# *lightMEC*: A Vendor-Agnostic Platform for Multi-access Edge Computing

Tejas Subramanya, Giovanni Baggio, Roberto Riggio
Wireless and Networked Systems
FBK CREATE-NET
Via Alla Cascata 56/C, 38123, Trento, Italy;
Email: {t.subramanya,g.baggio,rriggio}@fbk.eu

***Abstract*—Multi-access Edge Computing promises to improve mobile users experience by bringing storage and processing capabilities to the edges of the network. This approach allows third parties to deploy innovative services, like augmented reality, directly in the radio access network paving the way for new business models and new revenue stream for mobile network operators (MNOs). In this paper, we introduce *lightMEC* a lightweight solution for deploying mobile edge computing functionalities which allows hosting of low-latency and bandwidth-intensive applications at the network edge. Measurements conducted over a real-life test demonstrated that *lightMEC* can actually support practical MEC applications without requiring any change to the functionality of existing mobile network nodes both in the access and the core network segments. The significant benefits of adopting the proposed architecture are analyzed based on a proof-of-concept demonstration of the content caching use case.**

***Index Terms*—5G, Multi–access Edge Computing, LTE, Edge Caching, NFV, Proof-of-concept.**

## I. Introduction

In the past, cloud computing has pushed value creation away from mobile networks, and towards the cloud data centers. This, to the benefit of Over-The-Top (OTT) service providers, mostly at the expense of Mobile Network Operators (MNOs). In the sense, MNOs invest in expensive mobile network infrastructure necessary to deliver services, but their revenues keep shrinking. On the other hand, the revenues of Over-The-Top (OTT) service providers, who utilize the MNOs network and infrastructure to deliver services, continue to grow year on year. This argument is also backed up by the ETNO (European Telecommunications Network Operators Association) annual economic report 2017 [1] which clearly acknowledges the negative impact of OTT services on MNOs revenue. But, is there a way for MNOs to be more competitive? Yes, edge computing can provide MNOs a competitive advantage over the OTT players.

In recent years, with the rise in smartphone subscriptions, mobile data traffic has continued to grow at a significant rate. According to the Cisco Visual Networking Index [2], by 2021 the annual mobile data traffic will exceed half a zettabyte, of which 78% will account for mobile video traffic. With the proliferation of connected devices and digital services, mobile cloud computing is going through a radical transformation in which the conventional model of accessing highly centralized resources is replaced by a distributed architecture. This new paradigm called Multi-access Edge Computing (MEC) brings the core components of cloud i.e., compute, storage, and networking closer to the network edge (e.g. base station), facilitating the hosting of delay-sensitive services and applications (see Figure. 1). Thus, emerging low-latency 5G applications benefit from MEC since the end-to-end latency involved in the round trip to the cloud gets reduced thus providing better experience to end users. Additionally, to alleviate the load on backhaul networks, popular videos can be cached at the edge of the mobile networks to reduce the duplicate content transmission. On the other hand, to realize such a distributed network architecture, it is necessary to provide a flexible, scalable, and programmable networking platform over which various services with differing requirements can be deployed and managed under strict performance bounds. In this regard, Network Function Virtualization (NFV) is considered being as one of the key building blocks for MEC. NFV simplifies the network service creation process by decoupling network functions from the underlying hardware and transforming them into software entities called Virtualized Network Functions.

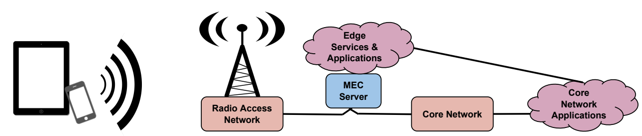



Fig. 1: MEC System Overview.

In this paper, we present a proof-of-concept implementation of a practical MEC architecture named *lightMEC* leveraging lightweight virtualization technologies such as Dockers Containers [3] and Click unikernels [4]. We further validate our architecture by performing real experiments with content caching as a use case. Results show how *lightMEC* can actually improve MEC specific metrics such as latency.

