SDEC: A Platform for Software Defined Mobile Edge Computing Research and Experimentation

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; E–Mail: giannoul@iit.demokritos.gr

***Abstract*—In recent times, mobile broadband networks are focused on bringing different capabilities to the edge of the mobile network. Mobile Edge Computing (MEC) addresses this issue by placing the compute and storage resources closer to the Radio Access Network (RAN), with an aim to reduce end-to-end latency, ensure better service delivery, and offer improved user experience. In this work, we propose SDEC - a Software Defined mobile Edge Computing platform, that provides mobile edge services to be consumed by authorized mobile edge applications and thus enabling the delivery of mission-critical applications over the mobile network. In this demonstration, we will illustrate the feasibility of SDEC platform by using video caching (static content) MEC application as an example use case.**

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

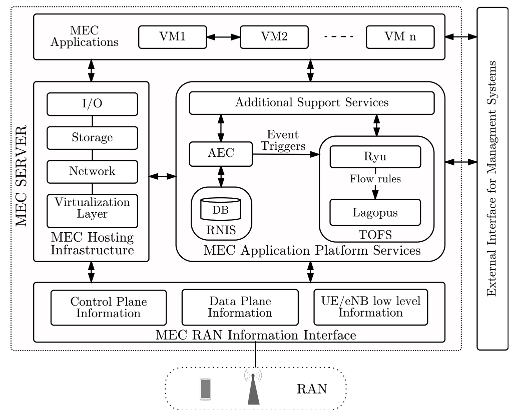

Fig. 1: SDEC Architecture Overview

I. Introduction

With the evolution of mobile networks from 2G through to LTE and onwards to 5G, user expectations on service experience have risen in tandem with wireless data speeds. Traditional network architectures cannot meet such high requirements of ultra-low latency and ultra-high network bandwidths that are necessary to achieve best user experience and faster service innovation. A key architectural reason for such large end-to-end network delay is the deployment of end services in large data centres, that serve a large number of users connected to a number of Radio Access Networks, in a highly centralized fashion. Therefore to support services with strict latency requirements it is necessary to move network functions, service processing functions and third party applications closer to the mobile network edge.

The design of SDEC addresses this need by placing intelligence at the network edge, i.e. close to cloud–enabled small cell integrating radio access with computing capabilities, with the overarching goal of allowing the deployment of novel applications and services closer to the actual end–users [2]. The proposed architecture, realizes the proper forwarding of data packets between the 3GPP mobile data path and the MEC applications, without requiring any changes to the functionality of existing mobile network nodes. This is achieved by adopting the concept of SDN and by developing a unified control-plane interface to retrieve real-time radio network information, based on which data traffic is steered from the radio access nodes to the MEC server. Section II describes the SDEC architecture in detail, while section III demonstrates video content caching application as an use case, to illustrate the benefits of inheriting SDEC framework for other research activities.

II. SDN-based MEC Architecture

Fig. 1 shows an overview of the SDEC Architecture and its main components in the context of an LTE network [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 exposing up-to-date radio network information regarding the current radio network conditions along with the context of UEs associated to the radio node. It enables the MEC server to communicate with other network entities (Mobility Management Entity (MME), Serving Gateway (S-GW), Packet Gateway (P-GW) and eNodeB) through RAN-specific control plane (S1-C & X2-C) and user plane interfaces (S1-U & X2-U). The MRI is implemented based on Tshark - a terminal based version of Wireshark, that monitors the S1 and X2 interfaces. The three types of information collected by MRI include the following:

- **Control Plane Information**, by capturing and processing signaling messages between eNodeB and MME (S1-C and X2-C).
- **Data Plane Information**, by processing data plane packets between eNodeB and SGW (S1-U and X2-U).
- **UE/eNodeB Low-level Information** include parameters relevant to Physical, Medium Access Control (MAC), Radio Link Control (RLC) and Packet Data Convergence Protocol (PDCP) layers of UE and eNodeB.

Some of the useful parameters collected include 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/to MEC applications.

B. MEC Application Platform Services

It provides the fundamental middleware services to support novel MEC applications inside the MEC server.

