A Mobility Management Architecture for Seamless Delivery of 5G-IoT Services

Balasubramanian, Venkatraman; Zaman, Faisal; Aloqaily, Moayad; Ridhawi, Ismaeel Al; Jararweh, Yaser; Salameh, Haythem Bany

DOI
10.1109/ICC.2019.8761658

Publication date
2019

Document Version
Final published version

Published in
ICC 2019 - 2019 IEEE International Conference on Communications (ICC)

A Mobility Management Architecture for Seamless Delivery of 5G-IoT Services

Venkatraman Balasubramanian*, Faisal Zaman†, Moayad Aloqaily‡, Ismaeel Al Ridhawi§, Yaser Jararweh¶, and Haythem Bany Salameh‖

*Delft University of Technology (TUD), Netherlands, V.balasubramanian@tudelft.nl
†Ciena Networks, Ottawa, ON, Canada, Fzaman@ciena.com
‡Canadian University Dubai, UAE, maloqaily@ieee.org
§University of Ottawa, Ottawa, Ontario, Canada, K1N6N5, Ismaeel.AlRidhawi@uottawa.ca
¶Duquesne University, Pittsburgh, PA, USA, Jararwehy@duq.edu
‖Yarmouk University, Irbid, Jordan, Haythem@yu.edu.jo

***Abstract*—Mobile Edge Computing (MEC) and Network Slicing techniques have a potential to augment 5G-IoT network services. Telecommunication operators use a diverse set of radio access technologies to provide services for users. Mobility management is one such service that needs attention for new 5G deployments. The QoS requirements in 5G networks are user specific. Network slicing along with MEC has been promoted as a key enabler for such on-demand service schemes. This paper focuses on radio resource access across heterogeneous networks for mobile roaming users. A unified service architecture is proposed enabling seamless handover between a 5G (New Generation Core) service and a 4G (Evolved Packet Core) service via the network slicing paradigm. An identifier-locator (I-L) concept that allows active source-IP sessions is used to handle the seamless hand-over. Signaling costs, service disruptions and other resource reservation requirements are considered in the evaluation to assure that profit for mobile edge operators is achieved. Simulation experiments are considered to provide performance comparisons against the state-of-the-art Distributed Mobility Management Protocol (DMM).**

***Index Terms*—Mobile Edge Computing, 5G, Network Slicing, Internet of Things.**

I. Introduction

In traditional cellular networks, event triggered handovers are controlled by the base station, such that a control signal is sent to the user's device directing it to report its network status to the serving base station constantly [1]. In [2], the authors compare and define the challenges of the service-oriented core for the Standalone (SA) mode against the recent release of the 5G Non-Standalone (NSA) mode. While NSA has a 4G anchor that allows seamless transitioning from 4G to 5G, SA deployment mode requires some integration techniques with the Evolved Packet Core (EPC) to allow mobility access in 4G and 5G. This leads to the need for on-demand service customization to address interoperability and session continuity at the time of mobility. For providing such a customization, network slicing was proposed.

Network slicing was designed as a major enabler for on-demand customized services in resource-constrained networks, allowing for optimal network resource utilization in both static and mobile environments [3]. Slicing a network allows several on-demand tailored services [4] to be provided with the same physical network, in which, resources can be dynamically allocated to logical slices based on QoS demands. Network Function Virtualization (NFV) and Software Defined Networking (SDN) are two main elements that enables programmatic control of resource allocation [5]. NFV [6] replaces the functionalities of Evolved Packet Cores (EPC) (e.g. Mobility Management Entity (MME)) with Virtual Machines (VMs) running on off-the-shelf commercial servers. Moreover, the servers utilize VMs to perform Radio Access Network (RAN) activities.

The backward compatibility of the Standalone 5G still continues to cause issues for mobile users. In pursuit of tackling this problem, the Mobile Edge Computing (MEC) paradigm was introduced in [7]. Implementations that exploit the association of MEC servers with RAN to enable proactive computation are described in [8]. The uncertainties of having the knowledge of future channel conditions and resource availability prevents seamless service guarantees for mobile users. The integration of MEC within 5G networks has enhanced service delivery and reducing latency down to 1ms [8]. However, there are no clear solutions for seamless session continuity among heterogeneous slices with diverse Radio Access Technologies (RATs). In this paper we investigate the mobility management problem in the context of 5G networks. Rather than making expensive Next Generation Core (NGC) level infrastructure changes, we employ a novel software design solution.