The rest of the paper is organized as follows. Section. II describes the related work. Section. III describes the *lightMEC* architecture and system design along with its functional elements, while Section. IV describes the implementation details. The testbed experiments and its numerical results are detailed in Section. V. Finally, we conclude the paper in Section. VI.

978-3-903176-14-0 © 2018 IFIP

## II. Related Work

MEC aims at converging IT services and telecommunications, providing cloud-computing capabilities to deploy various applications at the edge of the radio access networks. By bringing storage resources, computational resources and applications at the network edges, both service latency and backhaul load can be reduced significantly. MEC also provides real-time access to radio network information that can be leveraged by authorized third parties such as content providers, to develop innovative MEC applications and services targeted towards mobile users, enterprises and vertical segments.

The European Telecommunications Standards Institute (ETSI) is active in the MEC arena with a dedicated Industry Specification Group (ISG). The activities of this group are still in their early stages, nevertheless a number of drafts specification are already available. For example, [5] describes the technical requirements of MEC, while its framework and reference architecture are illustrated in [6]. The service scenarios benefiting from MEC and the proof-of-concept implementation details are illustrated in [7] and [8] respectively.

The MEC related research has received a considerable attention in recent times. The authors in [9] provide an architectural blueprint for MEC and discuss the technical advantages it offers by presenting a number of use cases. A taxonomy of MEC along with its key attributes are discussed in [10]. The authors also present some promising real-time MEC application scenarios. The challenges involved in commercial deployment of MEC and the progress towards it are discussed in [11]. In [12] and [13], the authors present different mobile offloading techniques in cellular networks to enable low-latency applications. The authors in [11] introduce a MEC platform called WiCloud that provides edge computing and proximity services for innovative applications. They also discuss on the current progresses and the challenges associated with MEC. In literature, several other MEC architectures have been proposed such as Mobile Micro-cloud [14], MobiScud [15], Follow-MeCloud [16] and CONCERT [17].

Most of the work reported above are either based on simulations and analytical evaluations that validate only simplistic scenarios. Conversely in this paper we provide a practical implementation of the proposed MEC framework based on caching use case.

## III. System Design and Architecture

In this section, we examine the potential options where the Multi-access Edge (ME) node can be deployed within the mobile network. Furthermore, we discuss the main benefits and design challenges associated with each option that needs to be considered before the commercial deployment of MEC. We determine the best option for our use case based on several factors such as performance, required resources, cost and physical deployment constraints. We then propose a MEC Architecture called *lightMEC* and present a high-level overview of the components involved and its interfaces.

### A. Deployment options

Figure. 2 depicts all three possible deployment options considered by ETSI in [18].

The first option is to deploy the ME node directly at the eNodeB. In this option, since the ME node is in close proximity to the end users, it results in very low end-to-end latency meeting the requirements of real-time applications. However, since computational power and storage capacities are limited at the eNodeB site, only applications requiring low compute and storage resources are served by the ME node collocated directly with the eNodeB.

The second option is to deploy the ME node at cell aggregation points, also called as Mobile Telephone Switching Offices (MTSOs) or at multi-RAT aggregation points. In this option, since each ME node serves a cluster of eNodeB's, the deployment cost is significantly reduced. Also, with the possibility of having high compute and storage capacities at MTSOs, the ME node can serve high demanding applications with ease. However, there is a trade-off between low-latency and deployment cost. From the system design point of view, first and second options are considered as Bump-in-the-wire (BITW) approaches, where the ME node is placed on the LTE S1-interface connecting the eNodeB and the EPC.

The third option is to deploy the ME node at the edge of the Evolved Packet Core (EPC). While this could result in a higher latency, with the recent advances in virtualization technologies, ETSI is also considering to include all or part of the EPC network functions as Virtual Network Functions (VNFs) together with the ME node which is placed on the LTE user plane interface (SGi) connecting the mobile network to the external packet data network.

One of the key functionalities of the ME node is to steer IP packets from eNodeB/EPC mobile network entities to MEC applications running on the node. The MEC applications can either terminate the IP packets by itself (end-point mode) or modify the IP packets and pass it back to the original PDN connection (pass-through mode). In BITW approach, since S1 user plane traffic is GTP encapsulated, the ME node has to monitor S1 control plane messages to maintain the UE context information and perform GTP decapsulation/re-encapsulation and routing operations on S1 user plane packets. However, this stateful operation can be performed requiring no modifications to either eNodeB or EPC protocol stacks which will be discussed later in detail.

