Enhancing Bandwidth Utilization for Video Streaming with Packet Trimming in 6G

Stuart Clayman*, Müge Sayıt† David Griffin*, Miguel Rio*,
*Dept. of Electronic and Electrical Engineering, University College London, London, UK
†Dept. of Comp. Sci. and Elec. Eng., University of Essex, Colchester, UK – Ege University, Izmir, Turkey

***Abstract*—The utilization of the packet trimming technique as part of 6G has previously been described. It allows a network node to reduce packet sizes, by removing some of the content, during the journey of each packet. This approach has been successful in many ways, but can be limited when there is very little bandwidth. In this paper, we propose an approach using an architecture suitable for 6G, which overcomes some of these issues by removing extra content from the packets, when network conditions are restricted.**

***Index Terms*—6G, In-Network Packet Trimming, Bandwidth Utilization, 3GPP, Video Streaming, Scalable Video Coding**

I. Introduction

Recent advancements in Next Generation Networks (NGNs) have accelerated rapidly in recent years, triggering researchers to explore new ideas to enhance the performance of applications running on these networks. Among these, multimedia applications such as Augmented/Virtual Reality (AR/VR), interactive video streaming, immersive gaming are expected to deliver low-latency communication.

Although the enhanced capabilities of Beyond 5G/6G networks provide a robust infrastructure for low latency and high throughput, the performance of multimedia applications can be further optimized by integrating new transport mechanisms that consider the specific characteristics of NGNs. Among these, edge computing is a technology that offers data processing, storage, and reduced latency due to its proximity to clients [1]. Furthermore, 3GPP proposed a new reference architecture recently for media streaming, in which edge network components for data collection and analysis are introduced [2].

As one of the recent approaches for better utilisation of the network conditions, in-network packet trimming is a promising technique that can be advantageous for video streaming applications. In our previous work, we showed how this technology can be utilised, in the wired networks [3], [4]. More specifically, with in-network packet trimming, video packets sent by the server can be modified in-transit by removing some data from the packet in order to manage network bandwidth throttling during transmission to the destination. In-network packet trimming effectively reduces packet loss by shrinking packets instead of dropping them during network congestion. This approach not only minimizes packet loss but also significantly decreases the need for retransmissions.

Although modifying packets during transmission might seem straightforward, it involves numerous challenges. These include deciding which data to remove, measuring the current bandwidth, and ensuring that the decision for each packet aligns with the available capacity over a longer time period than a single packet's transfer duration. Big Packet Protocol (BPP) provides the mechanism to signal the importance of these chunks to the network, via a significance value, as well as a threshold saying which chunks should never be removed [5]. Previously we demonstrated how this feature can be utilized for video streaming by defining the significance of the chunks within a packet based on the importance of video frames [4].

In [6], we provided some initial results for the in-network packet trimming at the edge of 6G wireless networks and showed the necessity of new approaches to cope with the dynamism of the NGNs to provide better utilisation of the network conditions. In 5G and 6G networks, there are scenarios where the bandwidth is severely limited, and where the initial trimming may not sufficiently reduce the data size. This presents two options: (i) dropping packets, which although normal, is undesirable, or (ii) performing additional trimming, which is preferable. The challenge lies in how to implement further trimming effectively.

In our previous work the significance value assignments [3] and the threshold were static. However, due to the dynamic nature of network conditions of wireless networks, more dynamic approaches are required, when using video streaming applications. In this paper, a proposed solution is to relax the predefined threshold for trimming, and allow the trimming operation to be more dynamic during restricted conditions at the edge. For this purpose, we introduce a new signaling mechanism by updating the fields of the packets at the sender, and enhance the architecture.

The contributions of this paper can be listed as: (i) we utilize the fields of BPP packets so that the sender signals the edge network functions for them to adapt quality considering the different network conditions; (ii) we propose an architecture for collecting observed bandwidth values from the clients and reacting based on this information, which is compatible with 3GPP reference architecture; (iii) we demonstrate and evaluate the performance potential of packet-trimming using results obtained from a real wireless network dataset.

