# An Edge and Fog Computing Platform for Effective Deployment of 360 Video Applications

Giovanni Rigazzi*, Jani-Pekka Kainulainen†, Charles Turyagyenda†, Alain Mourad†, Jaehyun Ahn‡
*InterDigital Germany GmbH, Berlin, Germany
†InterDigital Europe, London, UK
‡InterDigital Asia, Seoul, South Korea
E-mail: {name.surname}@interdigital.com

***Abstract*—Immersive video applications based on 360 video streaming require high-bandwidth, high-reliability and low-latency 5G connectivity but also flexible, low-latency and cost-effective computing deployment. This paper proposes a novel solution for decomposing and distributing the end-to-end 360 video streaming service across three computing tiers, namely cloud, edge and constrained fog, in order of proximity to the end user client. The streaming service is aided with an adaptive viewport technique. The proposed solution is based on the H2020 5G-CORAL system architecture using micro-services-based design and a unified orchestration and control across all three tiers based on Fog05. Performance evaluation of the proposed solution shows noticeable reduction in bandwidth consumption, energy consumption, and deployment costs, as compared to a solution where the streaming service is all delivered out of one computing location such as the Cloud.**

***Index Terms*—eMBB, 360 video, fog computing, edge computing.**

## I. Introduction

Augmented Reality (AR), Virtual Reality (VR) and Mixed Reality (MR) have become prominent technologies within the wide spectrum of video applications available in different markets such as cinema, gaming, education, healthcare, sports and advertisement [1]. While VR offers an immersive virtual user experience and AR augments a real or virtual environment by adding elements for interaction with the user, MR provides a reality-virtuality continuum consisting of different combinations and variations of real and virtual objects co-existing in the same environment. For instance, a 360 video streaming service can be thought of as an MR application, since users can panoramically watch a real video scene by seamlessly adapting the Field of View (FoV) or viewport, i.e., the fraction of omnidirectional view of the scene, as the viewing orientation changes.

In the context of Third Generation Partnership Project (3GPP) 5G system specifications, such 360 video streaming service would fit in the enhanced Mobile Broad-Band (eMBB) service class, due to the significantly high-bandwidth and constant-data-rate requirements [2]. For instance, assuming High Efficiency Video Coding (HEVC) compression, a live video service with 60 frames-per-second and 8K resolution requires 361 Mbps in order to ensure smooth content play [1]. In addition, a stringent latency requirement is dictated by the need to prevent motion sickness, which is originated from the variation of the video rendering latency after a FoV change. Finally, reliability is another crucial requirement as Quality of Experience (QoE) can be highly affected by video frames not correctly delivered or not meeting the associated deadline. As a result, such 360 video streaming service also falls in the 3GPP Ultra-Reliable Low-Latency Communication (URLLC) class of services.

Although, it is anticipated that 3GPP 5G New Radio (NR) will support such bandwidth-intensive and delay-sensitive applications by providing data rates between 1 and 20 Gbps as well as round-trip delays of few milliseconds, conventional video streaming solutions mainly rely on the huge processing power supplied by cloud data centers, resulting in high-bandwidth consumption on the backhaul, high latency and limited-deployment flexibility. By contrast, novel computing solutions, including edge and fog computing, can ensure low-latency and cost-effective deployment of end-to-end 360 video streaming service. As an example, efficient and flexible mechanisms to offload resource-demanding computing tasks, such as rendering and video processing, will play a key role, especially for battery-powered terminal clients. Different computing tasks may be offloaded from a remote cloud data center to edge servers located in the network infrastructure or onto fog clients in the proximity of the end users, depending on the amount of resources and their availability. The H2020 5G-CORAL project has been developing a solution that integrates distributed virtualized computing at the edge and in the fog including on terminal devices, with the aim to support a flexible and cost-effective deployment of several low-latency applications including 360 video streaming [3].