Our focus being mobile edge caching in heterogeneous networks, we consider placing the ME node at cell aggregation sites/multi-RAT nodes. The decision is driven by the design modularity of BITW approach and on the possibility of caching HD videos considering the high storage capacity that can be made available at cell aggregation sites.

### B. System Architecture

Figure. 3 depicts the *lightMEC* system architecture and its functional elements that comprise the mobile edge system. On a broader context, the mobile edge system consists of the mobile edge host and the mobile edge management that

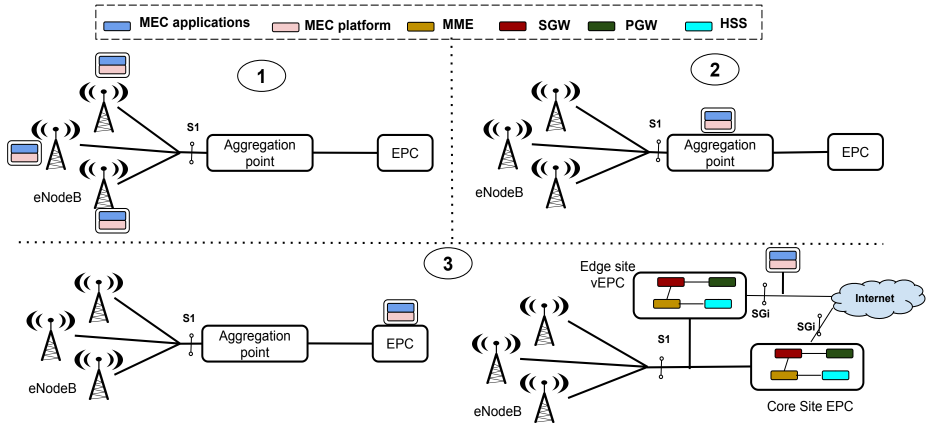



Fig. 2: ME node deployment options.

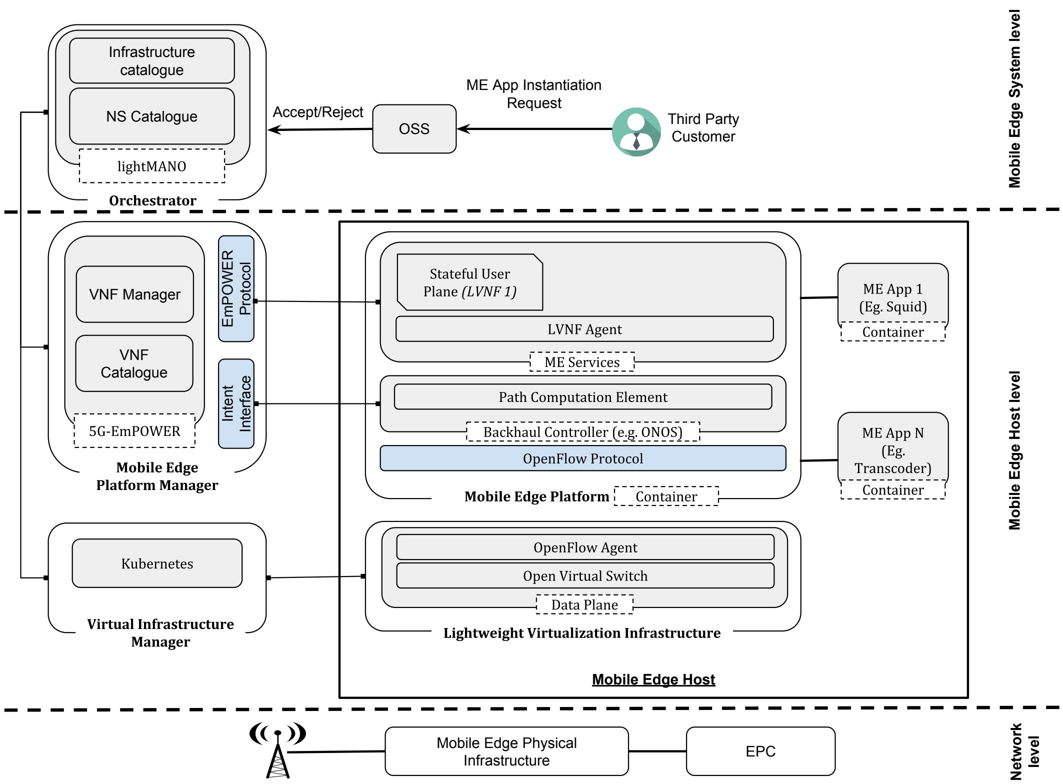
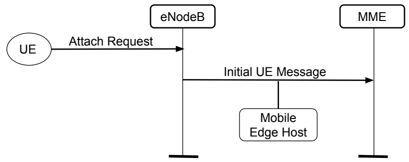