II. Background

1) Edge Computing for Video Streaming in Beyond 5G/6G: Edge Computing, often referred to as Multi-access Edge Computing (MEC) in Mobile Networks, involves shifting core network functions and certain cloud computing resources

978-3-903176-66-9

from the central core to the network's edge. This relocation brings the functionalities closer to the user equipment (UEs), thereby minimizing the communication distance. The primary benefit of this approach is the significant reduction in latency, alongside the potential for optimizing data flows through local processing and storage capabilities. In the context of 3GPP, the emphasis on edge computing is directed towards defining MEC for 5G/6G infrastructures, enabling the provision of service environments and cloud computing capabilities at the network edge [1]. As edge facilities bring resource closer to the users, there are many studies proposed in the literature that address the utilization of edge computing for video streaming services.

Edge computing is used for various purposes, among them edge caching being one most attractive research areas due to its advantages on providing low latency [7]. Utilising caching at the edge can provide improvement in QoE [8], [9]. The studies of HTTP Adaptive Streaming Systems this is addressed due to the importance of caches in these systems [8], [10].

While edge computing can be utilized to provide low latency and storage, edge analytics functions can be employed to analyze data collected from network functions and user equipment (UEs) [2]. The reference architecture proposed by 3GPP for the 5G Media Streaming (5GMS) domain includes a data collection application function, which receives data from UEs. Additionally, the 5GMS Application Provider Event Consumer application function allows for further analysis of this data by the provider. Most studies on edge analytics in the literature focus on video analytics for resource provision [11] or AI services [12] for object detection and security. In contrast to these approaches, our study leverages edge analytics capabilities in NGNs to collect bandwidth information for managing packet trimming.

In our recent work, we showed the advantages of implementing packet trimming at edge for Beyond 5G/6G systems [6]. In that paper, we focused on different bandwidth evaluation algorithms during packet trimming. As follow-on work of [6] here, we show an architecture for collecting bandwidth information plus an enhancement of the BPP protocol to cope with the dynamic nature of wireless cellular networks.

2) Low Latency Low Loss Video Streaming: Providing low latency in emerging video streaming applications is one of the most important research topics recently. For this purpose, there have been active studies for both Low-Latency DASH (LL-DASH) and QUIC. A recent study shows while QUIC can reduce the latency significantly compared to LL-DASH, it suffers from video stalls [13]. To ensure a seamless video streaming experience with minimal latency and packet loss, the authors in [14] address the use of Low Latency Low Loss Scalable Throughput (L4S) technology over 5G networks. This technology leverages Explicit Congestion Notification (ECN) signaling, which alerts the sender when the network is congested, thereby maintaining low latency.

Several other studies aim to achieve low latency and high QoE for video streaming applications. However, these studies typically do not incorporate emerging transport layer mechanisms or real-time congestion elimination methods. Our approach differs to these by employing in-network packet trimming in order to prevent or minimize congestion, thereby ensuring low latency and low packet loss.

3) Video Streaming with Scalable Video: Scalable Video Coding (SVC), also known as Layered Video Coding, encodes video in multiple layers. This approach is advantageous because it leverages the similarities between different versions of the same frame and can efficiently reassemble successive frames. SVC encoding includes various quality layers, such as temporal and spatial quality layers [15]. SVC has been defined for standardized video codecs, including H264, H265, and recently H266. WebRTC also has some SVC functionality, as the the AV1 codes now supports scalable video.

Our previous work [4] provides a detailed overview of the mechanisms and techniques for transmitting H264 SVC video over the network. This includes the use of multi-chunk BPP packets, video stream multiplexing facilities at the sender, the packet trimming capabilities of BPP in the network, and stream reconstruction functions in the client.

III. Packet Trimming at the Edge

1) Architecture for Packet Trimming at the Edge: In our system, we utilize the 3GPP functionality for a 6G architecture at the edge, outlined in [1]. Edge Computing Service Providers (ECSP) are responsible for the deployment of Edge Data Networks (EDN). Application Service Providers (ASP) are responsible for the creation of Edge Application Server (EAS) resident in an Edge Data Network, performing the server functions. We use two virtualised functions running as Edge Application Servers (EAS) provided by a media delivery company, acting as an Edge Cloud Service Provider (ECSP). For our application, the virtualised elements run the *Trimming Function* and a *Collector Function*. These functions are accessed via traffic steering at an I-UPF.

