A practical architecture for mobile edge computing

Conference Paper · November 2017

DOI: 10.1109/NFV-SDN.2017.8169855

CITATIONS
5

READS
333

6 authors, including:

Tejas Subramanya
FBK CREATE-NET
9 PUBLICATIONS **17** CITATIONS
SEE PROFILE

Shah Nawaz Khan
FBK CREATE-NET
26 PUBLICATIONS **44** CITATIONS
SEE PROFILE

Emmanouil Kafetzakis
National Center for Scientific Research Demokritos
34 PUBLICATIONS **207** CITATIONS
SEE PROFILE

Roberto Riggio
Fondazione Bruno Kessler
133 PUBLICATIONS **1,024** CITATIONS
SEE PROFILE

A Practical Architecture for Mobile Edge Computing

Tejas Subramanya*, Leonardo Goratti*, Shah Nawaz Khan*,
Emmanouil Kafetzakis†, Ioannis Giannoulakis‡, Roberto Riggio*
*FBK CREATE-NET, Trento, Italy; Email: {t.subramanya,lgoratti,s.khan,rriggio}@fbk.eu
†Orion Innovations, Athens, Greece; Email: mkafetz@orioninnovations.gr
‡NCSR Demokritos, Athens, Greece; Email: giannoul@iit.demokritos.gr

***Abstract*—Recently, mobile broadband networks are focused on bringing additional capabilities to the network edge. For instance, Mobile Edge Computing (MEC) brings storage and processing capabilities closer to the mobile user i.e., at the radio access network, in order to deploy services with minimum delay. In this paper, we propose a resource constrained cloud-enabled small cell that includes a MEC server for deploying mobile edge computing functionalities. We present the architecture with special focus on realizing the proper forwarding of data packets between the mobile data path and the MEC applications, based on the principles of SDN, without requiring any changes 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 analysed based on a proof-of-concept demonstration for content caching application use case.**

***Index Terms*—5G, Mobile Edge Computing, Content Caching**

I. Introduction

The growth in mobile data consumption has skyrocketed in recent times due to the growing demand for more contextual and immersive mobile experiences like Augmented Reality and Virtual Reality. These applications have requirements in terms of both latency and bandwidth that the traditional mobile network architecture fails to meet. Mobile Network Operators (MNOs) are trying to keep up with these demands by increasing the cellular network capacity through the densification of the Radio Access Networks (RAN). However, there is a strong need to rethink some of the fundamental aspects of network design to enhance mobile user experience. A key problem is the large end–to–end delay between the end users and the requested services as traditional services are deployed in data–centres serving a large number of users in a highly centralized manner. One promising solution is the emerging of the Mobile Edge Computing (MEC) concept which provides a distributed computing environment allowing applications and services to be executed in close proximity to the actual end–users, thus improving both time-to-response and end-to-end latency.

We present a SDN-enabled Mobile Edge Computing solution which places services at the network edge using cloud-enabled small cells. The proposed architecture handles the forwarding of data packets between the mobile data path and the MEC applications. Using SDN concepts, a unified control–plane interface has been developed to retrieve network context information which is subsequently used to selectively steer data traffic from the RAN nodes to the MEC server and vice–versa.

The main contribution of this work is twofold. On one hand, we present a modular SDN–enabled MEC architecture that integrates with the existing LTE systems. We also present a prototype realization of this architecture using a software-based LTE system implementation. On the other hand, we demonstrate the applicability of the proposed architecture to a practical use case, namely content caching.

The rest of the paper is organized as follows. Section II reviews related work on MEC, while Sec. III provides background information on LTE network architecture. The proposed MEC architecture, its communication interfaces and service development framework are discussed in Section IV. Section V demonstrates the proof of concept by analyzing various measurement results for content caching application. Conclusions and future work are discussed in Section VI.

II. State-of-the-Art

The MEC related research has received considerable attention in recent times. In [1], the authors 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 [2]. 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 [3]. In [4] and [5], the authors present different mobile offloading techniques in cellular networks to enable low-latency applications. Other similar concepts to enable edge computing capabilities such as Fog Computing and Cloudlet are presented in [6], along with a detailed comparison.

In this paper, we explore the possibility of leveraging SDN and cloud technologies to enable flexible inter-working between the proposed MEC application framework and the LTE mobile network, through MEC RAN Information Interface. It should be mentioned here that the deployment of our proposed architecture can be achieved without making any changes to the functionality of access and core network.

