Invited Paper: Intelligent Agent Support for Achieving Low Latency in Cloud-Native NextG Mobile Core Networks

Shalini Choudhury*
shalini@winlab.rutgers.edu
WINLAB, Rutgers University
New Brunswick, New Jersey, USA

Sushovan Das*
sd68@rice.edu
Rice University
Houston, Texas, USA

Sanjoy Paul
sanjoy.paul@accenture.com
Accenture Labs
San Francisco, USA

Ivan Seskar
seskar@winlab.rutgers.edu
WINLAB, Rutgers University
New Brunswick, New Jersey, USA

Dipankar Raychaudhuri
ray@winlab.rutgers.edu
WINLAB, Rutgers University
New Brunswick, New Jersey, USA

ABSTRACT

Next-generation mobile core networks are being designed to support a variety of latency sensitive applications based on emerging virtual, augmented or mixed reality technologies. A cloud-native approach for 5G core has been proposed to meet the diverse service requirements of NextG while reducing both CAPEX and OPEX. In this context, microservice architecture for network function virtualization is generally considered to be suitable for meeting NextG service requirements. Despite many advantages, the cloud-native core raises new challenges in the design of NextG systems for latency critical applications. An approach to achieving diverse QoS requirements is proposed in this paper. Specifically, the design is based on an orchestrator called the *MEC-Intelligent Agent* (MEC-IA) which enables dynamic compute resource distribution and network slice assignment in the core for improved QoS. The MEC-IA framework realizes resource management by intelligently assigning UEs to the access and mobility management function (AMF) while also performing slice provisioning. Simulation results are presented for the proposed MEC-IA framework showing the median control plane delay reduced by a factor of 1.67×. Further, robustness of the system improves significantly, reflecting a better overall user experience since the percentage connection dropped at 3× traffic volume reduces by 1.5× and slices assignment increases by 1.4× across all slices, even when the traffic arrival is skewed.

CCS CONCEPTS

• **Networks** → **Network architectures**; **Cloud computing**; • **Computer systems organization** → **Real-time system architecture**; • **Computing methodologies** → **Distributed computing methodologies**.

KEYWORDS

cloud-native core, low latency, QoS, slicing, resource distribution

*Both authors contributed equally to this research.

ICDCN 2023, January 4–7, 2023, Kharagpur, India

ACM ISBN 978-1-4503-9796-4/23/01.
https://doi.org/10.1145/3571306.3571386

ACM Reference Format:
Shalini Choudhury, Sushovan Das, Sanjoy Paul, Ivan Seskar, and Dipankar Raychaudhuri. 2023. Invited Paper: Intelligent Agent Support for Achieving Low Latency in Cloud-Native NextG Mobile Core Networks. In *24th International Conference on Distributed Computing and Networking (ICDCN 2023), January 4–7, 2023, Kharagpur, India.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3571306.3571386

1 INTRODUCTION

The 5G and beyond standard of cellular communications requires a minimum of 1 Gbps for mobile and stationary users. The demand for data rates is increasing rapidly because of multimedia applications [14] and according to Cisco, mobile data traffic in 2022 will increase to 77.5 exabytes [9]. The increase in the demand for high data rates and cellular traffic introduces challenges like scalability and flexibility in the NextG (5G and beyond) network. Advancements in virtualization and softwarization technologies have enabled a paradigm shift in how mobile networks are designed and operated to address the requirements of high data rates and mobile data traffic. A cloud-native core is one such virtualized technology solution leading to a significant transformation of communication networks [10] and enables keeping pace with the NextG network requirements by allowing increased service availability, cost-efficient operation and management, and on-demand service deployment. Further, cloud-native network core is gaining momentum in private 5G network, e.g. in the smart industrial manufacturing sector, to improve operational costs and support diverse quality of service (QoS) requirements of wide range of applications [21].

Figure 1 shows the high-level architecture of a cloud-native mobile core with containerized network functions (NFs) hosted in a server. Two key features of the cloud-native mobile core architecture are the high degree of functional decomposition and the distributed deployment of virtualized NFs. The network virtualization principle realizes NFs as software instances that can be deployed on commodity servers and storage devices, also enabling virtual network slices [16]. As per 3GPP service-based architecture (SBA), cloud-native core supports control and user plane separation (CUPS) [7]. These microservice oriented NFs' designs are superior to their monolithic counterparts [8].

During the deployment phase of cloud-native core, a cloud service provider (CSP) typically offers a static set of available hardware configuration blocks (i.e., the context of Openstack – a unique

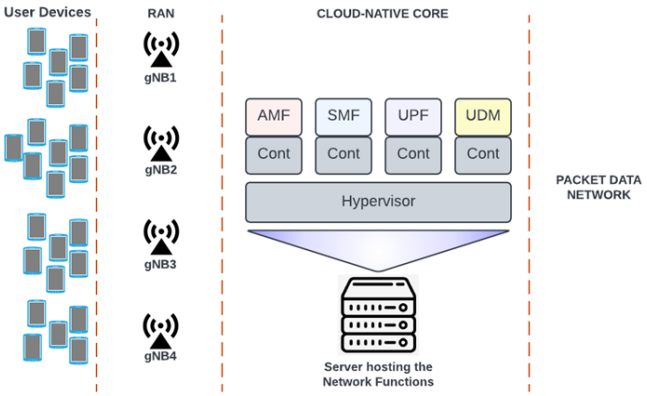

Figure 1: Overview of cloud-native mobile core.