Fig. 3: *lightMEC* System Architecture.

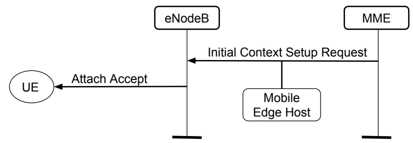
facilitates to run mobile edge applications. It is to be noted that the proposed architecture is in-line with the ETSI MEC reference architecture [6].

**Mobile Edge Host.** This entity contains a mobile edge platform and a virtualization infrastructure built on Docker Containers and Click unikernels which provides lightweight compute, storage and networking resources for running mobile edge applications. The virtualization infrastructure also includes an OpenFlow virtual switch to route traffic among 3GPP network elements, the mobile edge services, and the mobile edge applications. The mobile edge services running on the mobile edge platform are realized using the Click Modular Router [4] and are here referred to as Light Virtual Network Functions (LVNFs), the details of which are discussed in the next section. The mobile edge applications are running as individual containers and can interact with mobile edge platform to consume the services being hosted on it. The applications can range from caching to video transcoding to deep packet inspection.

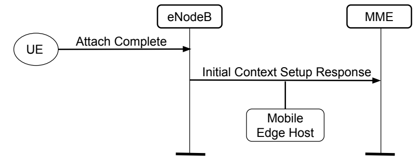
**Mobile Edge Management.** This entity includes the virtual




(a) Attach Request.




(b) Attach Accept.




(c) Attach Complete.

Fig. 4: UE Attach Process.

infrastructure manager, the mobile edge platform manager and the orchestrator. Kubernetes [19] is used as a platform for managing container-based virtual infrastructure within the mobile edge host while 5G-EmPOWER [20] and lightMANO [20] are used as, respectively, mobile edge platform manager and orchestrator. The mobile edge platform manager is responsible for interacting with the backhaul controller through an Intent-based networking interface and to deploy the mobile edge services LVNFs within the mobile edge host. The orchestrator has a global view of the mobile edge system regarding supported mobile edge services, available infrastructure resources, and the topology. Information about all supported LVNFs and mobile edge services are maintained in the application specific VNF catalogue. The orchestrator, depending on the application instantiation/termination request received from the OSS of an operator prepares the mobile edge platform manager and the virtualization infrastructure manager to allocate resources, to deploy services and to handle mobile edge applications.

### C. UE context management

We now describe two important functions of LTE i.e., session management and mobility management, to illustrate how the ME services extract the UE context information from LTE control plane messages during UE attach and handover procedures in a transparent manner. This function is fundamental in order to implement the BITW MEC deployment option used in this paper. We remind the reader that this option requires to put the ME host on the S1 interface between the eNodeB and the EPC. In order to allow ME Applications to access the inner IP traffic the GTP header must be removed. However, traffic flowing from the ME Applications to the end user must be GTP encapsulated in order to be properly processed by the eNodeB.

**Session Management.** Once cell search and radio synchronization procedures are performed, the UE sends an *Attach Request* message to the eNodeB as an initial step in the UE registration procedure (see Figure. 4a). The eNodeB embeds this message within an *Initial UE Message* message, which consists of a unique eNodeB-UE-S1AP-ID assigned by the eNodeB to identify the UEs within the eNodeB over S1-interface, and then forwards it to the MME. The ME host snoops this message and retrieves *eNodeB-UE-S1AP-ID* identifier.