III. Background: The LTE Mobile Network

In this section, we will briefly introduce the overall LTE network architecture (see Fig. 1) and we will provide an overview of the basic procedures in LTE. At a very high level, an LTE network is composed of two major elements: the RAN and the Evolved Packet Core (EPC). The RAN comprises one logical node, the evolved NodeB (eNodeB), which connects to the User Equipments (UEs). The EPC consists of many logical nodes such as Mobility Management Entity (MME), the Serving Gateway (SGW), the Packet Gateway (PGW) and

978-1-5386-3285-7/17/$31.00 ©2017 IEEE

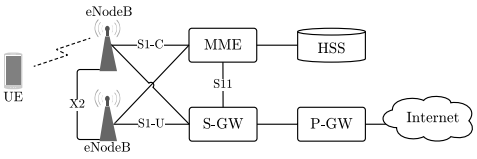


Fig. 1: LTE network overview

the Home Subscriber Server (HSS). Each of these elements are interconnected using standardized interfaces.

UE Attach Procedure. This consists of three phases: (i) IMSI Acquisition: Once a UE establishes a radio link synchronization with the eNodeB, it attempts to attach to the network by sending an 'Attach Request' message to the MME which contains International Mobile Subscriber Identity (IMSI) in it and thus MME obtains IMSI from this message. (ii) UE Authentication and Security Setup: Once MME acquires IMSI, various authentication and security procedures takes place between UE and MME with the help from HSS. Therefore the messages exchanged between them from here on are encrypted and integrity-protected for secure communication. (iii) Session Establishment or Data Bearer Setup: After security mechanisms and once MME obtains UE subscription information from HSS, a default bearer is created for the UE, by establishing tunnels between the eNodeB and the SGW and between the SGW and the PGW. The tunnels are established by exchanging Tunnel Endpoint Identifiers (TEID) for this bearer among eNodeB, MME, SGW and PGW.

UE Data Transfer: Once a UE attaches to the network by creating a default bearer through EPC, it can send/receive data to/from Packet Data Networks using GPRS Tunneling Protocol (GTP) [8]. In uplink, once eNodeB receives the data packets from UE over air interface, it encapsulates the packet in GTP, IP, UDP headers by setting the value of TEID field in GTP header as expected by SGW and forwards the packet towards SGW. The SGW removes the encapsulated headers and adds its own GTP, IP, UDP headers and forwards the packet towards PGW. The PGW removes the encapsulated header and adds its own GTP, UDP, IP headers and forwards the packet towards PDN. The similar process repeats in downlink data transfer from PDN to UE.

UE Detach Procedure: Once UE is done using LTE services, it may initiate a detach procedure by sending a 'Detach Request' message towards MME. The MME removes the UE context by releasing its bearers and sends a 'Detach Accept' response message to UE.

As we will see later, in order to transparently encapsulate and decapsulate GTP traffic so as to reroute user plane traffic towards RAN edge cloud, it is fundamental to correctly obtain the radio network information.

IV. System Architecture

A high–level overview of the SDN–enabled MEC architecture presented in this paper together with its main components is reported in Fig. 2. Our design is modular enough with the following main components: MEC RAN Information Interface (MRI), MEC Application Platform Services, MEC Hosting Infrastructure and MEC applications that leverage the underlying platform through Application Programming Interfaces (API), and is in accordance with the ETSI MEC