The *Collector Function* is responsible for collecting observed bandwidth values from the user equipment (UE) / clients. 3GPP defines the Direct Data Collection Client, embedded at the client, and the Data Collection AF, in [2]. These elements gather client data, which can be sent as Events to an Application Service Provider for immediate use, or sent to an NWDAF. By considering the functionality of the *Collector Function*, it is clear that this component of the system fits into the proposed 3GPP architecture.

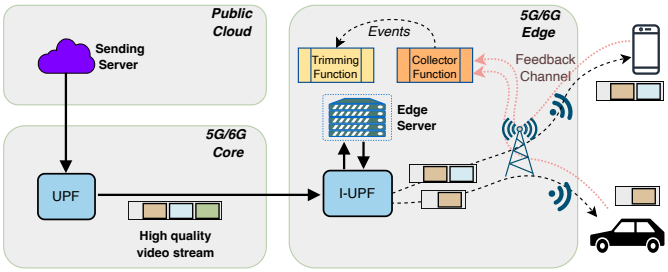


Fig. 1: Packet Trimming functionality in a 6G Architecture

An illustration of the proposed system is given in Fig. 1. Video data is sent at the highest quality from a server, via a UPF in the 5G/6G core, to an I-UPF at the edge. The I-UPF uses the Uplink Classifier and Branching Point mechanisms of 5G, to implement the traffic steering into the virtualised *Trimming Function*. Trimming of packets is done if the current bandwidth is limited, and packets, trimmed or not, are forwarded to the client. In our system the data which is collected at the client is the observed network bandwidth values, measured at 1 second intervals. The *Collector Function* receives the bandwidth data and then sends this information directly to the *Trimming Function* as Events, as per the definition in the 3GPP reference architecture [2].

The *Trimming Function* trims packets based on the video data within each packet, plus meta-data in the header of BPP packets. The header contains fields that allow the trimming network function to determine the importance of data chunks, and which parts can be removed during limited bandwidth periods. As part of the video streaming company, the sender sets the meta-data values based on the video file. In the following section, we define these fields and explain in detail how the *Trimming Function* operates based on them.

2) Packet Trimming with Threshold Relaxation: With BPP transmission, the video server constructs packets where the payload carries video chunks, and the packet header contains significance values and a threshold. The significance values indicate the importance of the video chunks, while the threshold signals to the trimming node which chunks can be trimmed during bandwidth restricted periods. The network node trims chunks whose significance values *is greater than* the threshold. The client reconstructs a video stream.

In our previous work, we utilized the BPP header fields to adapt video quality by utilising those fields [4]. In this work, we enhance the fields by updating the *function* field in the packets. This field defined in BPP allows network nodes to adaptively act in response to changing network conditions. Considering the dynamic nature of cellular wireless network capacity, we propose a use of the *function* field which passes on a relaxation value. Specifically, we inject a value in each packet, which can trigger a process at the network node to *reduce* the threshold value for that packet. This enables the *Trimming Function* to trim more data than would be possible with the fixed threshold value. Such a mechanism is used when the bandwidth is severely limited. After performing the first trimming operation, it checks the amount of data that has been trimmed. If needed, the threshold value is *relaxed* when the bandwidth is limited, and trimming is done for a second time.

The operation performed by the *Trimming Function* is detailed in Algorithm 1.

Algorithm 1: Algorithm for relaxing the threshold

```
▷ Calculate how far below the bandwidth to the client we are
below ← calculateBelow()
                                   ▷ Calculate amount we want to trim
trimLevel ← calculateTrimAmount(below)
                                   ▷ Check if trim is needed
if (trimLevel > 0) then
    // Need to trim
    threshold ← packet.threshold
    trimmedAmount ← trimContent(trimLevel)
                                   ▷ Check if we didn't trim enough
    if (trimmedAmount < packetTrimLevel && relaxOn)
     then
        // Fetch relax amount from packet
        relax ← packet.relaxValue
        // Apply relax amount to threshold
        threshold −= relax
        // Try and trim some more
        newTrimLevel ← trimLevel−trimmedAmount
        // Call trimContent() again
        nextTrimmed ← trimContent(newTrimLevel)
        if (nextTrimmed > 0) then
            // Some more was trimmed
        else
            // No more was trimmed
    else
else
    // Nothing to trim
```

IV. Performance Evaluations

In order to evaluate the performance of the sender setting a relaxation value, and triggering the network node to relax/reduce the threshold, we conducted tests. We compared using the fixed threshold approach, as in previous work, and then tested by using different relax values. We observed the difference in data forwarded from the network node.