In this paper, we propose a novel solution to effectively deploy a 360 video streaming service using the distributed edge and fog computing platform developed in 5G-CORAL. We consider a popular viewport-adaptation technique based on adaptive tile-encoding streaming, where the 360 video is partitioned in small tiles, which are independently encoded and transmitted according to the viewing orientation, and then stitched together to recompose the 360 frame [4]. To compensate for the extra computational complexity needed to continuously adapt the video stream quality, we spread the computing tasks across three different tiers, namely, fog, edge and cloud, according to their respective computational and latency requirements. The papers novel contributions can be

978-1-7281-0922-0/19/$31.00 ©2019 IEEE

outlined as follows:

1) Different computing processes, including DASH coding/decoding and video frame composition, are distributed across three different tiers, i.e., cloud, edge and fog, thus increasing system scalability and reducing the latency.
2) Specific GPU-intensive tasks are offloaded from the client. This results in a reduced terminal complexity as well as improved interoperability, as some of the protocols and video codecs employed may not be supported by legacy devices.
3) A novel fog computing framework, i.e., Fog05, is adopted, which enables management, monitoring and orchestration of diverse resources spanning across the three computing tiers, thus facilitating the deployment, operation and lifecycle maintenance of the 360 video streaming service.

Numerical results show that our approach can alleviate the footprint of the 360 video delivery service on a cloud data center by reducing the GPU load, the consumed power and the memory usage. Furthermore, we evaluate the fronthaul and backhaul bandwidth needed to deliver the video content and compare the adaptive tile-encoding approach with two non-optimized solutions, where all the tiles are encoded at low or high quality. In particular, we observe a bandwidth reduction equal to 33% and 5%, respectively in the backhaul and fronthaul, when adaptive tile-encoding is employed with respect to the full high-quality approach.

The rest of the paper is organized as follows: Section II describes the state of the art, while Section III presents the proposed solution based on the 5G-CORAL system architecture. Section IV illustrates the system setup and the evaluation methodology, while Section V presents and discusses the numerical results obtained. In Section VI conclusions are drawn and next steps outlined.

## II. Related Work

Over the past few years, significant research efforts have been devoted to address the bandwidth waste issue associated with the 360 video streaming. This is due to the fact that the user typically watches a single FoV at a time, which represents a small portion of the omnidirectional view fetched from the server. Nevertheless, the stringent latency constraint necessary to cope with motion sickness prevents from adopting traditional adaptive rate schemes based on the tight interaction between the client, which periodically reports the FoV being watched, and the server, in charge of delivering only the requested FoV.

Authors in [5] devise an adaptive bandwidth-efficient 360 VR video streaming system based on a divide-and-conquer approach. The 360 raw video is first spatially partitioned into multiple tiles, which are linked together based on the MPEG Dynamic Adaptive Streaming over HTTP (DASH) Spatial Representation Description (SRD) standard. Next, the underlying 3D geometry is divided into segments and a mapping between tiles and segments is applied. A viewport-adaptive

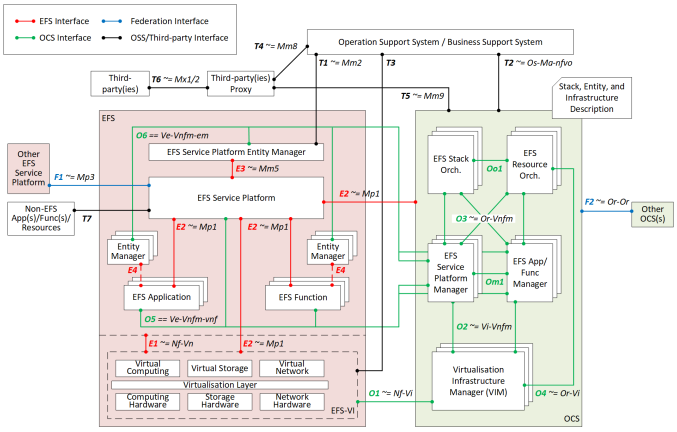


Fig. 1. Overview of the 5G-CORAL architecture.

streaming solution is proposed in [6]. Rather than varying the streaming data rate, this approach considers the Quality Emphasized Region (QER), which corresponds to the highest quality video region. An adaptive algorithm is then executed by the client in order to select a video representation fitting the available bandwidth, whose QER is close to the viewport.

Authors in [7] advocate the adoption of mmWave communications along with mobile edge computing and proactive caching mechanisms in order to meet strict latency and reliability requirements, while multi-path multi-tier 360 video streaming solutions for 5G wireless networks are designed and evaluated in [8], by analyzing the optimal trade-off between video quality enhancement and cost for using the 5G channel. In [9], the Scalable extension of the High-Efficiency Video Coding (HEVC) standard is employed to develop a Tile-Based video streaming solution over mmWave networks together with an offloading mechanism to assist mobile terminals in decoding the viewport. On the other hand, the idea of predicting users head movements in order for the user to download only the FoV is discussed in [10], yet prediction errors are inevitably introduced.