A. Problem Overview and Scenario Development

We address the common scenario depicted in Figure 1. IoT applications require seamless service delivery to consumers [9]. As shown in the figure, service request from a particular radio access network to another, changes according to user mobility (e.g. from a slice supported by a 4G RAT to a 5G RAT). A 5G slice for IoT service deployment in a business district will have much heavier channel loads than those serving suburban residential areas. Traffic demands vary as users move from a slice that serves a business district to a residential area served by an LTE slice; such that, the chang-

*author was visiting TUD during this work ing traffic demands causes extra signaling overhead. The Distributed Mobility Management (DMM) protocol standardized by Internet Engineering Task Force (IETF) supports mobility management. However, DMM is not capable of perceiving a complete view of the changing network topology which results in continuous signaling overhead for packet tunneling and flow management as the user moves. Additionally, DMM does not always guarantee per-flow mobility support in such environments.

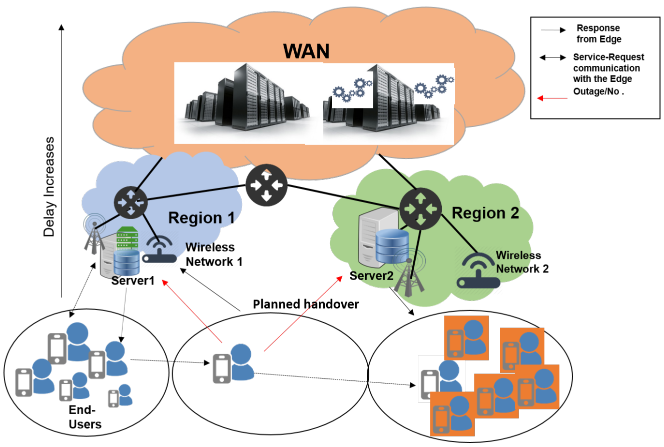

Fig. 1: Scenario Overview

B. Contributions

- We design an architecture for inter-slice communication initiated over an MEC platform. We employ a novel software design specifically tailored for 5G which we call "*Connection Mode as a Service (CMaaS)*". This module acts as a cloud broker that aggregates virtualized radio resources available at the edge.
- To achieve seamless handover, we propose a controller logic that runs inside the MEC server and identifies the source-IP. This creates usable connections with an exogenous network by locating other network agents. We determine the packet delivery ratio, average achievable latency at the time of the handover and evaluate service disruptions.
- Finally, we compare the IETF DMM protocol against our proposed solution. Results show over 12.5% improvements in bandwidth utilization for varying network traffic intensities. This in essence makes CMaaS a promising solution for heterogeneity issues in 5G-IoT.

The remainder of this paper is structured to define the novelty of the framework through a brief discussion of related work in Section II, followed by the system architecture and the mobility management scheme in Section III. In Section IV, we evaluate the performance of the proposed algorithm with the state-of-the-art DMM protocol. Finally, Section V concludes this work and proposes potential future investigations.

II. Background and Related Work

A. Background

The research on mobility management protocols focuses on two main entities (a) Home Agent (HA) (E.g. in Mobile IPv6 (MIPv6)) and (b) Local Mobility Anchor (LMA) (e.g. Proxy Mobile IPv6 (PMIPv6)) [10]. MIPv6 and PMIPv6 enables the identification of Mobile Nodes (MNs) when away from their home location. Such an approach introduces problems in terms of scalability and reliability as the central management entity acts as a single point of failure [11]. This in turn affects the overall network performance. To counter these issues, distributed management approaches were proposed, such as the Distributed Mobility Management protocol introduced by IETF [12], allowing for the distribution of the mobility management function. DMM also offers key features such as dynamic mobility features (per prefix granularity) and flow anchoring. The complexity here arises as the flows are anchored at the mobility access router. Therefore, prefixes when changing its point of attachment follow the access router's defined path and are routed via the tunnel.

In 5G scenarios, handover management does not only consider user mobility. Due to network slicing, a physical network maybe divided into logical networks with diverse Radio Access Technologies (RAT). Hence, an operator has to deal with a diverse set of RATs. For example, while transitioning from a 4G slice to a 5G slice, an operator has to adapt to the user requirements, which are bound to change. The operator will have to cater to the changing needs. To this end, there are SDN solutions such as [5], but are still vague, and do not show how such deployments would assist in different modes of operation in 5G. Additionally, handover management of such solutions affect the overall latency and bandwidth, due to inconsistent signaling and extra resource requests which leads to quality of service (QoS) degradation in the last mile. Thus, to maintain service continuity, the MEC paradigm provides a last mile specific QoS maintenance scheme [13]. The deployments of MEC in the context of 5G, such as [14], have never taken this holistic view of the heterogeneity created in the network due to diverse RATs. The proposed work in this paper is one of the first to design an MEC-specific middle-ware for seamless service continuity in 5G environments. In [15], the authors introduced the problem of data and service management in densely crowded topology.