1) Video and Experimental Parameters: In the experiments, we utilized a real dataset to evaluate the performance of our proposal over wireless cellular networks. For this purpose, we employed actual bandwidth traces collected by Ghent University for the bandwidth link to the UE/client [16]. Specifically, we used the bus dataset, which is a scenario of a person traveling on a bus, and displays highly dynamic bandwidth.

For our experiments we utilized the widely known Big Buck Bunny video, encoded with H264 SVC. This video includes one base layer and two enhancement layers. The highest quality, achieved with all enhancement layers, has a bitrate of approximately 3650 Kbps. The sender continuously transmits data near this bitrate, as there is no feedback mechanism to inform the sender of the client's varying bandwidth.

In the SVC encoded video, there are two distinct types of frames: I frames and P frames. Each frame contains both base and enhancement layers, which contribute to the video quality in terms of PSNR. Due to the Group of Picture (GoP) structure, P frames also include temporal layers that affect the playback rate, or the number of frames displayed per second.

Table I details the frame types, the number of frames, the byte size for each frame type across different quality and temporal layers, plus the significance values for those, which is embedded into the packets. In this table, the temporal layers

T1 to T4 correspond to frame rates ranging from 3.75 fps to 30 fps. If the client receives all enhancement and temporal layers within one second, it plays the video at the highest quality and full 30 fps. The frame types are listed in the table according to their importance and impact on perceived quality.

Frame Type	Frames	Bytes	Significance
I frame	625	4462085	1
I frame enhanced	1250	10353577	2, 3
P frame T1	625	3398816	2
P frame T1 enhanced	1250	8590333	7, 12
P frame T2 & T3	3750	18044105	3, 4
P frame T2 & T3 enhanced	7500	45150454	8, 9, 13, 14
P frame T4	5000	17625759	5
P frame T4 enhance	10000	47732927	10, 15
TOTAL	30000	155358056	

TABLE I: Original H264 SVC video info

2) Performance Evaluations on Transferred Data and Bandwidth Utilization: To demonstrate the advantages of using the relaxation value, we measured the forwarded data rate during video streaming sessions with relaxation, and compared these against streaming using a fixed threshold value of 5. We show the performance results and the improvement when the fixed threshold value is set to 5, compared to UDP transmission and HAS in our previous work [3]. It highlights how further improvement can be achieved using relaxation and allowing a second trim process, as described in Algorithm 1.

The graphs in Fig. 2, Fig. 3, Fig. 4, and Fig. 5 show the results for a fixed threshold value and relaxation values of 1, 2, and 3, respectively. In the graphs, the purple line represents the available bandwidth at the client. The sender rate is shown by the green line. The forwarded data rate, is depicted by the blue line, representing the data rate after the *Trimming Function* has performed its operation. If the value represented by the blue line exceeds the purple line, it indicates that more data than the network can handle is being transmitted. The graphs illustrate that when using a fixed threshold value, more data than the available bandwidth is transferred in many instances.

Forwarding more bytes than the available bandwidth *stresses* the link, causing congestion, latency, and packet losses. Although the graphs show the performance of different approaches, in order to demonstrate how much improvement is provided with the relaxation approach more clearly, we measured whether the forwarded data rate less than and more than the available bandwidth for each approach. These

	Fixed	Relax 1	Relax 2	Relax 3
Forwarded too much - stress	-3330053	-1353877	-1943733	-853982
Forwarded less than avail - ok	3091773	2675033	2853616	2920310
Improvement on reducing stress	-	59%	42%	74%
Improvement on network utilization	-	14%	8%	6%

TABLE II: Matching numbers to bandwidth (in Bytes)