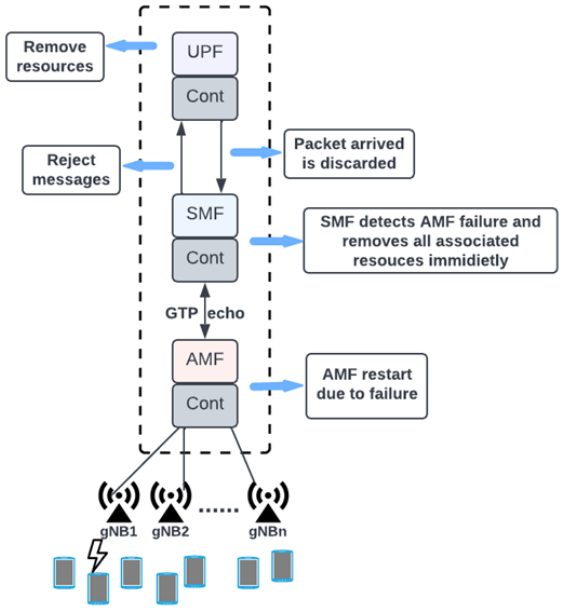

Figure 2: AMF overload scenario (within 3gpp specification)

combination of memory capacity and CPU) to a virtual NF, thus enabling operators to create and deploy network components in accordance with their needs without involving dedicated hardware components. Whilst, this could seem to be a good solution but the fluctuating nature of mobile traffic may result in poor performance for latency-critical NextG networks, resulting in resources being under-utilized or over-provisioned causing bottlenecks, especially at the access and mobility management function (AMF). The AMF participates in user equipment (UE) authentication, authorization, mobility, and session management which potentially makes it the system bottleneck. Hence, special focus should be given to the dynamic resource provisioning of the AMF to ensure network scalability. We propose to design an intelligent orchestrator to address this limitation. The intelligent orchestrator maintains awareness of the changing user demands and the real-time status of underlying resources. The objective of this paper is to devise an approach that has the intelligence to understand the network requirements based on the varying nature of the UE control traffic with the ability to automatically perform dynamic resource allocation at the core. The approach considers the cloud-native core's computation availability and quality of experience (QoE) of end-users in the decision to perform scalability in a cognitive manner making the system self-organized. The remainder of this paper is organized as follows. Sections 2 & 3 present design consideration and related work respectively. Section 4 describes the MEC-IA scheme showcasing its technical benefits. Section 5 lists the simulation setup parameters and discusses the obtained results. Concluding remarks are given in Section 6.

2 DESIGN CONSIDERATION

The state-of-the-art cloud-native and software defined network core architecture maintains compliance with the 3GPP communication reference interfaces. Notably, the access control and session management functions are separated in the core to better support fixed access, while ensuring scalability and flexibility. The most relevant core network functions are AMF, session management function (SMF), User plane function (UPF), Unified data management (UDM) and network exposure function (NEF).

The AMF ensures UE-based authentication, authorization, mobility management and communicates with the SMF to setup session management. UEs, even those using multiple access technologies, are connected to a single AMF since the latter is agnostic to the access technology. Based on the operator or service provider policy on packet flow, the policy and charging function (PCF) in the core defines policies about mobility and session management, which will be enforced by the AMF and SMF. Hence, AMF supports massive signaling traffic from connected devices, considering use cases such as the internet of things (IoT) and massive Machine Type Communication (mMTC), overutilizes the AMF network function and creates a bottleneck at the point of entry in the network [11].

According to the current state-of-the-art solution, AMF can be configured to restrict the number of received packets per second to address the bottleneck. The packets exceeding the number of permitted calls are rejected to prevent congestion at the AMF. To ensure there are no snowballing effects (e.g., due to re-transmission of rejected calls), a back-off timer is configured on AMF. When traffic congestion at AMF occurs, it sends an overload signal to gNB and gNB restricts signal distribution to the overloaded AMF [17]. Thus, this method is a reactive approach to preventing congestion at AMF.

Further, in the current cloud-native core, the overloaded AMF may fail and trigger restart event. Currently, this limitation is addressed by creating instances of AMF for restoring the service. However, the event of AMF failure and restart causes the cleanup of resources reserved for UEs being served by the AMF, which is very likely to lead to the deregistration or release of PDU sessions as

shown in Figure 2 [18]. Therefore, resulting in control plane delay along with service downtime at the UE's end.