Radio Network Information Service (RNIS), stores the radio network information in a database structure and includes information about the users and the radio cells captured by the MRI. The stored radio network information can be accessed by other services and authorized applications inside the MEC server.

Analytics and Event Capture, analyzes the control plane information from RNIS and sends attach/detach/handover event triggers to Traffic Offload Service along with the necessary radio network information.

Traffic Offload Service (TOFS), is realized together with 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. For a particular MEC application service chain requiring traffic redirection towards a service VNF inside the MEC server, the Ryu controller, based on triggers received from RNIS, pushes flow rules onto the Lagopus virtual switch using OpenFlow protocol. As seen in Fig. 2, TOFS is supplied to applications in the following two ways:

- *Pass-through mode*, where the user plane traffic is passed to MEC application VNF, which can modify and then send it back to the original Packet Data Network connection (3GPP bearer).
- *End-point mode*, where the traffic is terminated by the MEC application VNF, which acts as a server.

C. MEC Applications

An MEC application is a service VNF that can interact with the underlying framework and provide service to the end users. In this demonstration, a transparent caching application (Squid) [4] is implemented as a service VNF. We were also able to cache static content from YouTube using Store URL rewriting helper [5].

D. MEC Hosting Infrastructure

It includes a virtualization layer for providing compute, storage and network resources for the mobile edge applications. It also provides connectivity service among a chain of applications, services, 3GPP network, local networks and external networks, by routing traffic among them.

III. Demonstration Overview

Fig. 2 shows SDEC prototype and demonstration setup. We use Huawei LTE dongle with programmable sim card as the mobile user, one laptop to run the LTE small cell (OpenAirInterface-OAI) together with Software Defined Radio (USRP B210) and one laptop to run the mobile core network components (OAI). We use a compact, low-power, low-cost, advanced communication computer (Soekris net6501) as a light-weight MEC server that includes Lagopus switch, Ryu SDN controller and a caching application (Squid). The end-to-end architecture of SDEC is completely implemented based on open-source technologies and thus can be useful for other research activities.

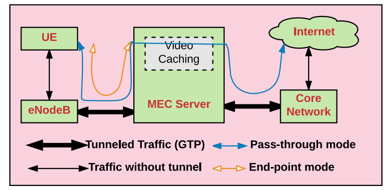

Fig. 2: SDEC prototype and demo setup

With respect to Fig. 2, the demo scenario is as follows:

① *Route:* UE attaches to the mobile network (OAI) by establishing a default EPS bearer. After the successful exchange of control messages, Ryu inserts open flow rules into Lagopus based on the triggers received from RNIS.

② *Request:* User makes a URL request in YouTube to watch a video or a simple web request to any website. For the very first time, the data traffic takes the path of Pass-through mode, since, Squid acts as a transparent caching proxy. The response is cached by the Squid application.

③ *Identical request:* When the user again makes a request to the same URL, the user traffic takes the path of End-point mode, since, the Squid application responds with the cached content. However, the user is completely transparent to the entire process.

④ *Statistics:* We will look at the improvements in Round Trip Time, number of cache hits or cache misses, total bytes of cache hits or cache misses, that helps in realizing the benefits of SDEC in addressing the issues of network latency and backhaul network congestion.

Acknowledgment

The research leading to these results has been performed in the scope of the H2020 5G-PPP project [1]. This project has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement No 671596.

References

[1] Small cElls coordinAtion for Multi-tenancy and Edge services. (2017). Retrieved from http://www.sesame-h2020-5g-ppp.eu/
[2] European Telecommunications Standards Institute, *"Mobile Edge Computing (MEC); Technical Requirements"*, ETSI GS MEC 002, 2016.
[3] European Telecommunications Standards Institute, *"Mobile Edge Computing (MEC); Framework and Reference Architecture"*, ETSI GS MEC 003, 2016.
[4] Squid, *"Transparent Caching"*. [Online]. Available: http://wiki.squid-cache.org.
[5] Squid Features, *"Store URL Rewrite"*. [Online]. Available: https://wiki.squid-cache.org/Features/StoreUrlRewrite.