B. Related Work

In [16], the authors examine an SDN and DMM based approach. The key advantage of this solution is the avoidance of infrastructure-deployment costs of mobility-related modules at the access. The solution also achieves orthogonality of the control and data-planes. In this work the scalability of SDN with DMM was evaluated, however, it does not show bandwidth utilization penalties and effects of inter-slice heterogeneity. A 5G low latency network slice that offers low latency services using the closest network edge node was presented in [11]. A prototype implementation verified the performance of the mobility anchor during each handover, as well as the required gateway relocation. Song et al. [17] proposed a solution that mainly focused on the SDN control procedures during network slicing. They proposed a scheme that can trigger handover in advance, and effectively enhance the handover success rate.

Bilen et al. [18] propose an SDN-based mobility and available resource estimation strategy to solve the handover delay issue in ultra-dense 5G networks. The solution provides an estimate of the neighbor eNB transition and available resource probabilities using a Markov chain formulation. Optimal eNBs are selected and assigned to mobile nodes virtually using OpenFlow tables. The authors show that their solution reduces handover delay by up to 52%, whereas handover failure is reduced by up to 21%, when compared against conventional handover approaches. Another solution which was proposed by Arshad et al. [19], considers handover management in dense 5G networks. The authors propose a smart handover solution which uses topology awareness and user trajectory estimation. The solution accounts for the location of the trajectory within the cells when taking the handover skipping decision. Simulation evaluations were considered to compare their solution against an always best connected scheme to show its effectiveness in terms of data rate gains.

None of the above mentioned works provide signaling cost evaluations. Some researchers have focus on the the architecture for the 5G application scenario for mobility management. However, our work differs from this in terms of provisioning mechanisms between heterogeneous network slices at the time of mobility and the cloud edge reservation strategy. The proposed CMaaS technique in this paper uses MEC to provide a possible mobility solution for 5G networks.

III. System Architecture

Our framework is divided into two core components namely the CMaaS controller and the S/P-GW module as depicted in Figure 2. The Edge entity is collocated with the 5G new radio, labeled as **Next Generation Core (NGC)** in the figure. It communicates via its software interface to the MEC. The **Evolved Packet Core (EPC)** user plane functions are shifted to the MEC such that a tailored service setup of Service Gateway (S-GW) and Packet Gateway (P-GW) together result in offloading a major portion of the back-haul signaling to the edge via the **S/P-GW**. The MEC acts like a cloud-broker that makes resource assignments based on requests generated.

1) The **CMaaS** control module runs the control logic. The controller creates the mapping between the subscribed 5G users and their location updates that are stored for creating a virtual tunnel with the LTE radio (eNB). The Mapping and Resource Management DB contains a tuple $< NodeID,\ Flow\ ID\ and\ \{QoS_requirements\} >$ that conveys the required information to the controller at the time of movement from the source location. The CMaaS is a global controller that retrieves information from the database for every $packet\_in$ information that it receives. A look-

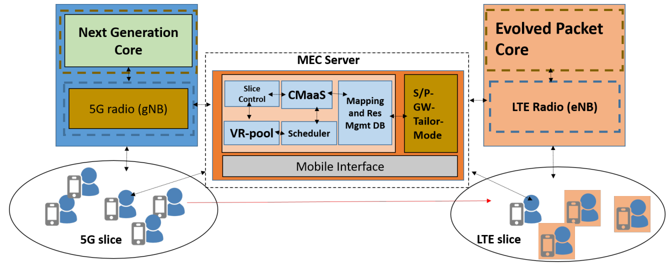

Fig. 2: System Architecture

up table that corresponds to those who have subscribed to CMaaS is maintained. The benefit of this service is that there is no extra cost for the Standalone (SA) mode when a new infrastructure is required. Instead, a software update of the standalone service that will serve the requests of a particular subscription can be performed in the CMaaS. Once subscribed to CMaaS, the CMaaS controller acts as the anchor point for information exchange and mapping. The sequence of steps followed in Figure 3, indicate how the signaling overheads are managed at the MEC. The *red* boxed outline shows the radio level connections that are followed in legacy networks (only the most integral states have been shown), and the *blue* boxed outline presents our proposed approach for slice-specific signaling. Leveraging the open-flow protocol benefits our framework, as the edge controller now becomes the flow manager.