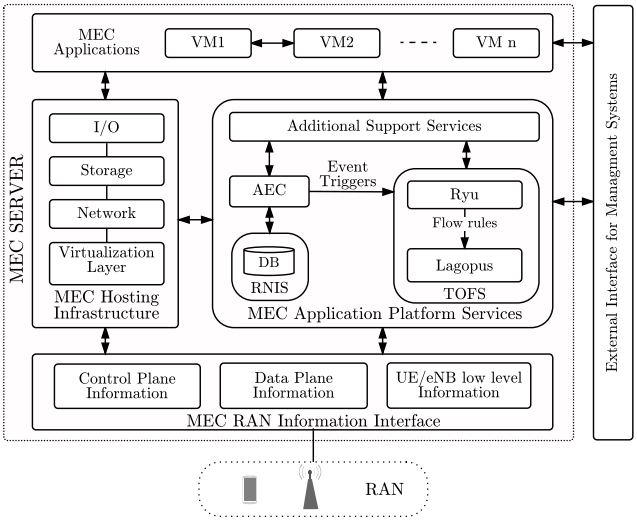


Fig. 2: System Architecture

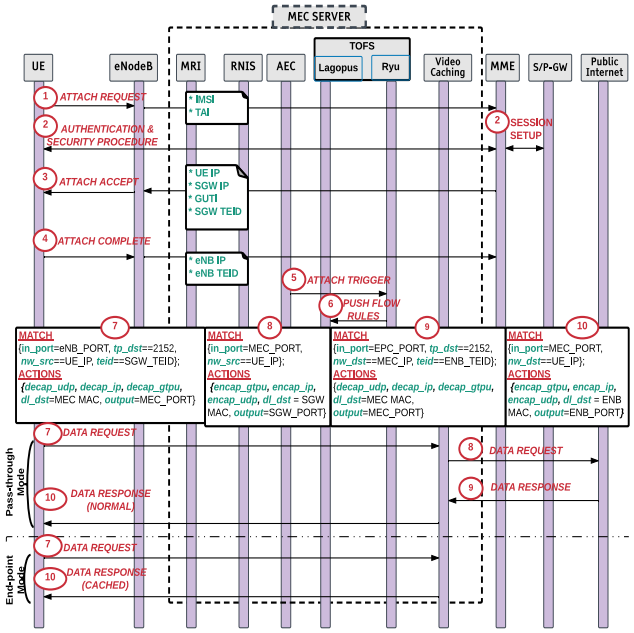


Fig. 3: Sequence diagram of the edge caching application

architecture as presented in [9]. For the moment, our design and implementation is mature enough to identify the benefits it offers in terms of latency, service delivery and user experience, based on the content caching MEC application as an use case.

In the rest of this section we will describe the various components of the proposed architecture. Moreover, we will also describe the role of such components in the implementation of a edge content caching application. The Sequence diagram associated to this application is reported in Fig. 3.

A. MEC RAN Information Interface (MRI)

MRI acts as a RAN–specific interface between MEC applications and the underlying physical or virtual network by

providing a real–time insight into radio network information of the mobile users, performance statistics of the cell and user location awareness. It enables the MEC server to communicate with other network entities, i.e., MME, S-GW, P-GW and eNodeB, through RAN-specific control plane (S1-C and X2-C) and user plane interfaces (S1-U and X2-U).

The MRI is implemented based on Tshark [10] (a terminal based version of Wireshark), that monitors the interfaces between eNodeB and EPC. The three types of information collected by MRI include: (i) Control Plane Information, by capturing and processing signalling messages between eNodeB and MME, (ii) Data Plane Information, by processing data plane packets between eNodeB and SGW, (iii) UE/eNodeB Low-level Information, relevant to Physical, Medium Access Control (MAC), Radio Link Control (RLC) and Packet Data Convergence Protocol (PDCP) layers of UE or eNodeB.

Based on our content caching use case, ① to ④ in Fig. 3, illustrates on the radio network parameters retrieved during UE Attach procedure (IMSI, Tracking Area Identity (TAI), UE/eNodeB/SGW IP, eNodeB/SGW GTP TEID and Globally Unique Temporary Identifier (GUTI)), to make decisions on transparently modifying and rerouting data traffic from mobile network nodes to MEC applications and vice-versa. The implementation of MRI can also be used for detecting UE Detach events, handover events and bearer modification events in the mobile network, which are not illustrated in this paper due to limited space constraints. Some useful parameters collected include Cell Radio Network Temporary Identifier (C-RNTI), UE capability, bandwidth, carrier frequency, Mobility state, QoS Class Identifier (QCI), Allocation and Retention Priority (ARP), Guaranteed Bit Rate (GBR), Maximum Bit Rate (MBR) and Packet Data Network (PDN) identity.

B. MEC Application Platform Services

It is the collection of essential services provided to mobile edge applications hosted within the MEC server.

Radio Network Information Service (RNIS), is basically a collection of radio network information related to users and cells captured by MRI and maintained in a database. The database can be accessed by other services and authorized applications within the MEC server.

Analytics and Event Capture (AEC), analyses the control plane information from RNIS and sends attach/detach/handover event triggers to Traffic Offload Service along with the necessary radio network information as seen in ⑤ in Fig. 3.