## III. Proposed Solution

In this section, we describe the proposed solution based on the 5G-CORAL system architecture. First, we introduce the 5G-CORAL framework and its major components. Also, we illustrate the Fog05 computing platform and list the main features available. We then examine each function needed to support the end-to-end 360 video streaming service and discuss how they are mapped onto the 5G-CORAL reference platform.

### A. The 5G-CORAL Framework

The main objective of the 5G-CORAL[1] project is to devise a unified platform architecture in line with the ETSI Multi-Access Edge Computing (MEC) [11] and the ETSI Network Function Virtualization (NFV) framework [12], capable of

[1]Project webpage: http://5g-coral.eu/

integrating and federating resources belonging to edge and fog computing systems. Fig. 1 illustrates the overall architecture consisting of two key building blocks, i.e., the Edge and Fog computing System (EFS) and the Orchestration and Control System (OCS). The former embraces and hosts a collection of virtualized and bare-metal applications, services and functions, which are made available for any requesting entity, whereas the latter supervises the EFS by managing the lifecycle of each EFS entity and deals with interconnecting the platform with external EFSs, third-party entities or even other OCSs. Furthermore, the OCS includes the EFS Stack Orchestrator module which builds the end-to-end service by instructing dedicated modules to instantiate the appropriate functions and applications. The OCS is also connected to the Operations Support Systems and Business Support Systems (OSS/BSS) and exchanges essential information for accounting, billing and system performance monitoring. 5G-CORAL also aims to ease coordination between multiple RATs by leveraging context information stored in the EFS.

### B. Fog computing platform: Fog05

Fog05 is an open-source fog computing platform providing a scalable infrastructure that allows to distribute computing, storage, control and networking functions near the users [13]. This is achieved by implementing abstractions that connect compute, storage and communication fabric end-to-end, and facilitate management, monitor and orchestration of applications across a cloud-to-thing continuum. An abstraction consists of a Fog05 entity, which can be atomic, such as a virtual machine, a container or a binary executable, or can be a direct acyclic graph (DAG) of entities. Fig. 2 shows the high-level architecture of Fog05. It can be run as a process on a traditional OS or as a trusted service inside a Trusted Execution Environment (TEE). Key component of Fog05 is the agent or node, which is the manageable system resource. Multiple Fog05 agents are capable of discovering each other through the distributed store, leveraging the dynamic discovery functionality provided by different Pub/Sub protocols, e.g., Data Distribution Service (DDS) or Zenoh. Finally, the agent supports a number of features based on the existing plugins in charge of managing an entity. For instance, an OS plugin can offer all the primitives necessary for Fog05 to manage and operate an OS. Due to its flexibility and ability to handle heterogenous resources, Fog05 represents an ideal fog computing platform to enable dynamic offloading and efficient resource distribution in a multi-tier architecture.

### C. 360 viewport adaptive streaming over 5G-CORAL

As shown in Fig. 3, our end-to-end 360 video streaming service consists of five separate entities, namely, video source, EFS, OCS, OSS and User Equipment (UE). In the following, we describe their role and the information exchanged between the interfaces.

*1) Video source:* one or more 360 cameras are meant to capture the scene and dispatch the live video content to a cloud data center, which represents the EFS entry point.

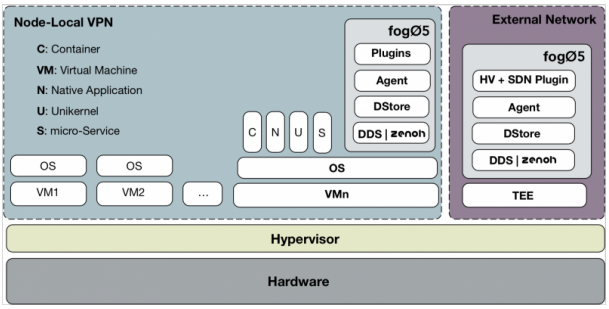


Fig. 2. Fog05 architecture.