Once the MME acquires IMSI from the *Attach Request* message, it performs several UE authentication and security procedures with the support of HSS. If successful, the MME embeds an *Attach Accept* response message within an *Initial Context Setup Request* message (see Figure. 4b). This message includes SGW GTP tunnel information and a unique MME-UE-S1AP-ID to identify the UEs in MME over S1-interface, and then forwards it to the eNodeB. The ME host retrieves eNodeB-UE-S1AP-ID, MME-UE-S1AP-ID, UE IP address, SGW TEID and SGW IP address information from this message to maintain UE context information for further processing. If the eNodeB-UE-S1AP-ID identifier in *Attach Request* matches to that in *Attach Accept*, the UE is added to the list of UE-associated logical S1-connections in the ME host. The eNodeB then forwards the *Attach Accept* message towards the UE.

The UE now sends an *Attach Complete* message to the eNodeB, which embeds this message in an *Initial Context Setup Response* message that includes eNodeB GTP tunnel information, and forwards it to the MME (see Figure. 4c). The ME host retrieves eNodeB-UE-S1AP-ID, MME-UE-S1AP-ID, eNodeB TEID, and eNodeB's IP address information from this message and updates the UE context with this new information. Completing this step establishes an S1 GTP-U tunnel for the UE to exchange uplink/downlink traffic with the mobile network.

The ME host now has the complete UE context information required to perform stateful decapsulation and re-encapsulation of GTP tunnels for routing UEs IP traffic between eNodeB, EPC and MEC applications.

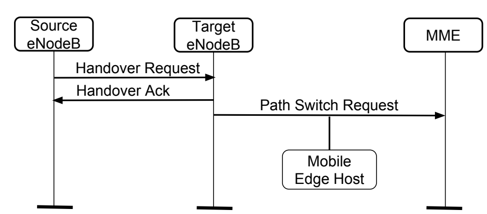



Fig. 5: Path Switch Request.

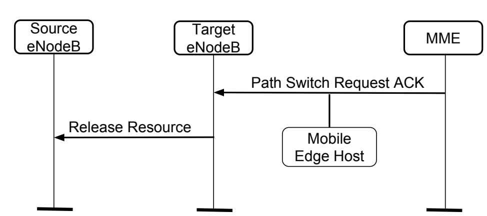



Fig. 6: Path Switch Request Ack.

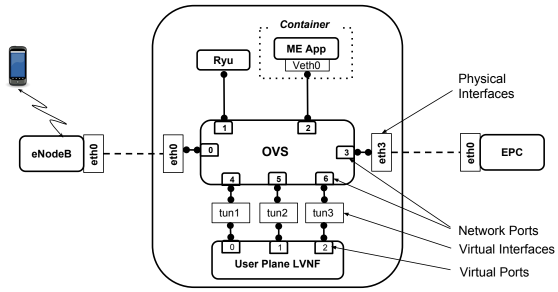



Fig. 7: Mobile Edge Platform.

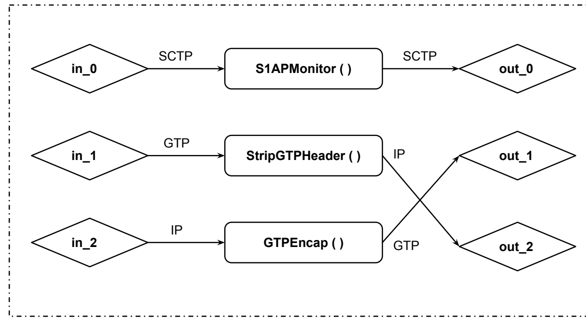



Fig. 8: Stateful User Plane LVNF.