Considering that separation of mobility management functions is advantageous from the scalability point of view, where the UE states and the session states will be hosted respectively in the UDM and the SMF, while the AMF will only be dedicated for processing tasks. Consequently, this separation augmented with a logically centralized orchestrator, provides a lightweight solution to mitigate the bottleneck created at the AMF.

In our design choice to remove bottleneck, we envision MEC-Intelligent Agent (or MEC-IA) as a logically centralized entity hosted in the mobile edge platform, periodically monitors AMF utilization, and thus realizes proactive and dynamic resource provisioning at the core in real-time. MEC-IA framework uses simple management interfaces (APIs) to collect RAN information and simultaneously communicates with NEF to import AMFs' compute resource statistics, which makes the system light-weight. The fundamental advantages of such MEC-IA assisted proactive and dynamic resource distributions are many-folds, such as:

- Proactive and dynamic resource distribution can avoid the heavy load imbalance across AMFs which in turn makes the system more robust in terms of preventing core-entry point congestion for considerably high network load and skewed traffic distribution.
- Resource distribution and management can be orchestrated intelligently, precisely and rapidly to provide a well-classified QoS for real-time communications while also performing the slice provisioning as per UE's service subscription.
- MEC-IA assisted proactive resource provisioning attempts to avoiding AMF failure by continuous network monitoring and handling background network management tasks e.g., analyzing the control traffic statistics for future resource provisioning, and foreseeing AMF failure.

3 RELATED WORK

The Cloud-native core concept is built around a philosophy of containerization, and the management of those containers through orchestration tools. Leveraging this feature, [2] proposes an algorithm to equilibrate the load on the AMF instances by scaling out or in AMF instances depending on the network load to save energy and avoid wasting resources. In [19], cloud resource allocation is considered alongside virtual machine (VM) placement and migration based on VMs' CPU, memory, storage, network bandwidth along with resource contention. Some mechanisms that automatically scale up or scale down VM instances using threshold values of the indicators to trigger the operation. [3] also proposes an auto-scaling mechanism for evolved packet core's monolithic mobility management entity (MME) in terms of scalability and efficient load balancing features in a cloud-native environment. In [15], the proposed approach considers the use of dynamic service level agreements (SLA) between the service provider and virtualized NF (such as AMF) provider. Instead of allocating a fixed amount of resources for the whole life-cycle, the resources are varied based on predicted user load [12]. Further, a distributed MME model is proposed in [4] and [6]. Contrary to the approach proposed in [20], the MME model is stateful and based on an external users' state storage system. Thus, the migration between MMEs is limited only to UEs in an idle state. However, in the active state, UEs are attached to an MME instance. Therefore, another MME instance may receive a network event for that UE. This request has to be forwarded to the correct MME, potentially increasing latency of EPC procedures. The above research projects aim to re-architect the cellular network and address the core bottleneck by initiating AMF migration, replication, instantiation, and scaling techniques. Such approaches add to the core overhead and ultimately increase the CAPEX and OPEX of the network.

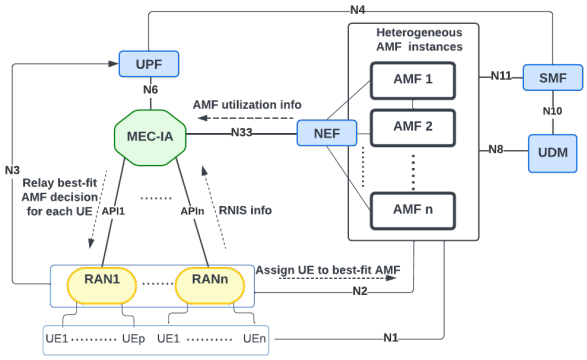

Figure 3: MEC-IA assisted NextG network architecture.

On the contrary, our MEC-IA proposed approach addresses the performance bottlenecks and latency associated with per-device connectivity at the control plane level, in a proactive and light-weight manner, i.e. no complex signaling and fast recovery of the load imbalance in the core. This work does not resort to vertical scaling (adding more processing resources like CPU, RAM, etc.), horizontal-scaling (add more instances of AMF), or heavyweight VM migration (incur paging latency) which would significantly add to the operational overhead. Instead, the proposed MEC-IA framework solves the system workload bottleneck problem by accurately assigning the UE and IoT bursty control signal traffic to a suitable AMF to provide efficient resource usage across operational AMFs in the core. The MEC-IA assisted mechanism ensures fast restoration of AMF failure without interruption to other services, and interoperability across heterogeneous AMFs, enabling AMF monitoring and control, while retaining compatibility with 3GPP specifications.