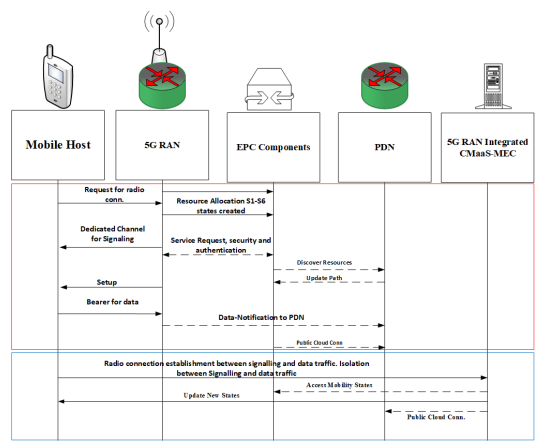

Fig. 3: Signaling procedure with CMaaS

Slice Control and VR-Pool - The VMs that are distributed in the edge run the slice control logic such as retrieving virtual resources (VR) *viz* for network and compute, instance availability and preparing a scheduling list. In scenarios when 5G users are bound to demand the highest available QoS in an LTE environment, the CMaaS provisioning module manages the available resources by following a break-before-make strategy and making a QoS readjustment. The edge server makes the reservations through the controller for future resources. The path computation is performed here to maintain location updates. The **MEC scheduler** presents a centralized pool of virtualized radio resources that are controlled by the CMaaS module. The next sub-section shows how the resource allocation constraints are formulated according to operator profit. It will be shown in the evaluation section, how service disruptions and mobility are related through the session to mobility ratio parameter.

2) **S/P-GW Tailor-Mode Service** - The handover procedure is highly dependent on the virtual resource (VR) pool that is maintained by the edge cloud entity. At the other end, the S/P-GW tailor-mode service performs data-plane forwarding to ensure data is transmitted to the targets. Moreover, the radio has the necessary information for base-band processing. If there are multiple slices, the CMaaS controller on each slice performs handover through cooperation. The handover operations are carried out and the new path reserved in the resource management DB is sent to the node. The flows that are involved in the transition are updated in the forwarding table at run-time allowing for a continuous operation of the service. From Figure 3, it is clear that the mobility states maintained in the EPC components are retrieved by the CMaaS which is executed in the MEC collocated with the 5G RAN.

The CMaaS controller controls the entire network, by constantly receiving information from the slice control units and the schedulers. The **Mapping and Resource management DB** stores all the necessary location information. It is always aware of services that have been allocated, and continuously monitors resources. Additionally, the paths through which the packets are routed are calculated by the CMaaS controller. Herein, we leave the path calculation for future work. Now, we formulate the operator profit based on the above constraints.

A. Formulation

The objective of this work is to maintain adequate QoS levels under mobility. This led us to investigate link resource utilization, by breaking down the path taken by the packets from the user to the edge.

Consider the following: assuming there is an aggregation site where all the information from other slices reside (in our case CMaaS module), including link requirements defined as $2-tuple$ (b^j, d^j). Additionally, the link $e \in E$ is taken from a set $1, 2, 3...e$, with bandwidth b^j and latency requirement d^j from the slice to the MEC where $j \in J$ is the requirement. When a request arrives at the MEC, the MEC provider decides whether to acknowledge or reject it. The decision is based on the availability in the resource pool and the requirements specified by the slice. Essentially, the link with maximum profit for the mobile operator is selected, such that:

$$max \sum_{i \in I} \sum_{j \in J} (x_j^e p_i^e - r_i^c b_j^e) \ ; \ \forall c \in C; \forall e \in E \qquad (1)$$

Typically, this relation follows that higher the selling price when compared to the cost price, higher will be the profit. Consider an MEC resource that serves virtual requests $c \in C$, where we define a decision variable r_i^c for every request i, such that:

$$r_i^c = \begin{cases} 1, & \text{if } \ the\ MEC\ c\ serves\ the\ request\ i \\ 0, & \text{otherwise} \end{cases} \qquad (2)$$

We define a binary variable p_i^e, such that

$$p_i^e = \begin{cases} 1, & \text{if } \ for\ a\ request\ i\ that\ takes\ the\ link\ e \\ 0, & \text{otherwise} \end{cases} \qquad (3)$$

The objective of link selection is to choose the path that best meets the requirements of the slice. Every link has an associated cost x^e dollars/mbps/month that is given by the operator, and link usage needs to be optimized to maximize profit for the operator for a link e with associated cost of link b_j^e dollars/mbps/month. We assume each link has a bandwidth capacity $B^e(t)$ at time t for the chosen link e. This objective is subject to the following constraints: The bandwidth usage B_j for every link must be based on the residual bandwidth available, such that:

$$\sum_{i \in I} \sum_{j \in J} r_i^c B_j \leq B^e(t) - B^e(t-1); \ \forall e \in E, c \in C \qquad (4)$$

For each link, the acceptable delay depends on the chosen link having lower delay than the requirement d^j,

$$\sum_{i \in I} \sum_{j \in J} p_i^e d_j^e \leq d_j; \ \forall e \in E \qquad (5)$$

Moreover, every link satisfying the above objectives is considered to provide a specific session to mobility ratio (SMR) given by $\frac{\lambda_p}{\sigma}$, where λ is the average packet arrival rate and σ is the mobility rate [20]. As this is an NP-hard problem, we use heuristics for sensitivity analysis of our proposed solution.