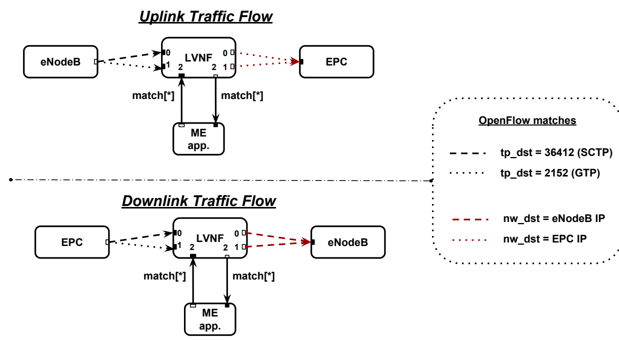



Fig. 9: Stateful User Plane LVNFs chain illustrating traffic flow.

**Mobility Management.** After the UE is attached to the network, it periodically sends serving cell and non-serving cell measurement reports to its eNodeB. Based on the measurements, the source eNodeB may decide to handover the UE to another eNodeB (X2-handover) by sending an *Handover Request* message to the target eNodeB (see Figure.5). The target eNodeB now allocates the necessary radio resources for the UE and responds with an *Handover Ack* message to the source eNodeB. On the other hand, the target eNodeB sends a *Path Switch Request* message to the MME asking it to prepare the new radio bearers. The eNodeB-UE-S1AP-ID of the UE to be handed over, the target eNodeB IP and the target eNodeB TEID information contained in this message are retrieved by the ME host for further processing.

The MME after receiving the *Path Switch Request* message, orders the SGW to establish new radio bearers based on the information received about the target eNodeB GTP tunnel. Once the bearers are established, the MME sends an *Path Switch Ack* message containing eNodeB-UE-S1AP-ID and MME-UE-S1AP-ID identifiers to the target eNodeB (see Figure. 6). The ME host checks if the identifiers belong to the concerned UE and if so, updates the UE context with new eNodeB tunnel endpoint information.

## IV. Implementation Details

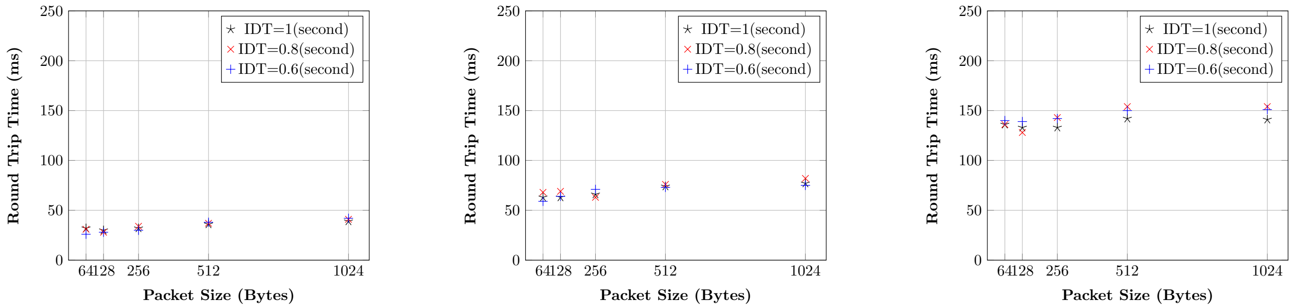
We developed a prototype implementation of *lightMEC* and deployed it over an open source LTE testbed. The RAN part comprises the 3GPP-complaint LTE stack provided by the srsLTE project [21] while as EPC we use nextEPC [22]. Both eNodeB and EPC nodes are running on Intel NUC boxes equipped with dual-core Intel i7 KabyLake CPU, 16GB RAM and 256GB storage. It is to be noted that *lightMEC* is vendor agnostic and can be used with any combination of eNodeB/EPC components.

To deploy the mobile edge host node we use a combination of Intel NUC and Soekris 6501 boards. The LVNF agent, the Ryu controller, and the caching application are all deployed as containers within a dedicated edge node. The lightMANO orchestrator, the 5G-EmPOWER controller, and the kubernetes controller are running on a standard linux machines with no particular hardware restrictions.

Squid [23] is used as ME Application. Squid is an open-sourced caching and forwarding web proxy that primarily supports HTTP, HTTPS and FTP traffic. In this experiment, Squid is configured as a transparent caching proxy i.e., all outgoing HTTP/HTTPS requests are intercepted by Squid and the corresponding responses are then cached, without requiring any changes in the client.

