Handover Authentication Latency Reduction using Mobile Edge Computing and Mobility Patterns

Fatima Abdullah [*1], Dragi Kimovski [†2], Radu Prodan [‡2], and Kashif Munir [§1]

[1] *National University of Computer and Emerging Sciences, Islamabad, Pakistan*

[2] *University of Klagenfurt, Austria*

Abstract

With the advancement in technology and the exponential growth of mobile devices, network traffic has increased manifold in cellular networks. Due to this reason, latency reduction has become a challenging issue for mobile devices. In order to achieve seamless connectivity and minimal disruption during movement, latency reduction is crucial in the handover authentication process. Handover authentication is a process in which the legitimacy of a mobile node is checked when it crosses the boundary of an access network. This paper proposes an efficient technique that utilizes mobility patterns of the mobile node and mobile Edge computing framework to reduce handover authentication latency. The key idea of the proposed technique is to categorize mobile nodes on the basis of their mobility patterns. We perform simulations to measure the

[*] i161036@nu.edu.pk

[†] dragi.kimovski@aau.at

[‡] radu.prodan@itec.aau.at

[§] Corresponding Author: kashif.munir@nu.edu.pk

networking latency. Besides, we use queuing model to measure the processing time of an authentication query at an Edge servers. The results show that the proposed approach reduces the handover authentication latency up to **54%** in comparison with the existing approach.
Keywords. Mobile Edge Computing, Handover Authentication, Mobility Patterns.

1 Introduction

The massive growth of mobile devices has increased network-related demands like increased network bandwidth, reduction in response time, etc. Latency minimization has become a great challenge for latency-sensitive applications, which require a response in real-time. So, service interruption is not affordable for such applications, especially during a handover event, when mobile networks are used. One of the main reasons for service interruption is longer delays occurring in the handover authentication process.

In mobile cellular networks, a Mobile Node (MN) is free to move or roam about anywhere in the geographic area. However, during a handover, MN has to move from one Access Node (AN) to another [1, 2]. The ongoing data sessions of the MN are transferred from one AN to the next one where the MN has moved [3]. Basically, handover authentication process in cellular networks involves three entities i.e., MN, AN, and authentication server [4, 5]. Handover authentication is a process in which the identity of MN is checked in order to allow it to get attached to the new AN.

Recently, various schemes have been designed for handover authentication latency reduction [6, 7, 8] in mobile networks. Many ticket-based schemes have been devised in order to reduce authentication latency in the handover process [9, 10]. In [11], the number of signaling messages involved in the authentication process has been reduced to minimize the authentication cost. However, none of the above-mentioned approaches have exploited

the resources at the Edge and mobility patterns to reduce handover authentication latency.

Usually, during a handover authentication phase, all processing is done in the Cloud. Due to the centralized nature of this architecture, central nodes are far away from the *MNs*. It results in a high latency during the handover authentication phase. Therefore, there is a need to reduce authentication latency in the handover process for the persistent connectivity of *MN*. An efficient scheme is required to address this issue, which can address the handover authentication latency problem.

Mobile Edge Computing (*MEC*) is a computing paradigm that provides services to the end-users with minimal latency by performing the computations within close proximity of end-users. However, as stated above, the authentication of *MN* in Cloud results in longer authentication delays, so the *MEC* copes up with this authentication delay problem by performing the *MN* handover authentication on Edge servers.

Integrating *MEC* with cellular networks is a big step towards latency minimization [12, 13]. Edge computing has significantly contributed towards latency minimization in many emerging applications such as augmented reality [14], online gaming, connected vehicles [15], and in health-care applications [16, 17]. *MEC* leverages the mobile network components such as Base Station (*BS*) to provide Cloud services and mobile computing at the edge of the network [18]. *MEC* is paving the way towards *5G+* and its main aim is the latency minimization [19, 20]. Therefore, *MEC* empowers the Cloud by extending its services.

In this paper, we design an efficient scheme that utilizes *MEC* framework for reduction of the handover authentication latency. The key idea is to categorize the mobility of *MN* on the basis of its network activity patterns. For a highly mobile *MN*, the authentication is done on Cloud, whereas the authentication of a low mobile *MN* is done on an Edge server. For a chosen duration of time, we categorize *MN* into one of the four classes,

namely: high mobility, moderate mobility, low mobility, and no mobility. Categorization is based on the mobility pattern (using network residence time) of *MN*. Mobile Edge servers are used for the handover authentication of that *MNs* that have either low or no mobility. However, the handover authentication of that *MNs* that have either moderate or high mobility is done on the Cloud.

The key contributions of the proposed work are as follows:

- Design of a categorization scheme for latency reduction that occurs during handover authentication process of *MN* using its mobility patterns.
- Exploiting the resources at the mobile Edge for latency reduction.
- Modeling an *MEC* server for the calculation of delay of an authentication request at the *MEC* server.

The paper is organized as follows. In Section 2, the related work is discussed. Section 3 describes the proposed approach for handover authentication latency reduction. In Section 4, we present the results. Finally, we conclude in Section 5.

2 Related work

We categorize the related literature in two parts: (1) Handover authentication latency reduction schemes; (2) Edge-Based techniques to reduce the response time of jobs.