IV. Performance Analysis

We divide the performance analysis section into two parts: 1) Network Analysis, and 2) Sensitivity Analysis of CMaaS.

A. Network Analysis

Network parameters such as bandwidth utilization and hand-off latency are compared against the legacy DMM protocol through simulations. Using Omnet++ [21] on a PC with core 2 duo CPU, 4 GB RAM and Ubuntu OS, we generate a custom topology with the controller logic running on the PC. The network slices are created as individual clusters of nodes of 20, 40, 80 incrementing up to 320. The slice with the highest density contains 320 nodes. As virtual machine instances provide isolation between non-paired instances, we maintain the characteristic features of the slice. Each cluster is modeled as an instance. Furthermore, to induce mobility, we use a generic random-way point model and calculate the hand-off latency while making a node handover from one slice (cluster) to another. We keep an average arrival rate λ of 1/600 and an average mobility rate of σ, varied to get the SMR.

Results depicted in Figure 4 show that signaling costs are relatively low. Since future path conditions are known, no new path requests need to be configured. This, in essence, abolishes some provisioning overhead. Additionally, Figure 5 shows the traffic intensity (λt Erlangs, t is the holding time) in terms of the percentage of traffic

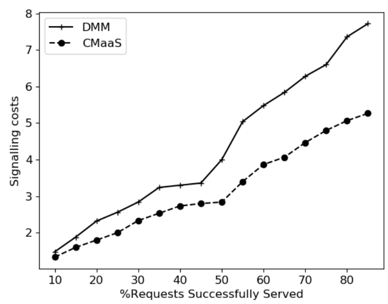

Fig. 4: Signaling Costs

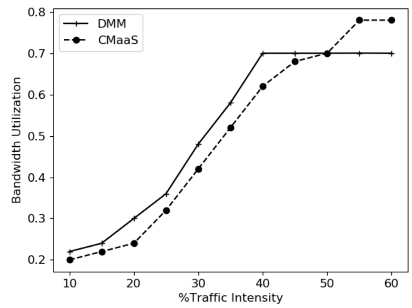

Fig. 5: Bandwidth Utiliz. vs Traffic Intensity

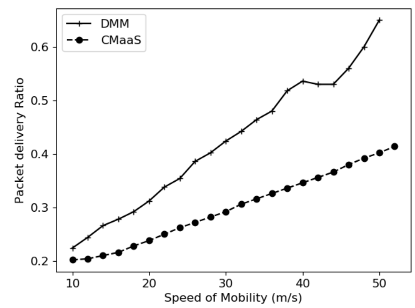

Fig. 6: Packet Delivery Ratio vs Mobility

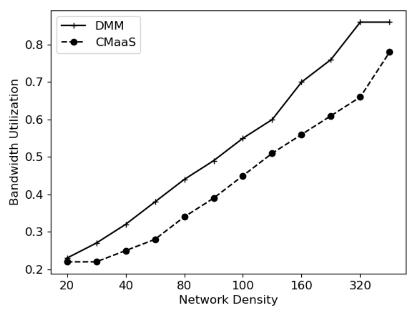

Fig. 7: Bandwidth Utiliz. vs Network Density

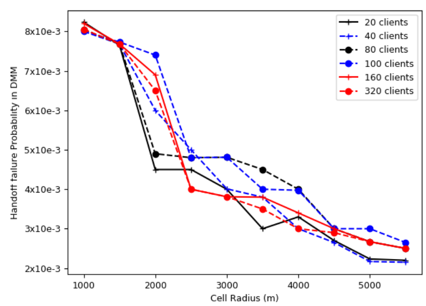

Fig. 8: Handoff Failure Probability in DMM

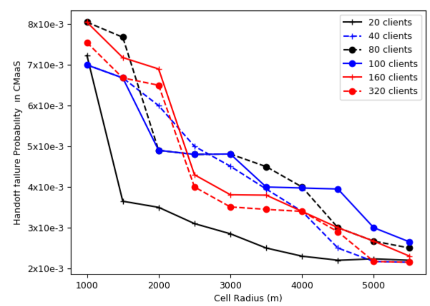

Fig. 9: Handoff Failure Probability in CMaaS

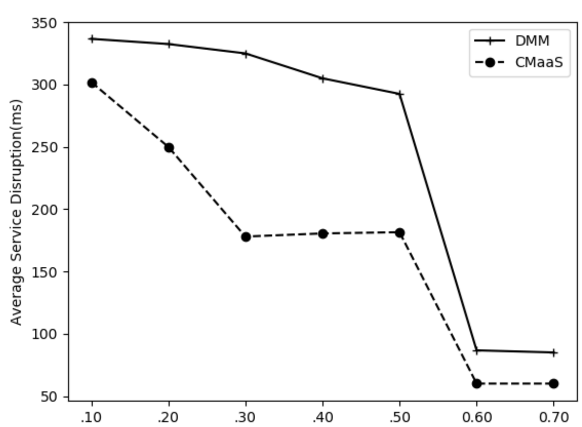

Fig. 10: Service Disruptions to SMR

($\frac{Total\ accepted\ requests}{Total\ Requests}$) generated based on different application scenarios. Results reveal that DMM achieves bandwidth utilization ($\frac{Avg.\ Throughput\ Rate}{Clock\ Rate}$x100%) saturation even before CMaaS. This is one possibility where CMaaS outperforms DMM due to pre-planned path verification in the Resource and Mapping DB. According to the results depicted in Figure 6, the packet delivery ratio shows a consistent increase in accordance to the change in velocity. However, with DMM there was a drop as the speed of a node reaches 40 m/s. This can be attributed to the complexity of the control plane in DMM which would take some time to cope with the changing environment. Packet loss is to be expected in such scenarios when using DMM. Likewise, in Figure 7, while simulating traffic conditions as the network density increases, the bandwidth utilization achieves a clear limit for DMM at over 80%, whereas the CMaaS logic continues to increase even beyond 320 clients. Our investigation warrants the fact that reduction in control plane complexity can lead to flexibility in managing resources at run-time.