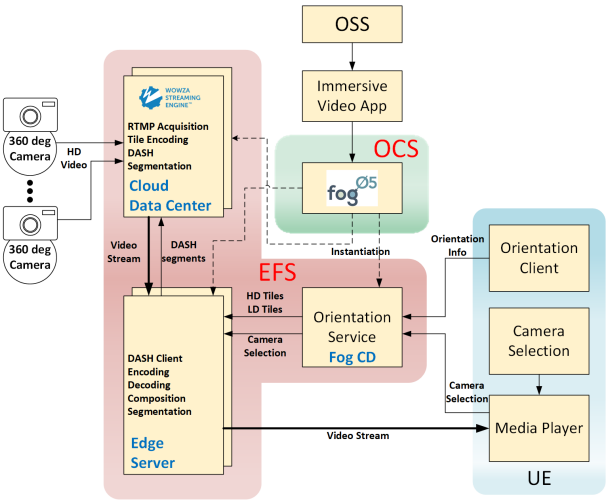


Fig. 3. End-to-end streaming service and related building blocks.

This is achieved by connecting the cameras with a streaming engine, e.g., Wowza Streaming Engine[2], and establishing a Real Time Multimedia Protocol (RTMP) live stream session, which ensures a TCP-based persistent connection and low-latency communication.

*2) EFS:* built upon three different tiers, i.e., *cloud data center*, hosting powerful processing units and located on cloud provider premises, *edge server*, located closer to the user equipment (UE) and providing limited capabilities, and *fog computing devices* (CD), resource-constrained devices operating in the UE proximity, the EFS offers all the essential functions and services to deliver the live video stream. After RTMP acquisition, the data center performs tile-based High-Efficiency Video Coding (HEVC) encoding, thus partitioning each video frame into three tiles ($3 \times 1$ uniform tiling), each capturing a 120-degree viewing angle, which are encoded at high or low-quality resolution. Next, the DASH streaming server packetizes the bitstream data into multiple chunks, i.e., DASH segments, which are requested by the DASH client running on the edge server. Furthermore, the edge server decodes the tiled video streams and composes the 360 video frame for the UE. A key EFS component of our solution

[2]Available on https://www.wowza.com/products/streaming-engine

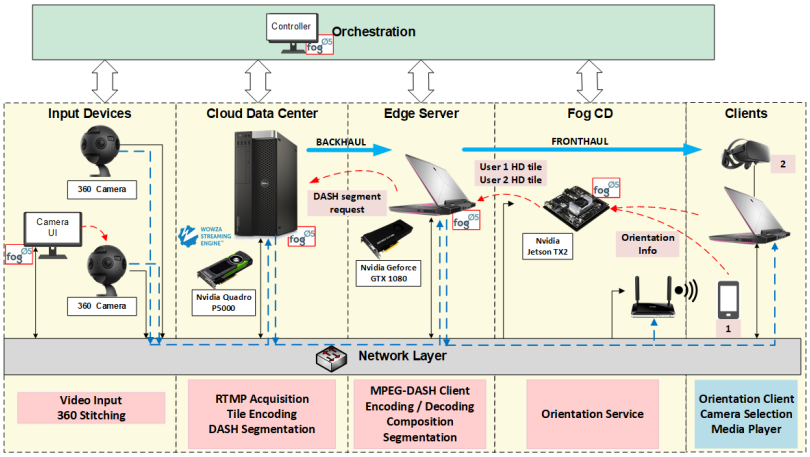


Fig. 4. Testbed overview with the specific HW employed.

are the fog CDs deployed nearby the UE, whose main task is to collect the viewing orientation from the UE and to provide via HTTP REST API the DASH client with the UE viewport, i.e., the portion of the 360 video that must be delivered at high quality. This way, the DASH client is able to quickly recompose the video frame by decoding the correct tiles, depending on the orientation information sent by the fog CDs. Note that all the EFS software processes are native applications, as virtualization has not been considered in this work.

*3) OCS:* orchestration and control are accomplished by means of Fog05. Specifically, Fog05 processes the request for instantiation of an immersive video app made by the OSS and assigns computing tasks to each EFS component running the Fog05 agent in order to build the end-to-end video streaming service. For instance, Fog05 may instantiate the orientation service into a number of fog CDs as well as instruct a service migration between two fog CDs to support user mobility.

*4) OSS:* this module triggers the instantiation of the end-to-end video streaming service by sending a request to the OCS. The OSS may be managed by the multimedia service provider or network operator offering the 360 video streaming.

*5) UE:* the user terminal consists of three components, namely, a media player, a camera selector and the orientation client. The first two elements are managed by the user, whereas the latter runs in the background and forwards information on the user orientation to the orientation service located on the fog CD.