Figure. 7 depicts the internal structure of the mobile edge platform. As it can be seen it consists of a virtual software switch, the Ryu controller, the ME application, and the Stateful User Plane LVNF. This LVNF has three virtual ports with each connected to one virtual interface, which in turn is connected to one network port of the virtual software switch. The packet processing elements of click script used for this LVNF is depicted in Figure.8. Port 0 of LVNF receives LTE control plane traffic (SCTP) which is then passed to the S1APMonitor element that extracts the UE context information as described in Section III. Port 1 of LVNF receives LTE user plane traffic (GTP) which is then passed to the StripGTPHeader element that removes the GTP header and forwards IP packets on output port 2. Port 2 of LVNF receives IP traffic from applications such as squid (caching), proxy servers, etc., which is then passed to GTPEncap element that performs GTP re-encapsulation and forwards GTP packets on output port 1.

Figure.9 illustrates the virtual connection points between the LVNF and other network elements implemented by means of OpenFlow rules in OVS switch. The control plane traffic (SCTP) from eNode/EPC is directed to port 0 of LVNF, the user plane traffic (GTP) from eNodeB/EPC is directed




(a) RTT to a server located in Edge node.

(b) RTT to a server located in France.

(c) RTT to a server located in USA.

Fig. 10: MEC Specific Latency Metrics.

to port 1 of LVNF and any other IP traffic from the ME application is directed to port 2 of LVNF. The traffic matching is performed by the openflow rules configured in OVS switch by Ryu controller. The output traffic from LVNF is forwarded to appropriate destination points depending on the destination IP address of the packets.

## V. Performance Evaluation

To illustrate the potential of our approach, we measure three network performance metrics, namely: MEC specific latency metrics, ME service VNF metrics, and cache latency metrics.

### A. MEC Specific Latency metrics

To assess the difference in latency performances i.e., round-trip-time (the time taken for receiving the response after the initial request was sent by the UE) between the MEC and the non-MEC deployment options, we perform simple ping tests using ICMP messages. Figure.10a plots the average RTT when the ping server was located at the mobile edge node, while Figure.10b and Figure.10c represents the average RTT when the server was located in France and in the USA respectively. The experiments were performed for different packet sizes and for different Inter Departure Times (IDT). As expected, the RTT is significantly lower when the ping is answered from edge node.

### B. ME Service VNF metrics

We performed a set of 10 UE initiated attach events to measure the time taken by S1APMonitor element of the Stateful User Plane LVNF to detect the attach events and to extract the UE context information. The average processing time was $1.4ms$. We then generated IP traffic from the UE to measure the time taken by GTPEncap/StripGTPHeader element of the LVNF to perform encap/decap operations. The average processing time was $30\mu s$ when analysed over a sample of 100 packets.

### C. Cache latency metrics

We performed an experiment to measure the difference in round trip time when the user is served by the caching application running on the mobile edge node, instead of the original web server. The measurements were carried out by

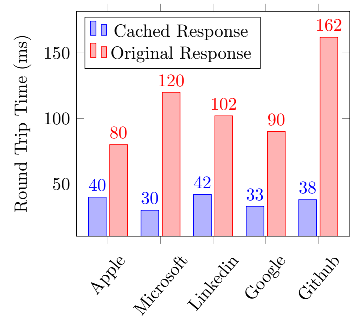



Fig. 11: Caching latency metrics.

making web requests to five popular web pages. The difference in latency is clearly evident from Figure. 11.

## VI. Conclusions

5G is expected to facilitate innovative low-latency applications to enhance the user experience, and it will (mostly) do so by using existing 4G network infrastructure. With video service dominating the mobile network traffic it is becoming challenging for network operators to avoid network congestion in mobile networks and to guarantee QoE for mobile users. This paper aims to address these problems by proposing a novel MEC architecture that leverages SDN and NFV technologies to reduce the barrier for deploying MEC applications and services with existing LTE installations. We proposed a modular MEC architecture, *lightMEC*, by describing the functionality of each element in the node. A proof-of-concept implementation of *lightMEC* was also introduced and validated in a real test-bed with content caching as a use case. As a future work, we plan to extend *lightMEC* platform to support additional MEC services further enabling more innovative applications.