4 PROPOSED MEC-IA ASSISTED REDUCED CORE BOTTLENECK SCHEME

In the proposed scheme, we consider UEs' control traffic arriving and departing dynamically at the AMFs. The control traffic corresponds to multiple UE states such as *initial access request* (newly arrived UE), *idle-to-connected request* (the device was in sleep and then wakes up) or *handoff request* (due to mobility). Additionally, since NextG technology supports segregating traffic into three types

Algorithm 1: Resource-Aware Best Fit AMF Selection

Input:
N: Number of AMF
$R_{util}[N]$: resource utilization of AMFs
$R_{th}[N]$: resource threshold of AMFs
$slice_{util}[N][3]$: slice utilization of AMFs
$slice_{th}[N][3]$: slice threshold of AMFs
UE_state: state of the UE
$intended\_slice$: intended slice of the incoming UE
Output:
$AMF_{assigned}$: Best fit AMF for the UE
$slice_{assigned}$: Whether UE gets intended slice

1 **for** $i \leftarrow 1$ *to* N **do**
2 $base\_set\_up\_cost[i] \leftarrow exec\_time\_per\_bit[i] * data\_size(UE\_state)$
3 $placement\_cost[i] \leftarrow base\_set\_up\_cost[i] * (1 + floor(R_{util}[i] + 1)/R_{th}[i])$
4 **if** $(slice_{util}[i][intended\_slice] + 1) > slice_{th}[i][intended\_slice]$ **then**
5 $placement\_cost[i] \leftarrow placement\_cost[i] * (1 + \alpha)$
6 $slice\_service[i] \leftarrow 0$
7 **else**
8 $slice\_service[i] \leftarrow 1$
9 $AMF_{assigned} \leftarrow min_{index}(placement\_cost)$
10 $slice_{assigned} \leftarrow slice\_service[AMF_{assigned}]$
11 $R_{util}[AMF_{assigned}] \leftarrow R_{util}[AMF_{assigned}] + 1$
12 **if** $slice_{assigned} = 1$ **then**
13 $slice_{util}[AMF_{assigned}][intended\_slice] \leftarrow slice_{util}[AMF_{assigned}][intended\_slice] + 1$

of pre-defined slices: type 1 is dedicated to supporting of enhanced mobile broadband (eMBB), type 2 is for Ultra Reliable Low Latency Communications (URLLC) and type 3 is for massive machine-type communications (mMTC) support, the UEs may subscribe to different intended slices depending on the service requirements.

4.1 Proposed Framework

MEC-IA, the logically centralized entity hosted in the mobile edge platform as shown in Figure 3, proactively and dynamically provisions the incoming traffic to the appropriate AMF in order to mitigate the core bottleneck. The high-level objective of such dynamic resource provisioning would be to maximize the AMF resource utilization while maintaining fairness proportional to their compute/storage capability and also ensuring desired QoS for different UEs. Algorithm 1 illustrates our proposed greedy heuristic for the MEC-IA assisted resource-aware best fit AMF selection scheme.

MEC-IA receives resource and slice utilization update for all the AMF instances from the NEF, over the N33 interface (a 3GPP northbound API for securely exposing NFs' information and capabilities) and maintains a logically centralized database. Upon arrival of a new UE request at the gNB, MEC-IA receives the corresponding UE information (i.e., UE state, intended slice requirement etc.) through Radio Network Information Service (RNIS) over the RESTful RNI APIs.

On receiving the arrival notification of a UE request, MEC-IA computes the estimated placement cost of that UE to all the possible AMFs (Algorithm 1: lines 1-8). Following this, the MEC-IA assigns the UE to the *best fit* AMF having the minimum placement cost and simultaneously performs slice provisioning (Algorithm 1: lines 9-10), and finally updating the corresponding resource utilization parameter accordingly (Algorithm 1: lines 11-13).

4.2 Designing the Cost Function

In our heuristic, we design a penalty-driven cost function to maximize AMF utilization while ensuring proportional fairness. In this context, the *delay* is considered as the cost metric. It should be noted that the AMFs can be heterogeneous in terms of compute/storage resources and slice availability, captured by the parameters $exec\_time\_per\_bit$, R_{th} and $slice_{th}$ respectively. The resultant placement cost of a UE at a given AMF is the function of resource and slice utilization (R_{util} and $slice_{util}$), as shown in eqn. 3. More concretely, we model the placement cost as the combination of base setup cost and the penalties associated with AMF resource (R_{util}) and slice ($slice_{util}$) utilization at that time instant. The base setup cost of a UE to a given AMF (shown in eqn. 1) depends on the processing speed (shown in eqn. 2 [1]) of AMF and its corresponding UE control traffic volume. In our algorithm, we consider two specific penalties for the placement cost such as a) resource-load related penalty (R_{pen}) and b) slice related QoS penalty ($slice_{pen}$). We assume a linear resource-load related penalty function (R_{pen}) which depends on the ratio of current resource utilization (R_{util}) corresponding to the resource threshold (R_{th}) of a given AMF (both measured in terms of the number of currently serving UEs). As shown in eqn. 4, while estimating the resultant resource penalty

(R_{pen}) for a UE to a given AMF assignment, 1 is added to the current AMF resource utilization to explore the possibility in case the arrived UE would be assigned to that AMF. Additionally, we consider a constant slice related QoS penalty function ($slice_{pen}$) depending on whether the UE's slice subscription (for *URLLC, eMBB, mMTC)* request at a given AMF can be accepted. If the UE is assigned the intended slice there is no penalty i.e., $slice_{pen}$ = 0, else the $slice_{pen}$ = α, given in eqn. 5.