## IV. System Setup

In this section, we present our testbed by illustrating the physical topology and the hardware equipment deployed. Also, we describe the methodology adopted, the system parameters and the metrics chosen.

### A. Testbed Description

Our testbed consists of a multi-tier computing, storage and networking platform capable of conveying multimedia traffic generated by one or more Insta360 Pro cameras to fixed and mobile clients. As shown in Fig. 4, each component is equipped with a Gigabit Ethernet network adapter and is connected to the network layer represented by a Gigabit Ethernet switch, whereas the phone terminal is connected to a Wi-Fi IEEE 802.11ac access point. For the sake of clarity, the tasks performed by each layer are listed at the bottom of the diagram. The top layer hosts the orchestration component, which deploys and manages all the entities running the Fog05 agent. This is achieved by using a laptop or any computer equipped with a screen and running the Fog05 agent, in order for the operator to execute scripts and verify the successful function onboarding.

As previously described, the 360 live video streaming is initiated by the cameras connected to Wowza Streaming Engine hosted by the cloud data center. To trigger the RTMP session, a computer laptop running the Fog05 agent is also plugged to the camera, thus allowing automatic session instantiation and termination. The Insta360 Pro camera is equipped with 6 fisheye lenses and can perform real-time stitching of 4K video sequences. Next, the data center handles the tile-based HEVC encoding by leveraging computing resources provided by the Nvidia Quadro P5000 GPU. Our data center consists of a Dell Precision Tower 7810 equipped with an Intel Xeon E5-2670 v4 2.30 GHz CPU, 64 GB RAM and 1 TB HDD storage. In addition, this machine runs a DASH server compliant with the latest MPEG immersive Omnidirectional Media Format (OMAF) standard. The DASH segments are then received by the edge machine represented by a Dell Alienware 17 laptop equipped with an Intel Core i7-8750H CPU, 16 GB RAM, 128 GB SSD and an Nvidia GeForce GTX 1080 graphic card, able to execute complex tasks such as tile decoding and video frame composition. Furthermore, the edge server exploits the orientation information supplied by the fog node. To this end, we use an Nvidia Jetson TX2 development board, consisting of a Jetson TX2 module, which embeds a powerful GPU and two ARM CPUs. It is worth pointing out that although this platform features a high-performance 256-CUDA

core graphic processor, we solely rely on the available CPU power, as the user orientation tracking is not meant to use hardware acceleration. The orientation service computes in real-time the video stream tiles that must be encoded in high definition by processing the orientation info, i.e., yaw, roll and pitch, periodically reported by the terminals according to the orientation report rate system parameter. Also, a count-down timer is associated with each reported tile: the timer can be configured by setting the orientation decay period system parameter, and is periodically decremented and reset whenever a new matching orientation is reported. Finally, the edge server transmits the optimized DASH video stream to the clients. Specifically, we consider two types of video terminals, i.e., a Samsung S9+ Android-based mobile phone and an Oculus Rift VR headset connected to a computer laptop. Also, we conveniently developed an Android app featuring a user media player capable of reporting the orientation info as well as tuning to one of the video streams according to the user choice.

### B. Evaluation Methodology

We ran a set of experiments by using a 35-second pre-recorded 360 video sequence stored on the data center, which feeds the Wowza streaming engine. The results are generated by executing the same experiment for 10 repetitions and by recording the metrics every second. A summary of the system parameters employed is reported in Table I. For the sake of simplicity, we only considered a stationary phone terminal and randomly changed its orientation in each repetition.

To show the impact of the video processing load on the data center, we retrieved the GPU load, the GPU power consumption and the memory usage by using a free system tool called GPU-Z, which supports NVIDIA video cards. The first metric gives an estimation of how active the GPU is in a given interval, whereas the second metric indicates the power in Watt consumed by the GPU, and the last one reports the memory used. Furthermore, we extracted such metrics for three different configurations: idle, i.e., the video streaming service is inactive; split mode, where all the computing tasks, except for the orientation service running on the fog CD, are distributed between the data center and the edge server; no split mode, where all the tasks are executed by the data center.

Finally, we obtained the bandwidth consumed in the backhaul and fronthaul, where fronthaul refers to the link between edge server and the terminals and backhaul to the link between the data center and the edge server. Moreover, we considered three different streaming modes in order to highlight the benefits of the adaptive tile-encoding strategy. The information on the bandwidth is obtained by using the Wireshark network monitoring tool, which allows to monitor the bandwidth consumed by distinct applications on the same machine.

