Edge-based 5G Network Architectures in support of Zero Downtime Mobility for Enterprise Applications

Dan Warren[†], Xenofon Vasilakos[‡], Walter Featherstone[†]

[†]*Advanced Network Research, Samsung Research UK*
Communications House, South Street, Staines-Upon-Thames, Surrey, TW18 4QE. UK

[‡]*Smart Internet Lab, Department of Electrical & Electronic Engineering,*
University of Bristol, Bristol, Clifton BS8 1UB, UK

dan.warren@samsung.com, xenofon.vasilakos@bristol.ac.uk, walter1.f@samsung.com

Abstract: Ultra-Reliable Low Latency 5G applications require Edge implementation deep in access networks, resulting in the need for inter-Edge Server application mobility. This paper identifies implementation methodology for zero-downtime Edge application mobility.

1. Introduction

5G networks are intended to support technology enablers that will open new revenue streams to mobile network operators, by allowing them to target enterprise customers with high reliability and low latency connectivity. Specifically, network availability in excess of the historic benchmark for fixed-line services of 99.999% uptime guarantee and round-trip latency of below 10ms have been proposed [1]. This level of performance is required to meet the demands of a class of services referred to as Ultra-Reliable Low Latency Communication (URLLC).

At this level of latency, delay induced by distance covered on transit networks becomes a major contributory factor to overall round-trip delay. Therefore, network topologies where the user plane of an application is generated close to the end-user are needed [2]. This has led to the definition of Multi-access Edge Computing (MEC), with ongoing activities in both ETSI [3] and 3GPP [4]. MEC deployment places the User Plane Function (UPF) and some elements of both the control plane and application, on servers located close to the Access Network. This implies that MEC servers need to be hosted in multiple, distributed locations, rather than as per previous network generations where core network control and user plane functions would be found in a smaller number of centralised points.

For a URLLC service to be delivered with low latency in a MEC deployment, application and user plane traffic will need to be moved from one MEC server to another. This must be achieved whilst the application is in operation at the end-user client. This is technically challenging to achieve, since URLLC applications are intolerant of service interruption, but such a significant mobility event would, under existing techniques, result in a service outage.

In order for the promise of URLLC service over 5G networks to be fulfilled, the three requirements of mobility, low latency and high reliability must be met simultaneously. The aim of the work summarised in this paper is to achieve zero downtime edge application mobility. The work is described in more detail in [5].

2. Approach

To experiment with approaches to achieving zero downtime edge application mobility, the Smart Internet Lab at the University of Bristol has established a network architecture as depicted in Fig. 1. This architecture is implemented to enable a demonstration of MEC mobility events for a multi-player gaming application. Multi-player gaming is a good application to use for such a demo since such games are highly interactive, hence demanding low-latency communication to convey both the actions of the user as they interact with other players, and for the user to experience the actions of other players quickly enough to be able to react. It is also important that no outage in the game is experienced when the gaming application instance is moved from one MEC server to another, as this could result in players 'losing' the game without being able to participate. The findings of the research can be generalised to other MEC use cases including Remote Desktop and Augmented Reality [6]; Industrial Internet-of-Things (IIoT) [7]; Autonomous Driving [8]; and eX-tended/Virtual Reality (XR/VR) or holographic services [9].

With reference to Fig. 1, a Care-of-App (CoA) and Video Application (VA) are provisioned for User *i* at the MEC server providing service to them. The CoA maintains an up-to-date view of a user's state in a game session. The CoA tracks state changes after the user's actions, receives state updates from other players' CoAs, and combines all states to update the user's VA. If there is also a Cloud back-end, CoAs sync with the Cloud for global states (e.g., game

meta-data). VA acts as the streaming source. In the experimental setup, both the CoA and the VA are implemented as Kubernetes-based containers.