$$base\_set\_up\_cost = exec\_time\_per\_bit * data\_size \quad (1)$$

$$exec\_time\_per\_bit = num\_inst\_per\_bit * CPI/clock\_rate \quad (2)$$

$$placement\_cost = base\_set\_up\_cost*(1+R_{pen})*(1+slice_{pen}) \quad (3)$$

$$R_{pen} = \lfloor \frac{(R_{util}+1)}{R_{th}} \rfloor \quad (4)$$

$$slice_{pen} = \begin{cases} 0, & \text{if UE gets the intended slice} \\ \alpha \in [0,1], & \text{otherwise gets a QoS penalty} \end{cases} \quad (5)$$

In our simulation (Section 5), we have implemented the centralized version for both the database management and best-fit AMF assignment algorithm. Note that, our proposed algorithm can also be modified in a logically centralized and physically distributed setting where the MEC-IA entity would be hosted by more than one server in the mobile edge cluster and those edge servers would maintain a distributed database. Implementing the distributed counterparts of the database and algorithm are part of our future work.

5 RESULTS AND DISCUSSION

In this section, we introduce our simulation framework in detail and discuss the results.

5.1 Simulation Setup

We have developed a flow-level simulator in MATLAB where we simulate three control-plane scenarios. On one hand, scenarios 1 & 2 correspond to resource distribution under the assumption of an unbounded system i.e., no UE gets dropped. We capture the latency impact of exponential back-off (discussed in Section 2) by incurring a higher resource-load related penalty in the placement cost. On the other hand, scenario 3 corresponds to a bounded system (i.e., UEs can get dropped) to ensure QoS guarantee to the incoming UE through a slicing mechanism.

(1) **Resource unaware AMF Assignment**: This is the baseline scenario, where the UE is statically assigned to the default AMF selected by the gNB.
(2) **MEC-IA assisted Best-fit AMF Assignment**: This scenario corresponds to the resource aware best fit AMF selection and the control plane delay is evaluated by the *placement_cost*. Note that, currently MEC-IA invokes the assignment algorithm once for each UE upon its arrival. Once the UE is assigned to the best-fit AMF, the assignment is not changed during its lifetime. However, the MEC-IA can invoke the best-fit AMF selection algorithm at a given interval to dynamically reassign the UEs to the least utilized AMF, which would ensure better AMF utilization and better QoS while incurring negligible service downtime.
(3) **Intended Slice Assignment at the AMF**: This scenario represents a bounded system (explained later in this section) where the following two approaches fit in this scenario, i.e., a) either the incoming UE is assigned a slice, b) or the UE's connection is dropped.

Table 1: Simulation setup parameters

Parameter	Values
N	5
Traffic skewness	AMF 2,3 - 30%
	AMF 4 - 20%
	AMF 1,5 - 10%
Slice threshold	URLLC: eMBB: mMTC = 0.3:0.5:0.2

5.2 Simulation Parameters

The numerical values for the simulation are listed in Table 1. In our simulation model, we consider five AMF instances in the network core, where the AMFs are heterogeneous in terms of computation capability (i.e., $exec\_time\_per\_bit$ and R_{th}). If the number of UEs assigned to an AMF exceeds its R_{th}, that latest assigned UE gets the resource-related penalty (R_{pen}), as discussed in eqn. 3. The incoming traffic volume is spatially skewed over time among the AMF instances [5, 13]. AMF 2 & 3 each receive 30% of the total system traffic, followed by AMF 4 receiving 20% and AMF 1 & 5 receiving 10% each. The slice threshold at each AMF is set as $URLLC : eMBB : mMTC = 0.3 : 0.5 : 0.2$. This implies that the AMF can serve 30% of the total control plane traffic for the URLLC slice, 50% for the eMBB slice and 20% for the mMTC slice. Any further slice request by a UE beyond this slice threshold capacity will either incur a QoS-penalty ($slice_{pen}$ in eqn. 3) or be dropped by the AMF, respectively for unbounded and bounded scenarios.

5.3 System Performance of Baseline and MEC-IA Resource Distribution Methods

Figure 4(a) shows the resource utilization of the AMFs for the resource unaware AMF assignment scenario (baseline). The resource utilization disparity in the system is well evident from the graph. It should be noted that since both AMF 1 and AMF 5 receive a lower volume of control traffic due to the traffic skewness added in the simulation model, the resource utilization of AMFs 1 & 5 are four folds less than that of AMF 2 given that the later receives three times more control traffic. AMF 2 performs worse than AMF 3 even though both receive the same volume of control traffic due to the fact that the AMFs have heterogeneous compute availability and the $exec\_time\_per\_bit$ is less in AMF 3 in comparison to AMF 2.