## V. Experimental Results

We now discuss the results obtained by running the video streaming session as described above.

TABLE I
System parameters.

| Parameter | Value |
|---|---|
| Video Resolution | 4K (3840x1920) |
| Video Bitrate | 5 Mbps |
| Video Frame Rate | 30 fps |
| Orientation Report Rate | 10 Hz |
| Orientation Decay Period | 250 ms |

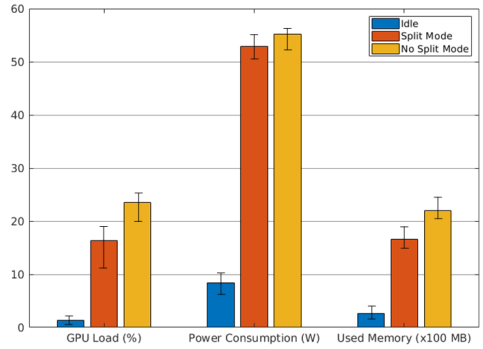


Fig. 5. GPU load, power consumption and memory usage on the cloud data center.

### A. GPU Load, Power and Memory Usage

Fig. 5 shows the GPU load, power consumption and memory used by the cloud data center measured through GPU-Z. As expected, the GPU computing load distribution between the data center and the edge server results approximately in a 7% reduction of the GPU load, which translates into more processing capacity available for other services running in the data center. Furthermore, the split mode leads to a small reduction of the power consumed by the data center. Specifically, up to 2 Watts can be saved by offloading some GPU processing onto the edge server. This also means that most of data center power is spent executing the tile encoding and the DASH segmentation together with the Wowza streaming engine. Finally, it is worth pointing out that our approach helps to reduce the data center memory occupancy, with a memory saving equal to 600 MB in comparison with the no split mode.

### B. Backhaul and Fronthaul Data Rate

The Empirical CDF of the backhaul and fronthaul data rate observed at the edge server side are shown in Fig. 6 and 7, respectively. In the first case, we note that the adaptive tile-encoding requires roughly a bandwidth equal to 21 Mbps, whereas a non-optimized approach encoding all the tiles with maximum quality leads to a bandwidth consumption equal to 31 Mbps, thus to a 33% increment. Obviously, this is due to the larger size of the DASH segment requested by the edge server, which results in a higher bandwidth consumption over the link between data center and edge server. Similarly, we point out that the adaptive approach ensures bandwidth reduction over the fronthaul as well. Specifically, we obtained an average value of 7.4 Mbps in adaptive tile-encoding mode and 7.8

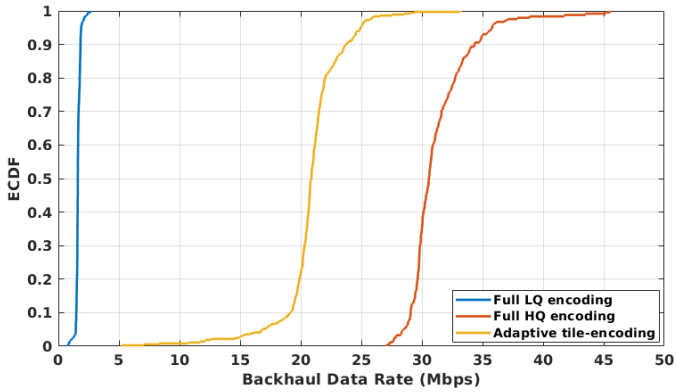


Fig. 6. ECDF of the backhaul link data rate.

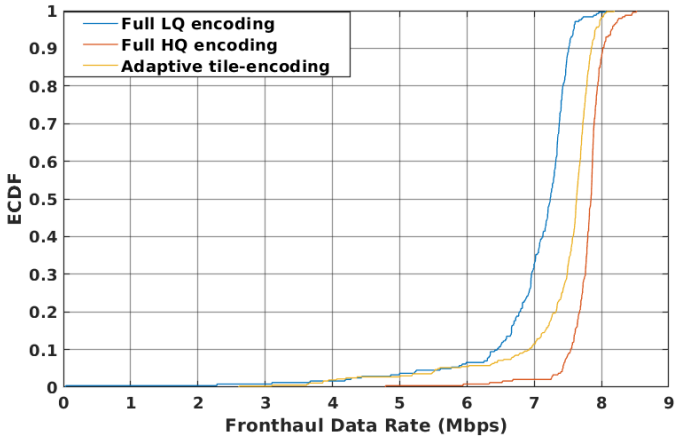


Fig. 7. ECDF of the fronthaul link data rate.