Figure 8 and 9 show an exhaustive simulation with increasing node densities to locate the point where the graph saturation occurs. As cell radius increases, CMaaS shows reduction in hand-off failure probabilities. This is because as the radius increases, the hand-off threshold does not fluctuate below a certain point, hence, having a much less handover requirement. In DMM, the same operation looks a lot more clustered, which can be attributed to the control plane signaling.

We calculate a session/service to mobility ratio (definition borrowed from [20]) to compare session disruptions. As shown in Figure 10, CMaaS provides relatively low service disruptions. This can be attributed to the fact that the CMaaS controller has resource reservation which consistently obeys the mapping database. In all cases, it was observed that CMaaS provides approximately similar performance with respect to handover latency and minimizing service disruptions.

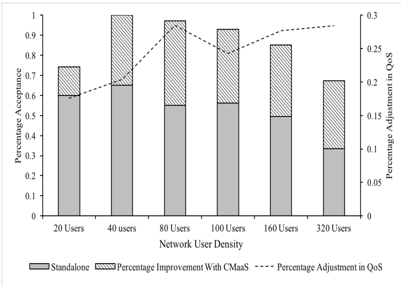

Fig. 11: CMaaS vs Standalone

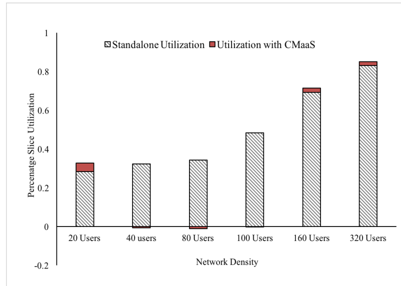

Fig. 12: Network Slice Utilization

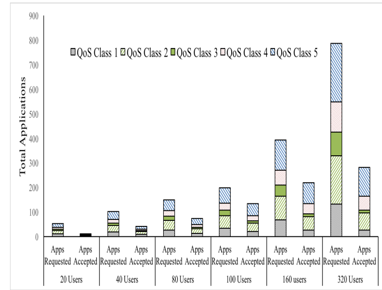

Fig. 13: Traffic Generated vs Accepted Applications

B. *Sensitivity Analysis of CMaaS*

Algorithms 1 and 2 provide details regarding how the MEC handles requests. As seen in line (6-8), a best QoS and least QoS definitions provide translation of applications running in device's user index (u), A_l^u. The applications are translated to network attributes like bandwidth and latency. As seen in line (12-36), the CMaaS allocation of resources is based on these values.

Algorithm 1 Translating User Requirements to Network Requirements

---

1: **procedure** QOSE() ▷ Maps User Applications to the Network Requirement
2: **Input:** A_l^u ▷ list of Applications running on user u
3: QoS_t ▷ Shown in Table I defines a hash map for different Web2.0 applications
4: L_m ▷ is the latency of Application which is most time sensitive
5: L_M ▷ is the latency of Application which is least time sensitive
6: $BestQoS = \dfrac{TotalBW}{L_m}$
7:
8: $LeastQoS = \dfrac{TotalBW}{L_M}$
9: **return:** $BestQoS$,$LeastQoS$
10: **end procedure**
11:

CMaaS shows a consistent improvement over the standalone deployment. In addition to the extra infrastructural costs, SA incurs an issue of isolation between 4G to 5G handover. A request from a 4G slice would not be accepted until there is an operator agreement, due to lack of a 4G anchor. As shown in Figure 11, the advantage that the CMaaS service layer provides is clear: a 15-30% increase in application acceptance with a network QoS readjustment. QoS readjustment is simply a recalculation of network requirements for user applications that happens at the target location, based on 4G network slice characteristics.

