A Distributed Fog/Edge Computing Solution for Enhanced 360-degree Video Streaming Services

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*—Extended-Reality (XR) technologies are expected not only to revolutionize the consumer market, particularly the entertainment and gaming business sectors, but also to impact a broad range of new use cases, such as construction, e-health and manufacturing. Due to the strict QoS requirements, such applications can be be supported only if the new 5G-NR wireless radio access technology is combined with flexible and agile edge and fog computing solutions. In this paper, we consider an end-to-end 360 video streaming service and distribute each single computing task over a three-tier system architecture provided by the 5G-CORAL framework. Furthermore, we design a D2D-aided clustered solution to drastically lower the signalization overhead originated from user terminals reporting their viewing orientation. Simulation results show that the probability of successful tile reporting increases when a small portion of users act as an aggregator and carry out the reporting on behalf of the other users.**

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

Fig. 1. A possible 360 immersive video streaming use case: multiple cameras can capture highlights from different angles during a live music concert.

I. INTRODUCTION

Recently, a wealth of innovative Extended-Reality (XR) use cases have drawn the attention of industry and consumer market, driven by the growing number of applications and services envisaged in diverse segments, including healthcare, manufacture, real-estate, entertainment and constructions. A recent market report forecast that applications based on Virtual Reality (VR) and Augmented Reality (AR) will reach \$150 Billions in revenue by 2020, creating new business opportunities and consumer products [1].

360 immersive video streaming is one of the most prominent XR applications, set to become mainstream in a wide range of use cases, such as music concerts, sport events or in a domestic scenario. Such technology allows a user to panoramically watch a video play by following the head movements. Specifically, the Field of View (FoV), i.e, the portion of the 360 frame being watched by the user, is continuously adapted as the viewing orientation changes. Fig. 1 shows a potential use case: multiple 360 cameras are deployed to record the live scene from different angles, allowing the users to enjoy the play from different viewpoints. However, this also comes with the price of having very strict QoS requirements to be met. As an example, a 360 video service streamed at 60 frames-per-second and 8K resolution results in a data rate around 360 Mbps, when High Efficiency Video Coding (HEVC) is employed [2]. Moreover, high reliability and low latency must be ensured, as a small delay in the data delivery may cause discomfort and motion sickness.

To meet such requirements, a combination of new technologies will be necessary. From one hand, the Third Generation Partnership Project (3GPP) 5G New Radio (NR) standard promises to provide large-bandwidth and ultra-low latency connectivity in high-density scenarios. On the other hand, fog and edge computing solutions offer new service deployment options, enhance flexibility and further reduce the latency between the service provider and the client. For instance, the H2020 5G-CORAL project focuses on integrating and orchestrating heterogeneous resources across multiple tiers with the goal of providing a more flexible and convenient deployment of diverse applications, such as the 360 video streaming service [3]. In this context, our work in [4] investigated the performance of a 360 video streaming service deployed over the 5G-CORAL platform and demonstrated how computing tasks can be offloaded from cloud datacenters and terminal devices onto edge servers and fog nodes.

In this paper, we aim to reduce the signaling overhead generated by users periodically reporting their viewing orientation to the fog nodes. We assume that a high number of users will be likely selecting the same viewing angle during a live event, and design a scheme to group these users in clusters and delegate helper nodes to perform the tile reporting on behalf of the cluster members. This allows to save uplink resources, as cluster members will notify the cluster head about the selected tile by using D2D connectivity, thus avoiding to transmit over the same channel. Simulation results show that the probability of successful tile reporting increases by adopting a small percentage of helper users as opposed to considering a traditional solution, where all the users report the viewing orientation independently.

The rest of the paper is organized as follows: Section II provides a state-of-the-art review of the past work related to this research topic, whreas Section III presents the 5G-CORAL system architecture as well as the 360 immersive video streaming use case. Moreover, Section IV describes the proposed solution and Section V discusses the numerical results obtained from simulations. In Section VI, we conclude the paper and outline the next steps.