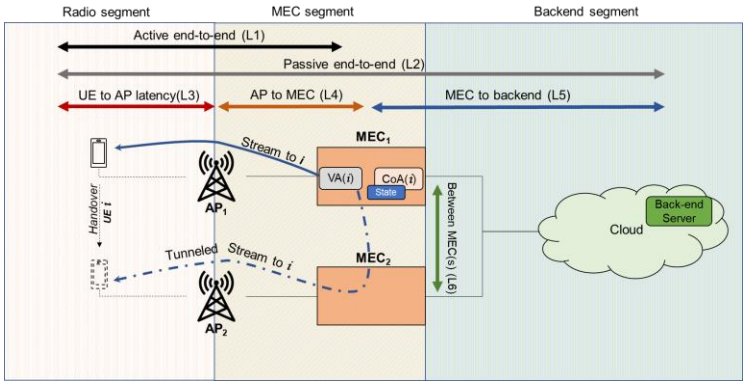

Fig.1. Edge Architecture used for zero downtime edge application mobility demonstration

Locating the VA on the MEC server allows video traffic to be delivered to the user with ultra-low delay. Collocating CoA and VA enables the CoA to pass the user's state to VA with ultra-low latency, so that VA can quickly reflect state changes to its video output stream to the user. Using this set-up, mobility from Access Point 1 (AP1) to AP2 requires handover of the application from MEC1 to MEC2. Further detail of this architecture is provided in [5].

3. Evaluation of Methods

In an operational network, MEC1 would have multiple 'neighbour' MEC servers, any of which could be the next MEC server that would optimally serve User *i*'s traffic following a handover. Three approaches are considered;-

- *Approach 1* - a reactive handover from MEC1 to a target MEC2 when requirement for handover occurs.
- *Approach 2* - pre-emptive establishment of duplicate CoA and VA, and replicating traffic to all possible target MEC servers, to enable User *i* to move to any neighbouring server, with service already established.
- *Approach 3* - prediction of mobility events far enough in advance of the event occurring, to allow pre-emptive establishment of duplicate CoA and VA at specific MEC2, ahead of the handover event.

For Approach 1, when the handover of the application happens reactively, current state-of-the-art methodology is to use Checkpoint/Restore in Userspace (CRIU). CRIU freezes a containerised application and then allows it to be moved and re-instantiated on a different server. However this process can take several seconds. For gaming, this break in activity is a serious outage in service which will severely damage the end-user's perception of Quality of Experience (QoE). For mission critical applications in enterprise use cases, or for Vehicle-to-Anything (V2X), such a break in service could result in severe damages, costs or accidents.

To overcome this, it is necessary for the CoA and VA of the user to be available in MEC2 ahead of the handover event taking place. Approach 2 would allow User *i* to move to any potential MEC server that neighbours MEC1, but would result in massive redundant usage of resource. In a fully implemented network, all potential target MEC servers would have to provision resource and capacity to instances of CoA and VA for every user that has a MEC-based application running live on a neighbouring MEC server. However, only one of these servers would actually take up service in event of a handover, and there is no guarantee that a handover between MEC Servers would actually occur.

Approach 3 requires a method to accurately predict when handover is likely to occur, and to pre-provision a parallel instance of CoA and VA for User *i* ahead of the handover taking place. This can be achieved through the use of Machine Learning (ML). The ML method learns the conditions of users that are about to undergo a MEC handover, by receiving inputs from the network. ML techniques such as Artificial Neural Networks (ANNs) and Long/Short Term Memory (LSTM) are particularly useful since they have the ability to adapt to dynamic conditions and are able to deliver system-wide predictions to fine tune parameters.

The employed ML approach needs to strike a fine balance between the accuracy of its prediction (which will increase as the user becomes more likely to handover, but leave less time for the instantiation of the CoA and VA in

the target MEC Server) and the need to provision CoA and VA early enough for instances to be active on the target MEC Server prior to the handover occurring. If the handover is predicted too early, then the chance that the handover doesn't occur will remain high, and so resources on the target MEC for CoA and VA instances may be allocated unnecessarily. If the handover is predicted too late, however, there will be a service outage since CoA and VA instances will not be established before a handover does occur. This is described in detail in [5].