Since the work performed in [5] is not available for comparison using simulations, we emulate a standalone system by using an IP locator that is isolated from the network and compare it with CMaaS behaving in a full duplex fashion with the IP locator agent creating a tunnel between the two slices. The utilization of a slice increases with an increase in network user density. Moreover, as shown in Figure 12, the utilization in the CMaaS-based system increases or remains

---

Algorithm 2 Assignment of Network Slices to User Applications

---

1: **procedure** CMAAS ASSIGMENT()
2: **Input:** $L_u = \{L_i, L_{i+1}.., L_n\}$
3: ▷ L_u List of Users in the Network
4: **Input:** $A_u = \{a_u^i; \forall i \in$ All the application per user$\}$
5: **Input:** N = All the Available Network Slices for allocation
6: **Output:** Network slices to whom the users are mapped.
7: **for** u in L_n **do**
8: $BestQoS_{A_u}, LeastQoS_{A_u}$ = $QoSe(A_u)$
9: **for** $slice_i$ in N **do**
10: **if** $BestQoS_{A_u} \leq slice_i^{aR}$ & $slice_i$ ==“5G” **then**
11: $slice_i \leftarrow u$
12: ▷ $slice_i^{aR}$ Available Network Resources in $slice_i$
13: ▷ mapping user application A_u to network $slice_i$
14: **break**
15: **end if**
16: **if** $LeastQoS_{A_u} \leq slice_i^{aR}$ & $slice_i$ =“5G” **then**
17: $slice_i \leftarrow u$
18: ▷ mapping user application A_u to network $slice_i$
19: **else**
20: Go to next $slice_i$ in the list N
21: **end if**
22: **end for**
23: **if** A_u not Mapped to any $slice \in N$ **then**
24: ▷ Try to Find a Network Slice in 4G Spectrum
25: ▷ Repeat the CMaaSAssigment() For “4G” Network Slice
26: **end if**
27: **end for**
28: **end procedure**

---

equivalent to the standalone approach. CMaaS acts as a manager that can handle dynamically varying loads.

Traffic is generated according to user requests made to the CMaaS. As shown in Figure 13, the total accepted applications are user-centric, such as messaging, gaming and other applications with specified QoS classes. In practice, we define different user profiles according to the QoS. The QoS classes defined are shown in Table I. Consider a network of 20 users, if every user is running two applications (e.g. messaging and on-line gaming), then the total application traffic is ($'20|QoS-Class1$ and $20|QoS-Class4'$). Essentially, the total traffic amounts to 40 applications running. The traffic generated on a link corresponds to the values mentioned in the table with CMaaS accepting these requests for resource provisioning. The slices that accept the applications are purely based on the resources that are available at that point, while rejecting the requests which cannot be accepted due to finite resource availability. It is clear that the readjustment strategy works to avoid a disconnection but with reduced QoS. In such cases, a trade-off exists, such that, we choose to maintain the connection and minimize QoS for that transition time.

TABLE I: QoS Considered

QoS Class	Bandwidth Requirement	Latency Requirement	Example
1	Very High (>50 Mbps)	Very Low	Real Time Gaming (Black Desert)
2	High (>25 Mbps)	Very Low	Video Conference (Hangout, Zoom)
3	High (>10 Mbps)	Low	Real Time Video and Photo sharing (Facebook, Instagram)
4	low (>5 Mbps)	Low	Voip (Cisco jabber, Whatsapp)
5	low (<5 Mbps)	High	Web browsing

V. Conclusion and Future-Work

In this paper, we proposed an architecture for mobility management between heterogeneous network slices in a 5G network. A novel, mobile edge cloud-based management architecture that controls inter-slice handover was developed, The architecture incorporates a service module (CMaaS) that can determine and provide subscription-based connectivity services for end users. Owing to the performance benefits in comparison to the DMM protocol, the proposed solution is a possible candidate for 5G mobility management. We evaluated how signaling costs are reduced relative to the DMM protocol. We have shown how bandwidth utilization can be maximized to obtain over 12.5% improvements when compared to the DMM technique. Additionally, a 15-30% increase in request acceptance was achieved using the QoS readjustment logic. In the future, we intend to study radio resource virtualization and workload placement issues in vehicular edge Clouds service management based network slices [22] [23] and multi-interface device characteristics such as [24]. We also plan to investigate how the readjustment logic influences workload placements behaviors in such environments.

References

[1] XU Jiangwei and Moon et al., "Method and apparatus for handover in mobile content centric network," Jan. 9 2018, US Patent 9,867,094.