Mbps in full HQ encoding, which results in 5% reduction of the bandwidth. It is also worth noting that for low values of bandwidth, full LQ and adaptive approach achieves the same performance, while the latter tends to be closer to the full HQ approach as the consumed bandwidth increases.

## VI. Conclusions

This paper presented a solution for decomposing the end-to-end 360 video streaming service into micro-services which are then distributed across three computing tiers, namely cloud, edge, and fog, in order of proximity to the end-user client. The solution uses an adaptive viewport technique whereby only the field of view capturing the end-user client orientation is delivered in high quality whereas the rest of the 360 scene is delivered in low quality, yielding significant bandwidth saving. In addition, all the three tiers, which are composed of heterogeneous computing resources, are orchestrated and controlled using a unified management system based on Fog05.

Performance evaluation has been conducted using an actual testbed using actual hardware equipment, measuring different metrics, such as the GPU load, power consumption, memory usage, backhaul and fronthaul data rates. These measurements clearly demonstrated the benefits of the proposed solution compared to a conventional approach where the 360 video streaming service is executed in the Cloud.

However, the measurements obtained assume only a single-user scenario. It is anticipated that with our distributed solution a more significant reduction in bandwidth consumption can be achieved especially in dense environments where several users share the same field of view. This is thanks to the potential of aggregation across multiple users with same orientation angles. The performance evaluation with multiple users is planned in future work. Furthermore, as a next step, we plan on carrying the 360 video streaming out of cameras, on-board moving devices, such as robots or drones, using 5G wireless connectivity to the fog and edge tiers. This will help obtain meaningful insights on the latency measures and the deployment topology in these mobile scenarios.

## Acknowledgments

This work has been partially funded by the H2020 collaborative Europe/Taiwan research project 5G-CORAL (grant num. 761586).

## References

[1] Huawei Technology, "Whitepaper on the VR-Oriented Bearer Network Requirement, available at http://www-file.huawei.com/̃media/CORPORATE/PDF/white%20paper/whitepaper-on-the-vr-oriented-bearer-network-requirement-en.pdf." 2016.

[2] "3GPP TR 26.918 V15.2.0 (2018-03) Technical Report 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Virtual Reality (VR) media services over 3GPP (Release 15)."

[3] D. Rapone and et al., "An integrated, virtualized joint edge and fog computing system with multi-rat convergence," in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. IEEE, 2018, pp. 1–5.

[4] M. Graf, C. Timmerer, and C. Mueller, "Towards bandwidth efficient adaptive streaming of omnidirectional video over http: Design, implementation, and evaluation," in *Proceedings of the 8th ACM on Multimedia Systems Conference*. ACM, 2017, pp. 261–271.

[5] M. Hosseini and V. Swaminathan, "Adaptive 360 vr video streaming: Divide and conquer," in *Multimedia (ISM), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 107–110.

[6] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–7.

[7] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward low-latency and ultra-reliable virtual reality," *IEEE Network*, vol. 32, no. 2, pp. 78–84, 2018.

[8] L. Sun, F. Duanmu, Y. Liu, Y. Wang, Y. Ye, H. Shi, and D. Dai, "Multi-path multi-tier 360-degree video streaming in 5g networks," in *Proceedings of the 9th ACM Multimedia Systems Conference*. ACM, 2018, pp. 162–173.

[9] D. Nguyen, T. Le, S. Lee, and E.-S. Ryu, "Shvc tile-based 360-degree video streaming for mobile vr: Pc offloading over mmwave," *Sensors*, vol. 18, no. 11, p. 3728, 2018.

[10] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. ACM, 2016, pp. 1–6.

[11] "Mobile Edge Computing (MEC); Radio Network Information API, European Telecommunications Standards Institute, GS MEC 012," *ETSI*, 2017.

[12] "Network Functions Virtualization (NFV); Management and Orchestration; Architectural Options, European Telecommunications Standards Institute, GS NFV-IFA 009," *ETSI*, 2016.

[13] "fog05, Available on: https://projects.eclipse.org/proposals/eclipse-fog05."