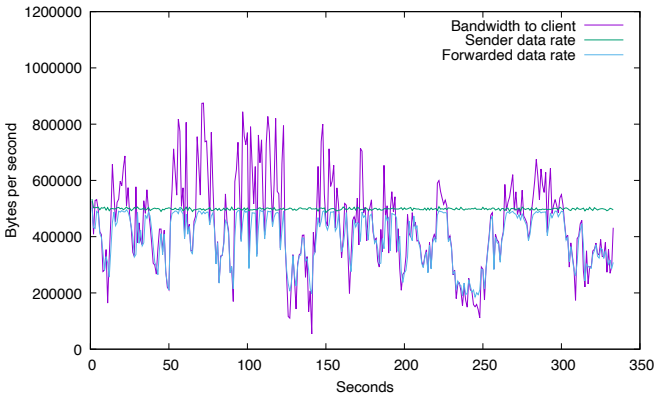


Fig. 2: Flow rates with fixed Threshold: 5

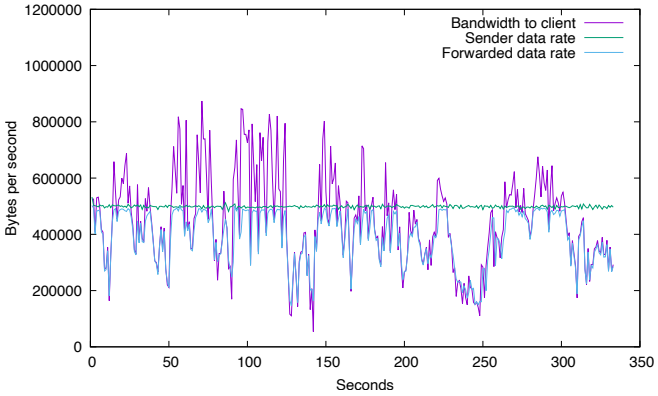


Fig. 3: Flow rates with Threshold Relaxation of 1

measurements are provided in Table II. In the table, the number of forwarded bytes more than and less than the available bandwidth are shown for fixed threshold and the three relaxation values. Our results indicate that using relaxation can reduce this network stress by up to 74%. Conversely, forwarding fewer bytes than the available bandwidth reduces network utilization. Comparing the approaches, the best performance in terms of network utilization is achieved by setting the relaxation value to 1, which provides a 14% increase in network utilization compared to the fixed threshold value.

To assess the improvement provided by using a relaxing threshold value, a detailed view around the 250^{th} second is shown for both the fixed threshold and the approach with a relaxation value of 2 in Fig. 6. The figure reveals that while the fixed threshold value results in more bytes being forwarded than the network can accommodate, the forwarded data rate using relaxation *closely matches the available bandwidth*.

The results demonstrate that relaxation, with a value in a packet and in-network functionality, is an effective mechanism for coping with the dynamism of wireless cellular networks. Considering the promising success of the fixed threshold approach over UDP and TCP previously seen in [3], we believe that utilization the relaxation technique will help achieve higher QoE by providing lower loss and lower latency.