[2] "Service oriented 5G core network," [Online] http://carrier.huawei.com/ /media/CNBG/Downloads track-HeavyReading.pdf, G. Brown, Feb 2017.

[3] "Ericsson white paper: 5g system," Ericsson, Jan 2015.

[4] R. Wen et al., "Robust network slicing in software-defined 5g networks," in *IEEE Glob. Comm. Conf.*, Dec 2017, pp. 1–6.

[5] H. Zhang and N. Liu et al., "Network slicing based 5g and future mobile networks: Mobility, resource management, and challenges," *IEEE Comm. Magazine*, vol. 55, no. 8, pp. 138–145, 2017.

[6] O. Houidi et al., "An efficient algorithm for virtual network function scaling," in *2017 IEEE Glob Comm. Conf.*, Dec 2017, pp. 1–7.

[7] Ismaeel Al Ridhawi, Moayad Aloqaily, Yehia Kotb, Yousif Al Ridhawi, and Yaser Jararweh, "A collaborative mobile edge computing and user solution for service composition in 5g systems," *Transactions on Emerging Telecommunications Technologies*, p. e3446.

[8] Y. Ch. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile Edge Computing A key technology towards 5G," *ETSI White Paper No. 11 Mobile*, , no. 11, pp. 1–16, 2015.

[9] "Unique ericsson and bt study: Network slicing is key for the iot business case," [Online] https://www.ericsson.com/en/digital-services/trending/economic-study-5g-network-slicing , 2018.

[10] P. Seite et al., "Requirements for distributed mobility," RFC7333.

[11] J. Heinonen, P. Korja, T. Partti, H. Flinck, and P. Pyhnen, "Mobility management enhancements for 5g low latency services," in *IEEE International Conference on Communications Workshops (ICC)*, May 2016, pp. 68–73.

[12] Luca Cominardi, Fabio Giust, Carlos J. Bernardos, and Antonio De La Oliva, "Distributed mobility management solutions for next mobile network architectures," *Computer Networks*, vol. 121, pp. 124 – 136, 2017.

[13] Mahadev Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[14] Solozabal Rubén et al., "Exploitation of mobile edge computing in 5g distributed mission-critical push-to-talk service deployment," *IEEE Access*, vol. 6, pp. 37665–37675, 2018.

[15] Moayad Aloqaily, Ismaeel Al Ridhawi, Haythem Bany Salameh, and Yaser Jararweh, "Data and service management in densely crowded environments: Challenges, opportunities, and recent developments," *IEEE Communications Magazine*, March 2019.

[16] Tien-Thinh Nguyen, Christian Bonnet, and Jérôme Harri, "Sdn-based distributed mobility management for 5g networks," in *Wireless Communications and Networking Conference (WCNC)*. IEEE, 2016, pp. 1–7.

[17] H. Song, X. Fang, and L. Yan, "Handover scheme for 5g c/u plane split heterogeneous network in high-speed railway," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4633–4646, Nov 2014.

[18] T. Bilen et al., "Handover management in software-defined ultra-dense 5g networks," *IEEE Network*, vol. 31, no. 4, pp. 49–55, July 2017.

[19] R. Arshad, H. Elsawy, S. Sorour, T. Y. Al-Naffouri, and M. Alouini, "Handover management in 5g and beyond: A topology aware skipping approach," *IEEE Access*, vol. 4, pp. 9073–9081, 2016.

[20] Yinan Li and Ray Chen, "Mobility management in wireless mesh networks utilizing location routing and pointer forwarding," *IEEE Transactions on Network and Service Management*, vol. 9, no. 3, pp. 226–239, 2012.

[21] A. Varga, "Using the omnet++ discrete event simulation system in education," *IEEE Transactions on Education*, vol. 42, no. 4, pp. 11 pp.–, Nov 1999.

[22] M Aloqaily, V Balasubramanian, F Zaman, I Al Ridhawi, and Y Jararweh, "Congestion mitigation in densely crowded environments for augmenting qos in vehicular clouds," in *Proc. of the 8th ACM Symp. on Des. and Analysis of Intel. Vehi. Netw. and App.*, New York, NY, USA, 2018, DIVANet'18, pp. 49–56, ACM.

[23] Ala Abu Alkheir, Moayad Aloqaily, and Hussein T Mouftah, "Connected and autonomous electric vehicles (caevs)," *IT Professional*, vol. 20, no. 6, pp. 54–61, 2018.

[24] V. Balasubramanian, M. Aloqaily, F. Zaman, and Y. Jararweh, "Exploring computing at the edge: A multi-interface system architecture enabled mobile device cloud," in *2018 IEEE 7th International Conference on Cloud Networking (CloudNet)*, Oct 2018, pp. 1–4.