## Acknowledgements

Research leading to these results received funding from the European Union's H2020 RIA Action under Grant Agreement H2020-ICT-644843 (5G-ESSENCE Project).

## References


[1] "European Telecommunications Network Operators' Association." [Online]. Available: https://etno.eu/home/library/etno-reports/

[2] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016 to 2021," in *White Paper*, 2017.

[3] "Docker." [Online]. Available: https://www.docker.com/

[4] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click Modular Router," *ACM Transactions on Computer Systems (TOCS)*, vol. 18, no. 3, 2000.

[5] ETSI, "Mobile Edge Computing (MEC);Technical Requirements," , ETSI GS MEC 002, Mar 2016.

[6] ETSI, "Mobile Edge Computing (MEC);Framework and Reference Architecture," , ETSI GS MEC 003, Mar 2016.

[7] ETSI, "Mobile Edge Computing (MEC); Service Scenarios," , ETSI GS MEC 004, Mar 2016.

[8] ETSI, "Mobile Edge Computing (MEC); Proof of Concept Framework," , ETSI GS MEC 005, Mar 2016.

[9] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing a key technology towards 5G (White Paper)," *ETSI*, 2015.

[10] A. Ahmed and E.Ahmed, "A survey on mobile edge computing," in *Proc. of 10th International Conference on Intelligent Systems and Control (ISCO)*, Coimbatore, Tamilnadu, India, 2016.

[11] H. Li, G. Shou, Y. Hu, Z. Guo, "Mobile Edge Computing: Progress and Challenges," in *Proc. of 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, Oxford, United Kingdom, 2016.

[12] J. Cho, B. Nguyen, A.Banerjee, "SMORE: Software-Defined Networking Mobile Offloading Architecture," in *Proc. of 4th workshop on All things cellular: operations, applications, and challenges*, Chicago, Illinois, USA, 2014.

[13] A. Banarjee, X. Chen, J. Erman, V. Gopalakrishnan, S. Lee, and J. Van Der Merwe, "MOCA: a lightweight mobile cloud offloading architecture," in *Proc. of 8th ACM international workshop on Mobility in the evolving internet architecture*, Miami, Florida, USA, 2013.

[14] S. Wang, G.-H. Tu, R. K. Ganti, T. He, K. Leung, H. Tripp, K. Warr, and M. Zafer, "Mobile micro-cloud : Application classification, mapping, and deployment," *Annual Fall Meeting of ITA (AMITA)*, 2013.

[15] K. Wang, M. Shen, and J. Cho, "MobiScud: A Fast Moving Personal Cloud in the Mobile Network," in *Proc. of 5th workshop on All things cellular: operations, applications, and challenges*, New York City, NY, USA, 2015.

[16] T. Taleb, A. Ksentini, and P. A. Frangoudis, "Follow-MeCloud: When-Cloud Services Follow Mobile Users," *IEEE Transactions on Cloud Computing*, 2016.

[17] J. Liu, T. Zhao, S. Zhou, Y. Cheng, and Z. Niu, "CONCERT: a cloud-based architecture for next-generation cellular systems," *IEEE Wireless Communications*, vol. 21, no. 6, 2014.

[18] ETSI, "MEC Deployments in 4G and Evolution Towards 5G (Whitepaper)," , Tech. Rep., Feb 2018.

[19] "Kubernetes." [Online]. Available: https://kubernetes.io/

[20] R. Riggio, S.N. Khan, T. Subramanya, I.G.R Yahia, D. Lopez, "LightMANO: Converging NFV and SDN at the Edges of the Network," in *Proc. of IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Taipei, Taiwan, 2018.

[21] "srsLTE." [Online]. Available: http://www.softwareradiosystems.com/tag/srslte/

[22] "nextepc." [Online]. Available: http://nextepc.org/

[23] "The Squid Caching Proxy." [Online]. Available: http://www.squid-cache.org/