Traffic Offload Service (TOFS), is realized together with the Lagopus virtual switch and Ryu SDN Controller, which has been extended to support GPRS tunneling protocol. Additionally, a Ryu application has been developed to provide stateful GTP/IP/UDP header encapsulation/decapsulation service which is required to support MEC functionality. Based on the triggers received from AEC, Ryu controller pushes flow rules onto Lagopus switch using OpenFlow protocol for selectively routing data traffic towards MEC application service chain as seen in ⑥ from Fig. 3.

Based on our use case, four flow rules ⑦ to ⑩ as seen in Fig. 3, are important to understand. For example, in ⑦ Lagopus matches on uplink packets, to check if the traffic is GTP (UDP port=2152), originated from UE and if TEID is same as that of SGW TEID. If so, Lagopus strips outer UDP, outer IP and GTP headers on the matched packets and forwards the traffic towards MEC application service chain by changing the destination MAC address to that of next hop in the path. In ⑧, IP packets are matched to see if the packets arriving from MEC application service chain are originated from UE (source IP=UE IP), and if so, Lagopus performs GTP, IP and UDP header encapsulation and forwards the traffic towards EPC by changing the destination MAC address to that of next hop in the path. Similarly, ⑨ and ⑩ in Fig. 3 matches packets in downlink and performs respective routing actions.

TOFS is supplied to MEC applications in 2 ways:

- *Pass-through mode [⑦ to ⑩].* In our use case, if the user requested content is not available in local edge cache, the caching application VNF can modify the data traffic and pass it back to the original PDN connection (3GPP bearer). In this case, caching application acts as a transparent proxy to the user.
- *End-point mode [⑦ and ⑩].* In our use case, if the requested content from the user is already cached, the data traffic is terminated by the caching application VNF which acts as a server to the user.

C. MEC Applications

MEC applications are an individual or chain of service VNFs belonging to a specific use case, that run on top of the proposed architecture. We implemented content caching application Squid as a VNF.

D. MEC Hosting Infrastructure

It provides virtual resources for computation, storage and networking within the MEC server. It also provides connectivity service among a chain of applications, services, 3GPP network, local networks and external networks, by routing traffic among them.

V. Performance Evaluation

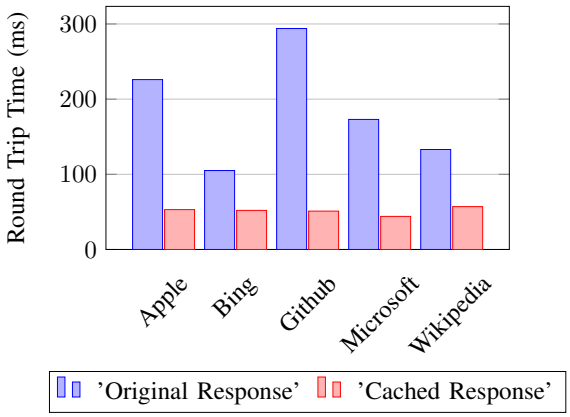
The experiments were performed using Open Air Interface (a software-based LTE system covering 3GPP complaint full LTE protocol stack for both eNodeB and EPC). We used commercial LTE smart phones with programmable SIM card in them as mobile users. We used a low-power, low-cost, advanced communication computer (Soekris net6501) as a light-weight MEC server that includes Ryu controller, virtual Lagopus switch and a caching application (i.e., Squid). We measured the performance of individual components within the MEC server in terms of processing time, latency and responsiveness, and illustrated the potential for reducing average round trip time (RTT) and overall backhaul traffic by caching content at the MEC server based on our approach.

A. MEC RAN Information Interface

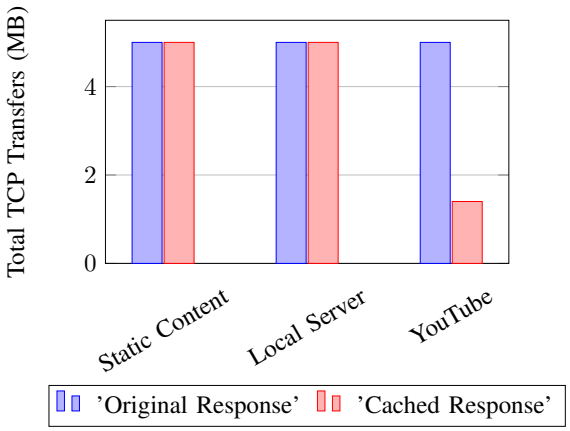
We performed a series of 10 UE initiated attach and detach events to measure the time taken by MRI to detect the UE events. The average processing time to detect a UE attach event is 1.4ms and to detect a UE detach event is 1ms, without considering the Tshark packet parsing time (additional 600ms).

