav video streaming," *Sensors*, vol. 19, no. 15, 2019.

[8] 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.

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

[11] M. Bishop, "Hypertext Transfer Protocol Version 3 (HTTP/3)," Internet Engineering Task Force, Internet-Draft draft-ietf-quic-http-34, Feb. 2021. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-quic-http-34

[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 *Workshop on the Evolution, Performance, and Interoperability of QUIC*. ACM, 2018, p. 43–49.

[14] S. Clayman, , M. Tüker, E. Karakış, and M. Sayıt, "In-Network Video Quality Adaption using Packet Trimming at the Edge," in *26nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, Paris, France, 2023.

[15] A. Heppes and H. Melissen, "Covering a rectangle with equal circles," *Periodica Mathematica Hungarica*, vol. 34, no. 1, pp. 65–81, 1997.

[16] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3," 2018.

[17] (2023, Jan.). [Online]. Available: https://www.microsoft.com/en-us/ai/autonomous-systems-project-airsim?activetab=pivot1%3aprimaryr3

[18] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," in *2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 325–338.