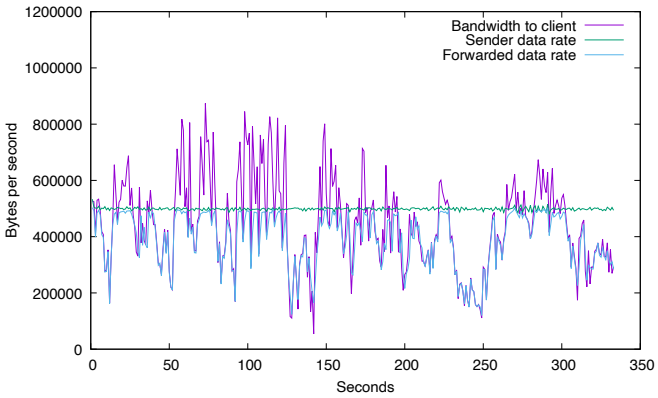


Fig. 4: Flow rates with Threshold Relaxation of 2

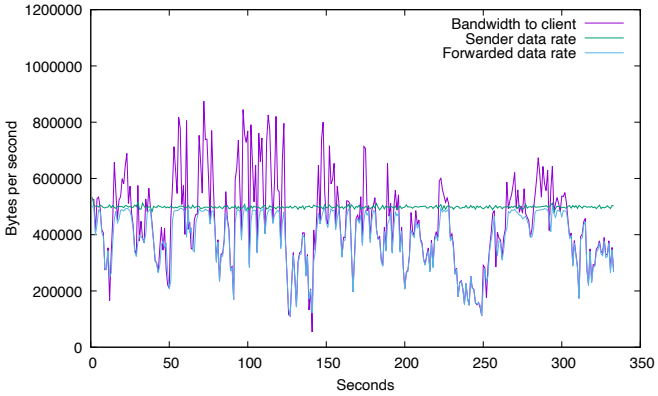


Fig. 5: Flow rates with Threshold Relaxation of 3

V. Conclusions

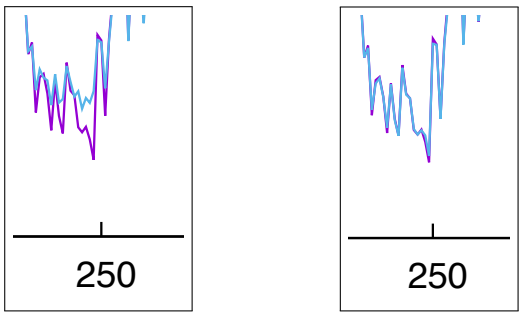
The packet trimming technique, as part of 6G, has been successful in many ways, but can be limited when there is very little bandwidth. We proposed an approach here, which overcomes some of these issues by removing extra chunks from the packets, when network conditions are restricted.

In this work we have shown how the combination of a sender, a network node, and a client can participate in video streaming. By having a suitable protocol where we can inject a meta-data value in each packet, which can trigger a mechanism at the network node, we have shown how dynamic behaviour can be activated. This approach enables a *Trimming Function* at the edge, to trim more data than would be possible with the fixed header values. Such a mechanism was used when the bandwidth was severely limited, and allowed the network node to attempt a second round of trimming on the packet.

For future work we plan to consider further techniques for matching the video streams to the available bandwidth. Also an investigation into the overhead of doing a second round of trimming will be undertaken.

Acknowledgment

Stuart Clayman and David Griffin are funded by the TUDOR project (Dept. for Science, Innovation and Technology under the Future Open Networks Research Challenge).



(a) Fixed threshold (b) Relax 2

Fig. 6: Zoom in at 250 seconds

References

[1] J. Chou, "Edge computing management and orchestration," 2023. [Online]. Available: https://www.3gpp.org/technologies/edge-mao-2

[2] R. Bradbury, "UE Data Collection, Reporting and Exposure," 2022. [Online]. Available: https://www.3gpp.org/technologies/ue-data-sa4

[3] 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.

[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.

[5] 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.

[6] S. Clayman, M. Sayıt, D. Griffin, and M. Rio, "Using Edge-Based Packet Trimming for Effective Bandwidth Utilization in 6G," in *3rd International Conf. on 6G Networking (6GNet)*, Paris, France, 2024.

[7] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen, and W. Zhu, "Understanding performance of edge content caching for mobile video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1076–1089, 2017.

[8] S. Bayhan, S. Maghsudi, and A. Zubow, "EdgeDASH: Exploiting network-assisted adaptive video streaming for edge caching," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1732–1745, 2020.

[9] C. Li, L. Toni, J. Zou, H. Xiong, and P. Frossard, "Qoe-driven mobile edge caching placement for adaptive video streaming," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 965–984, 2017.

[10] A.-T. Tran, N.-N. Dao, and S. Cho, "Bitrate adaptation for video streaming services in edge caching systems," *IEEE Access*, vol. 8, pp. 135 844–135 852, 2020.

[11] D. Xu, A. Zhou, G. Wang, H. Zhang, X. Li, J. Pei, and H. Ma, "Tutti: coupling 5g ran and mobile edge computing for latency-critical video analytics," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022, pp. 729–742.

[12] P. Basaras, E. Vasilopoulos, S. Magklaris, K. V. Katsaros, and A. J. Amditis, "Experimentally Assessing Deployment Tradeoffs for AI-enabled Video Analytics Services in the 5G Compute Continuum," in *2023 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, 2023, pp. 99–104.

[13] Z. Gurel, T. E. Civelek, D. Ugur, Y. K. Erinc, and A. C. Begen, "Media-over-QUIC Transport vs. Low-Latency DASH: a Deathmatch Testbed," in *Proceedings of the 15th ACM Multimedia Systems Conference*, ser. MMSys '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 448–452.

[14] D. Brunello, I. Johansson S, M. Ozger, and C. Cavdar, "Low latency low loss scalable throughput in 5g networks," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, 2021, pp. 1–7.

[15] 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.

[16] Ghent University, "4G/LTE Bandwidth Logs," https://users.ugent.be/~jvdrhoof/dataset-4g/.