B. MEC Application Platform Services

Once the UE attaches successfully to the network, we measure the total time taken by the AEC service and the Ryu controller to install/uninstall traffic rerouting flow rules onto Lagopus. The mean value for the 10 UE attach events is 16ms with a variance of 5.55ms and standard deviation of 2.35ms.



(a) Round Trip Time.



(b) Cache Hits.

Fig. 4: Evaluation Results

C. RTT and Backhaul Congestion Improvement

The first experiment focuses on the different response times the user gets when served by caching application in the MEC server, instead of directly from the origin Web server. The comparison among them is seen in Fig. 4a, which is obtained by making a web request using Curl (a command-line tool) to five popular web pages, from the LTE UE terminal (Huawei USB dongle) located about 2 meters to the serving eNodeB. The measurements have been repeated for 10 successive times. The difference in average RTT is significantly lower with caching, mainly because of the evasion of core network and web server processing delay in the backhaul.

The second experiment focuses on the amount of web content that can be cached to serve the users directly from local cache instead of the origin web server. The analysis is shown in Fig. 4b, which is based on three web request operations performed by the LTE UE terminal with Squid as the caching application. (i) User makes a GET request to receive static content such as html pages, css scripts, javascripts, images and binaries. The response is completley cached by Squid, and any identical requests later on are served directly from the local cache. (ii) Similarly, when a user requests a video hosted on our local web server (Apache), the entire video is cached by Squid. (iii) However, when a user requests a video from some of the major media sites like YouTube or Google Videos, Squid alone cannot cache the entire video, since this traffic does not behave like simple HTTP web pages [11]. In such a situation, Squid uses custom video caching solutions to cache only the static content from these websites (25%). As seen in Fig 4b, in all three cases, mobile backhaul traffic is significantly reduced with caching. It is important to note that, in this paper, we do not focus on the performance of different caching algorithms.

VI. Conclusion

We leveraged SDN and cloud to present a ETSI complaint modular MEC architecture for LTE mobile networks. The architecture proposes three important components: MRI, to communicate with the underlying radio network; MEC Application development Platform, that provides a number of value added services and APIs to application developers; and MEC applications, for offering new personalized services to specific customers. We also analyze the video content caching use case as a proof of concept for the proposed architecture by transparently intercepting and rerouting the traffic towards MEC applications. In future, we plan to extend the proposed architecture with additional value added services such as policy control and service registry to support more low-latency mobile edge applications.

Acknowledgements

Research and development leading to these results has received funding from the European Framework Program under H2020 grant agreement number 671596 (SESAME).

References

[1] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, *"Mobile edge computing a key technology towards 5G"*, ETSI White Paper, vol. 11, 2015.
[2] A. Ahmed and E.Ahmed, *"A survey on mobile edge computing"*, in Proc. of IEEE ISCO, 2016, Coimbatore, India.
[3] H. Li, G. Shou, Y. Hu, and Z. Guo, *"Mobile edge computing: progress and challenges"*, in Proc. of IEEE MobileCloud, 2016, Oxford, UK.
[4] J. Cho, B. Nguyen, A.Banerjee, *"SMORE: Software-Defined Networking Mobile Offloading Architecture"*, in Proc. of ACM ATC, 2014, Chicago, IL, USA.
[5] 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 the ACM MobiArch, 2013, Krakow, Poland.
[6] K. Dolu, S.K. Datta, *"Comparison of Edge Computing Implementations: Fog Computing, Cloudlet and Mobile Edge Computing"*, in Proc. of Global IOT Summit, 2017, Geneva, Switzerland.
[7] 3rd Generation Partnership Project, *"Radio Access Network; Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Architecture description"*, 3GPP TS 36.401, 2015.
[8] European Telecommunications Standards Institute, *"General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U)"*, ETSI TS 129 281, 2014.
[9] European Telecommunications Standards Institute, *"Mobile Edge Computing (MEC); Framework and Reference Architecture"*, ETSI GS MEC 003, 2016.
[10] Wireshark, *"Tshark Manual Pages"*. [Online]. Available: https://www.wireshark.org/docs/man-pages/tshark.html.
[11] Squid, *"Caching Dynamic Content"*. [Online]. Available: http://wiki.squid-cache.org/ConfigExamples/DynamicContent