In Figure 4(b), due to intelligent UE traffic assignment performed by the MEC-IA, the AMFs are neither under-utilized nor over-utilized. The over-utilized AMFs lag in servicing the control plane traffic request, thus becoming a source of the bottleneck at the point of entry in the core network. The system performs well at both the service provider's and the UE's end since the MEC-IA framework

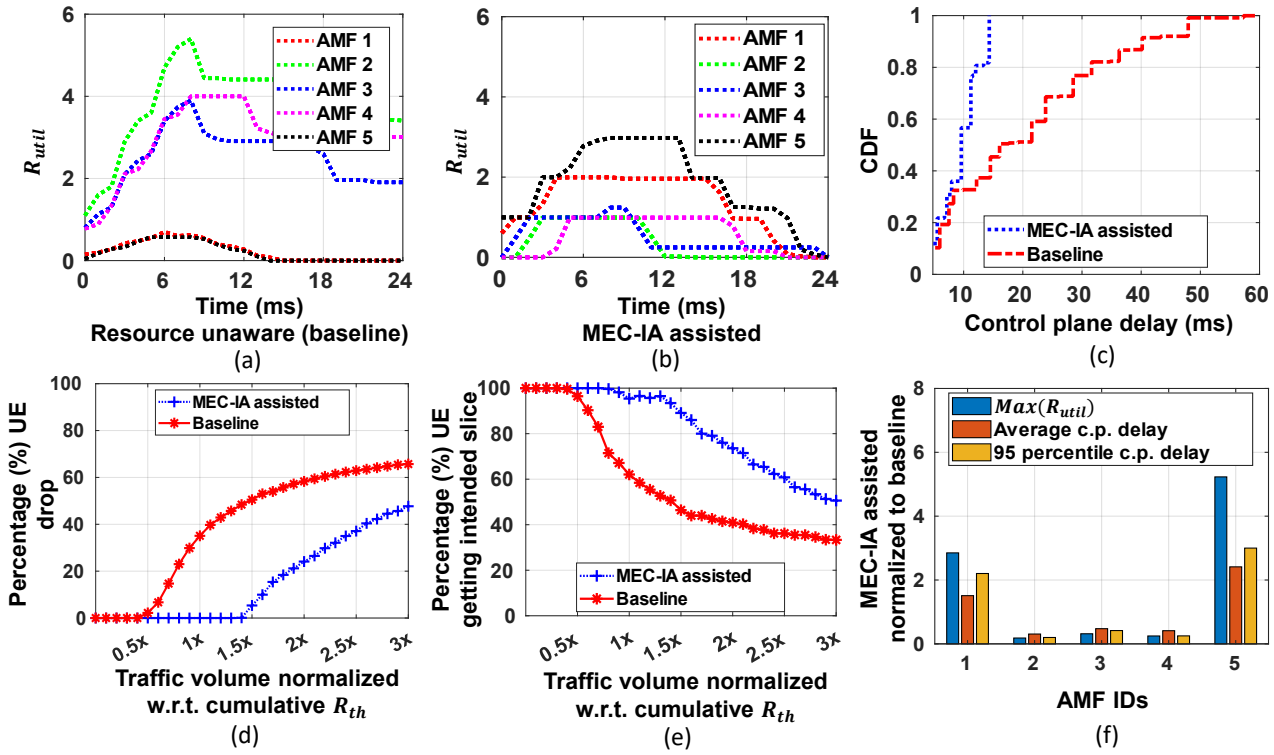

Figure 4: (a) Resource unaware UE to AMF assignment: utilization of AMF instances, (b) MEC-IA assisted resource aware UE to best-fit AMF assignment: Utilization of AMF instances, (c) CDF plot of core control plane delay, (d) Percentage of connection drop for MEC-IA assisted resource aware UE to best-fit AMF assignment and baseline, (e) Percentage of connection getting intended slice with MEC-IA assignment and baseline, (f) Max resource utilization scenario for MEC-IA: Relative change of control plane delay during peak resource utilization.

achieves fairness in terms of resource distribution in the core as well as reducing the control plane delay as seen in Figure 4(c).

The CDF plot of the control plane delay is shown in Figure 4(c) and it is evident that the control plane delay for MEC-IA assisted AMF selection scheme is significantly low in comparison to the baseline case of resource unaware AMF assignment. The MEC-IA evaluates the placement cost of a UE's traffic with respect to all the AMF instances in the core, followed by selecting the best-fit AMF. The placement cost includes a penalty in terms of the delay based on resource and slice availability. In a scenario where more than one AMF is selected as best-fit AMF (the AMFs were evaluated to have min [$placement\_cost$]) for a given UE's control traffic, in that case, the AMF with least resource utilization is preferred over the AMF with slice availability. We can observe from Figure 4(c) that with the MEC-IA approach, median control plane delay improves 1.67× the baseline control plane delay.