II. Related Work

In literature, several papers have focused on enhancing the tile encoding approach with the goal of reducing the bandwidth waste. As an examples, authors in [5] propose an adaptive view-aware bandwidth-efficient 360 VR framework, which spatially divides the 3D mesh into multiple sub-meshes, with a 72% save of required bandwidth. In [6], a FoV rendering solution deployed at the mobile network edge is presented, which is meant to optimize the bandwidth and the latency requirements dictated by the 360 VR video streaming service. A viewport-adaptive streaming solution is also discussed in [7]. Such solution focuses on the Quality Emphasized Region (QER), which corresponds to the highest quality video region. An adaptive algorithm is then executed by the user in order to select a video representation suitable for the available bandwidth, whose QER is close to the viewport.

Multicast delivery of 360 video content is addressed in [8]. The authors optimize the transmission time and the power allocation in order to minimize the average transmission energy for a given video quality. Authors in [9] evaluate a new multi-path multi-tier 360 video streaming technique and investigate design tradeoffs on streaming quality and reliability. Also, multi-tier rate allocation schemes and chunk scheduling algorithms are assessed and extensive experiments are carried out by using real 5G 802.11ad bandwidth traces and real user FoV traces with various head movement patterns.

III. THE 5G-CORAL EDGE/FOG INTEGRATED PLATFORM

In this section, we first present the fog/edge computing platform proposed by 5G-CORAL. Secondly, we discuss the 360 immersive video streaming use case and explain how such service can be delivered through 5G-CORAL, dissecting the video streaming service and mapping each element onto the appropriate 5G-CORAL building blocks.

A. System architecture

The 5G-CORAL[1] project aims to develop an integrated fog/edge computing platform based on specifications introduced by ETSI Multi-Access Edge Computing (MEC) [10] and ETSI Network Function Virtualization (NFV) framework [11]. As shown in Fig. 2, 5G-CORAL architecture comprises two main logic entities, described as follows:

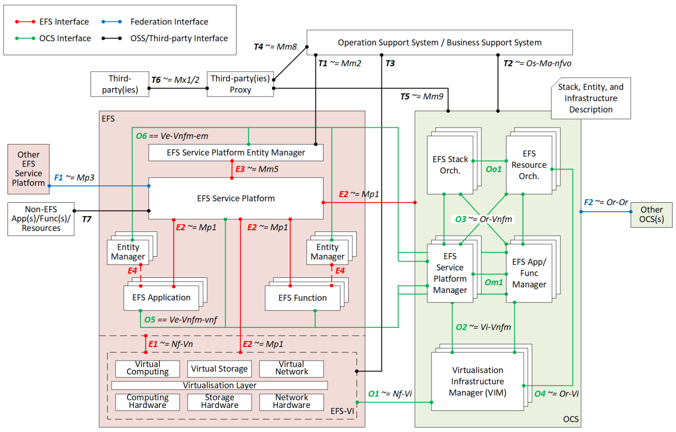

Fig. 2. Overview of the 5G-CORAL architecture.

1) *Edge and Fog computing System (EFS)*, storing virtualized and bare-metal applications, functions and services, which may be consumed by users or third-party entities. Specifically, the EFS Service Platform maintains information generated by applications/functions and edge/fog resources, and collects/provides data through messaging protocols, such as Message Queuing Telemetry Transport (MQTT).
2) *Orchestration and Control System (OCS)*, responsible for the EFS supervision and lifecycle and in charge of managing communications between the platform and external EFSs as well as OCSs. Moreover, the OCS features the EFS Stack Orchestrator, whose main task is to compose the end-to-end service by orchestrating specific modules and triggering the instantiation of appropriate EFS functions and applications.

A key OCS component is the Virtual Infrastructure Manager (VIM), which operates and manages the EFS compute, storage and networking resources. Due to its scalability and highly distributed nature, Fog05 has been selected as VIM in 5G-CORAL. Fog05 is an open-source fog computing platform providing abstractions that interconnect resources among different tiers, such as cloud, edge and IoT devices [12]. Such abstractions are coupled with Fog05 entities, which can be atomic, including virtual machines, executables or containers, or can be represented by a direct acyclic graph (DAG) of entities. To manage a system resource, a host must install and run the Fog05 agent, which can discover other Fog05 nodes by relying on the distributed store, a shared environment accessible through Pub/Sub protocols, such as Data Distribution Service (DDS) and Zenoh. Furthermore, Fog05 ensures extensibility and compatibility through plugins, that can be employed to manage an OS executed in a remote fog node.