4. Results and Further Work

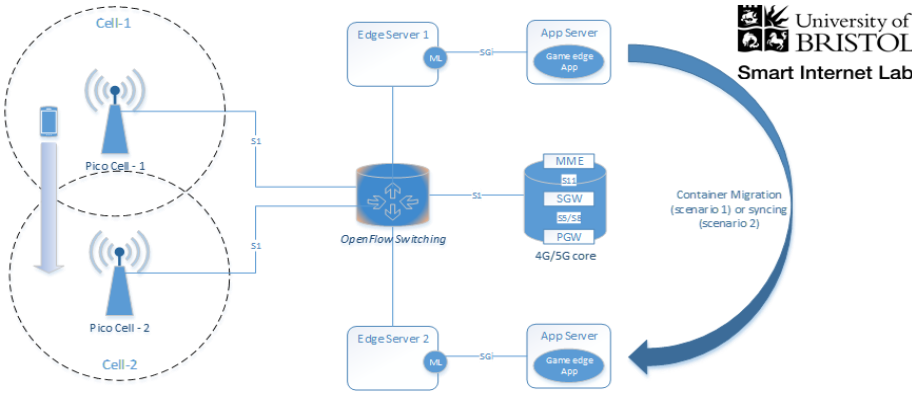

Fig.2. Demonstration experimental setup at Smart Internet Lab

To demonstrate and test the above approaches, a physical experimental testbed configuration portrayed in Fig. 2 was setup at the Smart Internet Lab premises. Approach 1 was used to set a benchmark for service interruption without any additional mechanism targeting a handover between MEC servers. Using CRIU, the interruption in the application service was found to average 5.49 seconds. This is clearly unacceptable for URLLC applications.

When Approach 3 was employed a vast reduction in the service interruption was achieved: a mere 25ms. The phenomenal improvement in interruption time is sufficient that, for gaming applications and for many URLLC applications, no perceptible interruption would be experienced by the end-user (even when the end-user is non-human). Because the interruption is so short, it is clear that the ML method applied to predicting handover is performing extremely well, since the CoA and VA for the user are fully provisioned when the handover event occurs.

The demonstrations conducted so far have used video streaming and multi-player gaming applications to illustrate the differences in results between approaches. While neither application could be considered 'mission critical', the principles apply to applications employed in Enterprise use cases with criticality for minimised interruption.

Further research work continues to consider: how the methodology could be optimised further to specific MEC and Access Network conditions through profiling; how the CoA and VA state impacts the speed of migration (and hence the lead time required for handover prediction and its relative accuracy), and hence to consider state as an input to ML; and how IP addressing and routing could be optimised to reduce mobility impact on route selection through use of data-plane technologies.

5. References

[1] 5G Initiative Team. 5G White paper by NGMN Alliance, February 2015
[2] Warren, D and Dewar, G, "Understanding 5G: Perspectives on future technological advancements in mobile", GSMA Intelligence, December 2014
[3] Multi-access Edge Computing (MEC), https://www.etsi.org/technologies/multi-access-edge-computing
[4] "Architecture for enabling Edge Applications (EA)", 3GPP TS 23.558, Release 17.
[5] Vasilakos, X et al., "Towards Zero Downtime Edge Application Mobility for Ultra-Low Latency 5G Streaming", 2020 IEEE 9th International Conference on Cloud Networking (CloudNet), 2020.
[6] A. Ceselli et al. "Mobile edge cloud network design optimization". IEEE/ACM Transactions on Networking, 25(3):1818–1831, 2017
[7] B. Yang et al. "Mobile-Edge-Computing-Based Hierarchical Machine Learning Tasks Distribution for IIoT". IEEE Internet of Things Journal, 7(3):2169–2180, 2020.
[8] G. Tang et al. "QoS Guaranteed Edge Cloud Resource Provisioning for Vehicle Fleets". IEEE Transactions on Vehicular Technology, 2020.
[9] L. Wang, L. Jiao, T. He, J. Li, and M. Muhlhauser. "Service entity placement for social virtual reality applications in edge computing". In IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pages 468–476. IEEE, 2018.