5.4 Intended Slice Assignment or UE Connection Drop

The simulation model discussed so far was unbounded, i.e. when the AMF's resources have been exhausted, the incoming UE's control traffic is serviced at the cost of adding penalty (in terms of delay (ms)) to the $placement\_cost$, but no UE connections were dropped. However, now we will analyze a bounded system where the AMF will deny UE connectivity if resources are exhausted. Figure 4(d) & 4(e) depicts the percentage of UE connection drops and UE connections that are assigned to the intended slices (i.e., the slice the UE has subscribed) respectively for both baseline and MEC-IA assisted best fit AMF selection.

In Figure 4(d), the control traffic is increased with respect to the cumulative resource threshold (R_{th}), 0.5 to 3. For the MEC-IA approach, there is no connection drop observed until 1.4× the traffic volume in the core. This means that the MEC-IA core network system shows robustness by serving all the incoming UE traffic, even though the incoming traffic load is 1.4× more in comparison to the available resources in the core. On the contrary, for the baseline case, a drop in UE connections begins when the available resources is 0.6× more than the incoming UE traffic. Hence, intelligent resource provisioning can potentially save 40% resource cost.

On analysing the Figure 4(e) it can be inferred that the percentage of end devices being assigned to the intended slice it had subscribed, at peak traffic volume (normalized w.r.t. cumulative R_{th} is 3×) is 50%, which is 1.5× more than the intended slice assignment for the baseline scenario. Additionally, it is observed that even when

the incoming traffic volume is less than 50% of the total resource, the resource unaware scheme is unable to assign the intended slice for some UEs. Therefore, slice drop (no slice assignment) events occur even before the traffic volume reaches 0.5×. While in the case of MEC-IA, premium service is availed by all the UEs that have subscribed for slice assignment, until 0.8× traffic volume, followed by a gradual drop in slice assignment. At 1.5× traffic volume, the MEC-IA framework provides better QoS guarantees to the UE, since the slice assignment is 1.8× more than that in the baseline.

Note that, we analyze both Figures 4(d) and 4(e) for even higher control traffic volumes. We observe that MEC-IA assisted resource provisioning approaches the baseline both in terms of percentage slice assignment and UE drop when the control traffic volume is around 12× w.r.t. cumulative resource threshold (R_{th}). However currently, our simulation model does not consider any downtime during the best-fit relay computation and also does not simulate the data plane performance. As a future step, we plan to extend our simulation to implement the end-to-end system considering all such realistic scenarios.

5.5 Control Plane Delay Performance During Peak Resource Utilization

The objective of Figure 4(f) is to show the relative change in control plane delay during the phase of maximum resource utilization at each AMF. The peak resource utilization of AMFs, for the MEC-IA scheme, is normalized w.r.t. baseline, along with the normalized average and 95 percentile control plane delay during that peak utilization instant. In MEC-IA assisted resource-aware scheme, AMFs peak utilization decreases due to resource aware provisioning, correspondingly the control plane delay improves significantly. For example, AMF 2 has 0.18× peak utilization w.r.t. baseline which leads to 3.27× and 6× average and 95 percentile control plane delay improvements respectively. Additionally, in our proposed scheme, the traffic volume across AMFs increases gradually. As a result, even though some AMFs serve more traffic due to higher available resources, the average and tail control plane delays do not overshoot abruptly. For example, although AMF 5 has 5.23× more peak utilization, the average and 95 percentile latency are degraded only by 2.4× and 3× respectively. Therefore, our scheme provides a better trade-off between resource utilization vs. QoS.

6 CONCLUSION

This paper proposes a *MEC-IA* framework that performs intelligent resource management in the cloud-native core network in order to address the issue of the bottleneck at the core, while simultaneously improving the traffic capacity and QoS. The MEC-IA is hosted at the mobile edge computation platform and is tasked with collecting AMF resource utilization statistics from the network exposure function over the N33 interface while maintaining a centralized database repository. The MEC-IA executes the *Resource-Aware Best Fit AMF Selection* algorithm. Additionally, this framework has been validated by a simulation study. It can be concluded from the results that the MEC-IA is able to address the mobile core network bottleneck problem by distributing the UE control traffic amongst available AMF instances and performing UE assignments to the best-fit AMF, thus also guaranteeing better QoS and enhancing end user experience. The control plane delay at the 95 percentile shows an improvement of 3.33× in comparison to the baseline, which clearly shows that the connection time is significantly reduced and the UE obtains more than 3× faster access to the network core. Similarly, it is observed that with the premium service assignment through slicing mechanism, the MEC-IA framework succeeds in assigning more intended slices (*URLLC, eMBB, mMTC*) to the UEs subscribing for the slicing services with fewer UEs being denied connectivity.

In the future, we will investigate the performance of the MEC-IA framework in a private 5G cloud-native smart factory environment. The MEC-IA will be incorporated in a 5G open-source implementation and a study will be conducted to evaluate how the core resources will be orchestrated in real-time to achieve low control plane delay. It will be interesting to study the MEC-IA performance in a smart industry setup since it supports a wide variety of applications with stringent QoS.