---

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

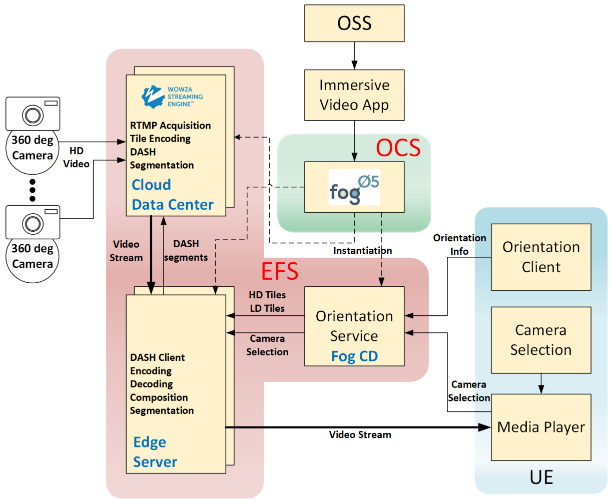

Fig. 3. A breakdown of the 360 streaming service with the 5G-CORAL blocks overlaid.

B. Use-case study: 360 immersive video streaming

The 360 immersive video streaming service builds upon a collection of computing tasks heavily relying on GPU-computing modules as well as specific source and terminal HW equipment, such as 360 panoramic cameras and smartphones or head-mounted devices (HMD). This well fits with the 5G-CORAL computing platform, being highly distributed and capable of managing heterogeneous resources. In Fig. 3, we illustrate each component taking part in the end-to-end video streaming service along with the overlaid 5G-CORAL building blocks.

First of all, the 360 panoramic video input is supplied by multiple 360 cameras, each installed in different locations and capturing the same scene from various angles. Each video stream is sent to a streaming engine through the Real Time Multimedia Protocol (RTMP), thus establishing a reliable TCP-based persistent connection with the cameras.

The EFS platform hosts all the services, applications and functions necessary to execute the streaming service, distributing them across three different tiers, as follows:

- **Cloud Datacenter**. This is the place where the most computation- and power-demanding tasks are performed. As mentioned above, the cloud datacenter first establishes an RTMP connection with the 360 cameras. Next, tile-based High-Efficiency Video Coding (HEVC) is carried out: the video frame is divided into three tiles, each mapped onto a different 120-degree view angle, and only the FoV being watched by a specific user is encoded at the highest resolution possible. Furthermore, the bitstream data is segmented and the DASH segments are sent to the DASH client.
- **Edger Server**. An edge server is meant to be deployed closer to the end user, e.g., in a shopping mall or a concert venue. Here, DASH segments are reassembled and frame composition is performed. In addition, the edge server receives up-to-date notifications on the tiles being watched by each user and uses this information to request optimized DASH segments from the DASH server.
- **Fog Nodes**. Such computing devices are placed nearby the users and periodically fetch the user viewing orientation, which is then translated in the appropriate viewport ID and forwarded to the edge server. A fog node can be any HW-constrained computing device, ranging from access points and mini-PCs to even smartphones.

The OCS module leverages on Fog05 in order to process service instantiation requests made by an Operation Support System (OSS) and deploy functions and applications on the physical resources belonging to the EFS. Also, the OCS can trigger the migration of a service, e.g., the orientation service, from a fog node to another based on measurements supplied by a monitoring plugin.

Finally, the User Equipment (EU) integrates the orientation client as well as a media player application combined with a camera selection block, which allows the user to switch among the available 360 cameras. As mentioned earlier, the orientation client periodically transmits to the fog node the tile ID being watched by the user, with the goal of feeding the adaptive tile-encoding process running on the cloud datacenter.

IV. D2D-AIDED CLUSTERED TILE REPORTING

In this section, we describe the system model and present our solution to reduce the signaling overhead produced by multiple users watching the same live event. The performance of such technique is then evaluated through simulations and discussed in the following section.

A. System Model

We consider a scenario where the users are stationary and randomly located on a square of side L with density λ_U (users$/m^2$). As shown in Fig. 4, each user is assumed to watch only one of the three tiles with a certain probability p_t in a given interval. This is motivated by the fact that a user may be interested in watching the play from a different angle, and a specific concert or sport event highlight may be captured by a certain tile, therefore selected with a higher probability than the rest of the 360 frame. We call it the *popular tile*. Furthermore, a number of fog nodes are randomly deployed on the same square with density λ_f (nodes$/m^2$) and each user associates with the closest fog node, thus applying a Voronoi tessellation. We also ignore wireless channel impairments and assume that all the users are able to successfully deliver the tile report within the time interval D_{UL}, provided that enough resources have been allocated.

We focus on the tile ID reporting performed by each user. Specifically, the users send the ID of the tile being watched to the closest fog AP by adopting a Time Division Multiple Access (TDMA) scheme, which will then forward the information to the edge server. We model the time spent to execute the first operation as follows:

$$D_{UL}(t) + D_{comp}(t) \le D_{th}, \tag{1}$$

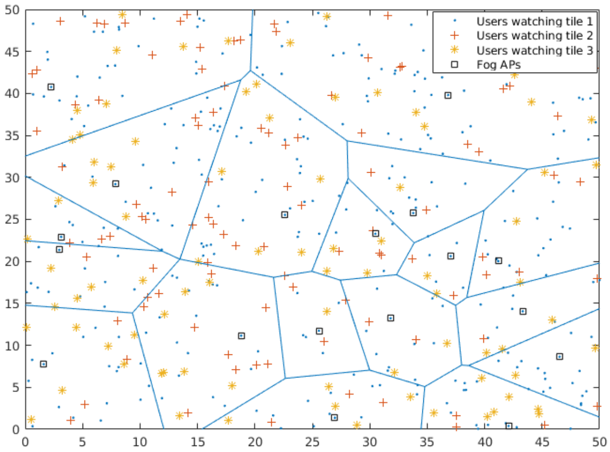

Fig. 4. A Voronoi tessellation of the reference scenario.

where D_{UL} is the uplink transmission delay and D_{comp} is the computing delay. We also assume that their sum must be lower than a given threshold D_{th}, otherwise the tile report is not successful. Therefore, the probability of a successful tile report can be formulated as follows:

$$P_R = Pr\{D_{UL}(t) + D_{comp}(t) \leq D_{th}\}. \tag{2}$$

As the number of users consuming the 360 video streaming service increases, such probability will decrease, being the system not able to meet the deadline D_{th}, with consequent user QoE degradation.

B. Proposed Solution

In order to enhance the overall system scalability, we propose a method to increase the number of users successfully reporting the tile watched. Fig. 5 shows a sequence diagram capturing the message exchange among all the entities involved in the process. We assume that a percentage of users watching the popular tile are elected as helpers or aggregators, since a high number of users are likely to watch such tile. These users will carry out the tile reporting on behalf of the other users watching the same popular tile, thus acting as cluster-heads, where all the associated users will belong to the same cluster. As a consequence, only the helpers will report the tile ID to the fog AP. Also, since normal users will not transmit on the uplink channel, more resources will be available for users watching non-popular tiles, which results in a higher probability of successful tile reporting. We also assume that the helpers can identify the users watching the same popular tile through context information exchanged via D2D communications. As an example, a user can disseminate this information to nearby users by relying on D2D broadcast announcements. Once the fog AP has received the user tile ID, the information is sent to the edge server. Finally, the DASH server processes the tile ID and adapts the encoding process accordingly. In the next section, we will evaluate the benefit of this approach through simulations and show the achievable gain in terms of reduction of uplink connections and processing time on the fog nodes.

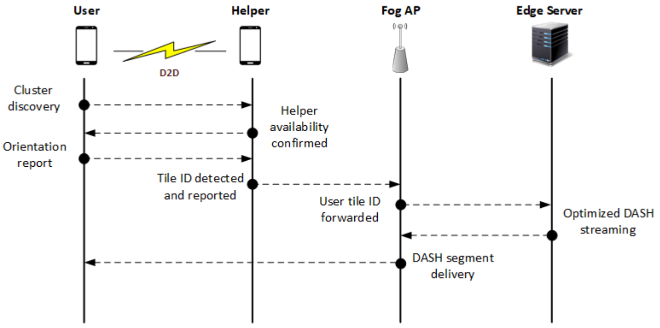

Fig. 5. Sequence diagram of the interaction between user, cluster head, fog AP and edge server.

TABLE I
SYSTEM PARAMETERS.