ACKNOWLEDGMENTS

Research supported in part by a research gift from Accenture Technology Labs and in part by National Science Foundation (NSF) grant number 2148104.

REFERENCES

[1] 2022. *Basic performance equation.* http://www0.cs.ucl.ac.uk/teaching/B261/Slides/lecture2/tsld015.htm.
[2] Imad Alawe, Yassine Hadjadj-Aoul, Adlen Ksentini, Philippe Bertin, and Davy Darche. 2018. On the scalability of 5g core network: the amf case. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 1–6.
[3] PC Amogh, Goutham Veeramachaneni, Anil Kumar Rangisetti, Bheemarjuna Reddy Tamma, and A Antony Franklin. 2017. A cloud native solution for dynamic auto scaling of MME in LTE. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 1–7.
[4] Xueli An, Fabio Pianese, Indra Widjaja, and Utku Günay Acer. 2012. DMME: A distributed LTE mobility management entity. *Bell Labs Technical Journal* 17, 2 (2012), 97–120.
[5] Ashutosh Balakrishnan, Swades De, and Li-Chun Wang. 2020. Traffic skewness-aware performance analysis of dual-powered green cellular networks. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 1–6.
[6] Arijit Banerjee, Rajesh Mahindra, Karthik Sundaresan, Sneha Kasera, Kobus Van der Merwe, and Sampath Rangarajan. 2015. Scaling the LTE control-plane for future mobile access. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*. 1–13.
[7] Gabrial Brown. 2017. Service-based architecture for 5G core networks. *Huawei White Paper* 1 (2017).
[8] Shihabur Rahman Chowdhury, Mohammad A Salahuddin, Noura Limam, and Raouf Boutaba. 2019. Re-architecting NFV ecosystem with microservices: State of the art and research challenges. *IEEE Network* 33, 3 (2019), 168–176.
[9] J Clement. 2019. Global mobile data traffic 2017-2022. *Statista, Available: https://www. statista. com/statistics/271405/global-mobile-datatraffic-forecast/(Accessed September 2022)* (2019).
[10] Qiang Duan, Shangguang Wang, and Nirwan Ansari. 2020. Convergence of networking and cloud/edge computing: Status, challenges, and opportunities. *IEEE Network* 34, 6 (2020), 148–155.
[11] Endri Goshi, Michael Jarschel, Rastin Pries, Mu He, and Wolfgang Kellerer. 2021. Investigating inter-nf dependencies in cloud-native 5g core networks. In *2021 17th International Conference on Network and Service Management (CNSM)*. IEEE, 370–374.
[12] Adlen Ksentini, Tarik Taleb, and Khaled B Letaif. 2015. QoE-based flow admission control in small cell networks. *IEEE Transactions on Wireless Communications* 15, 4 (2015), 2474–2483.
[13] Dongheon Lee, Sheng Zhou, Xiaofeng Zhong, Zhisheng Niu, Xuan Zhou, and Honggang Zhang. 2014. Spatial modeling of the traffic density in cellular networks. *IEEE Wireless Communications* 21, 1 (2014), 80–88.
[14] Jayakumar Loganathan, S Janakiraman, and TP Latchoumi. 2017. A novel architecture for next generation cellular network using opportunistic spectrum access

scheme. *Journal of Advanced Research in Dynamical and Control Systems,(12)* (2017), 1388–1400.

[15] Bipin B Nandi, Ansuman Banerjee, Sasthi C Ghosh, and Nilanjan Banerjee. 2013. Dynamic SLA based elastic cloud service management: A SaaS perspective. In *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*. IEEE, 60–67.

[16] Matteo Pozza, Patrick K Nicholson, Diego F Lugones, Ashwin Rao, Hannu Flinck, and Sasu Tarkoma. 2020. On reconfiguring 5G network slices. *IEEE Journal on Selected Areas in Communications* 38, 7 (2020), 1542–1554.

[17] Rajaneesh Shetty, Anil Jangam, and Ananya Simlai. 2021. Intelligent Strategies for Overload Detection & Handling for 5G Network. In *2021 IEEE 4th 5G World Forum (5GWF)*. IEEE, 135–140.

[18] Lucas BD Silveira, Henrique C de Resende, Cristiano B Both, Johann M Marquez-Barja, Bruno Silvestre, and Kleber V Cardoso. 2022. Tutorial on communication between access networks and the 5G core. *Computer Networks* (2022), 109301.

[19] Gaurav Somani, Prateek Khandelwal, and Kapil Phatnani. 2012. VUPIC: Virtual machine usage based placement in IaaS cloud. *arXiv preprint arXiv:1212.0085* (2012).

[20] Yusuke Takano, Ashiq Khan, Motoshi Tamura, Shigeru Iwashina, and Takashi Shimizu. 2014. Virtualization-based scaling methods for stateful cellular network nodes using elastic core architecture. In *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*. IEEE, 204–209.

[21] Jinjun Xiong and Huamin Chen. 2020. Challenges for building a cloud native scalable and trustable multi-tenant AIoT platform. In *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 1–8.