Parameter	Value
L	50 m
λ_f	0.2
p_t	$[0.6, 0.2, 0.2]$
D_{UL}	$10^{-4}s$
D_{comp}	$0.5^{-3}s$
D_{th}	$10^{-2}s$

V. NUMERICAL RESULTS

In this section, we discuss the performance of the method proposed and compare it with a baseline approach, where helper nodes are not employed and all the users periodically report the tile ID to the closest fog node. Numerical results have been obtained through Matlab simulations by employing 10^5 repetitions of the same experiment, and a list of system parameters is reported in Tab. I.

Fig. 6 shows the probability of successful tile reporting for different values of user density. Green lines refer to the baseline approach, whereas the remaining lines capture the performance of our solution in terms of successful reporting of the popular tile and non-popular tiles. We first note that users watching the popular tiles, i.e., the majority of the users, can significantly benefit from the D2D-aided clustered approach. Specifically, the probability is close to the maximum value for low user densities and small percentage of helpers employed. This is due to the reduced number of cluster-heads requesting uplink resources to forward the viewing orientation. The probability significantly decreases when the amount of cluster-heads deployed doubles, whereas the impact is less evident on users associated with non-popular tiles, since the clustered approach mainly focuses on reducing the signaling overhead generated by the majority of the users. By contrast, it is worth pointing out that the QoE of such users quickly degrades as the user density increases when the proposed approached is not employed, due to the higher number of users watching the popular tile.

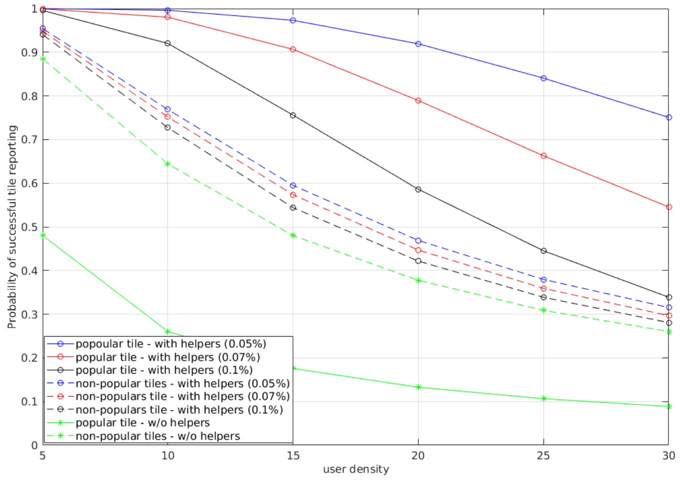

Fig. 6. Probability of successful tile reporting for increasing values of λ_U and different configurations.

VI. Conclusions

In this paper, we presented a novel method to deploy XR applications, in particular 360 immersive video streaming services, by distributing the processing power among cloud datacenters, edge servers and fog nodes. We introduced the 5G-CORAL approach and grouped the 360 video computing tasks into the EFS, thus managing the service deployment through the OCS. Also, we proposed a new method to reduce the signaling overhead on the uplink channel by electing some users as cluster-heads and using D2D connectivity to forward the viewing orientation. As future work, we intend to evaluate mechanisms to enable aggregation on the downlink channel and classify users in multicast groups based on the tile ID reported.

Acknowledgments

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

References

[1] Digi-Capital, “Augmented/Virtual Reality Report - Q2 2019,” 2019.

[2] 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.

[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] G. Rigazzi, J.-P. Kainulainen, C. Turyagyenda, A. A.-M. Mourad, and J. Ahn, “An Edge and Fog Computing Platform for Effective Deployment of 360 Video Applications,” in *Cloud Technologies and Energy Efficiency in Mobile Communication Networks (CLEEN 2019) (WCNCW-CLEEN 2019)*, Marrakech, Morocco, Apr. 2019.

[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] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, and M. D. Silva, “Vr is on the edge: How to deliver 360 videos in mobile networks,” in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*. ACM, 2017, pp. 30–35.

[7] 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.

[8] C. Guo, Y. Cui, and Z. Liu, “Optimal multicast of tiled 360 VR video,” *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 145–148, 2019.

[9] 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.

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

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

[12] “fog05, Available on: https://projects.eclipse.org/proposals/eclipse-fog05.”