2.1 Handover Authentication Latency Reduction Approaches

Some existing approaches using ticket-based authentication schemes for latency reduction [9, 10]. HOTA [9] reduce authentication latency by securely reusing a ticket (user creden-

tials) during a handover. The authentication server distributes the ticket to the *MN* and the serving Mobile Access Gateway (MAG) during the initial authentication phase. This ticket is reused by the *MN* whenever it undergoes the handover authentication phase. The *MN* proves its validity through this ticket and does not involve an authentication server in this re-authentication process [9].

In [10], the authors introduce the hand-off authentication scheme based on ticket use for re-authentication in wireless mesh networks. This scheme comprises two phases, namely ticket issuing and re-authentication phases. In the first phase, tickets are issued by the authentication server for the mesh client. The mesh client stores tickets for the future re-authentication phase. Whenever the mesh client undergoes the re-authentication phase, it selects the ticket according to the identity of the target mesh router and sends its corresponding parameters. Then the target mesh router calculates the authentication key based on the ticket parameters. Then the mesh client and target mesh router authenticate each other by showing the shared key information. The scheme reduces the latency by eradicating the involvement of a third party, i.e., authentication server, during the re-authentication phase.

In [7], an efficient public-key based authentication (*PK-AUTH*) scheme for PMIPv6 domain is discussed. The scheme addresses the inter-domain handover authentication scenario, which lacks in the existing PMIPv6 authentication schemes. *PK-AUTH* scheme consists of four phases; initial registration, initial authentication, intra-domain handover authentication, and inter-domain handover authentication. During initial registration, *MN* generates its public and private keys. Then the *MN* sends its *ID* and Public key to the Certification Authority. As a response, *MN* certificate is generated and sent to the *MN* along with the Certification authority's public key. After the initial authentication phase, the *MN* and *MAG* authenticate each other. *MAG's* certificates are broadcast on a periodic basis. On receiving the *CertMag*, *MN* verifies the *MAG* certificate. After successful

verification, *MN* sends the authentication request to the *MAG*. *MN* is authenticated by verifying the *CERT-MN*. After its verification, authentication acknowledgment is sent to the *MN* along with the local symmetric key of *MN*. After this acknowledgment process, both entities, i.e., *MN* and *MAG* authenticate each other. Now, when the *MN* undergoes handover, re-authentication latency is reduced by reusing the local symmetric key of *MN*, the previous *MAG* sends this local symmetric key to the new *MAG*.

In [8], a local authentication scheme for intra-domain movement of *MNs* is discussed. This scheme eradicates the need for communication between the Authentication Authorization Accounting (AAA) server and the *MN* whenever the handover occurs within the same domain. In this scheme, the security association is locally maintained, and consequently, the intra-domain handover authentication latency is reduced. In [21], the authors present a Chord-based handover authentication scheme for ID/location separator architecture. The scheme addresses the issues of long handover delays. Fast handover authentication (between different networks) is achieved after the completion of two-fold authentication between the mobile station and Access Node (AN).

2.2 Edge-Based Techniques to Reduce Job Response Time

According to the authors in [22], the main aim of Mobile Edge Computing (MEC) is to reduce latency by offloading the compute-intensive tasks to the network Edge. *MEC* utilizes the radio access network to provide the Cloud services at the network Edge. *MEC* attributes include close proximity, low latency, location awareness, and context information. Furthermore, the communication of mobile or portable devices with the Edge network is aided by wireless links. In [23], security challenges and threats of the Edge computing paradigms (Fog, mobile Edge and, mobile Cloud computing) are discussed.

The authors in [24] discuss the scheduling and resource allocation issues in the Fog-

Cloud environment. As different IoT applications have different latency requirements, these applications may either be latency-sensitive or latency-tolerant applications. Therefore, the authors highlight the impact of different scheduling techniques (First-Come-First-Served, Concurrent, and Delay-priority) on the quality of service of different classes of applications (latency-sensitive and latency tolerant).

In [25], the authors present an Edge-based system for the efficient offloading of computationally intensive tasks on Edge nodes. It is the client-Edge system, which is constructed to provide reduced latency in video analytics. The system performs optimization for the selection of offloading tasks and their processing prioritization in terms of minimal response. Furthermore, the Edge nodes collaborate with one another to provide minimal latency by processing the offloaded task within the close proximity of the end device.

In [26], the authors propose an efficient approach to enhance the quality of experience for video streaming in the smart cities context. *MEC* architecture is employed to provide optimum *QOE* with reduced latency to users. Video streaming latency is reduced by streaming it through Edge instead of Cloud. As the Cloud is multiple hops away, it is not feasible to provide high-resolution video to smart city residents while connecting to the Cloud.

In [27], an efficient approach is designed for the deployment of *MEC* platform in 4G LTE networks. In this design, *MEC* middlebox is placed on *S1* interface, between the core network and eNodeB. *MEC* middlebox hosts application servers in order to enable *MEC* services for users. Residing in between the eNodeB and signaling gateway, it filters and forwards the data traffic. The proposed approach is efficient in terms of its easy installation and less deployment cost. Moreover, it does not need any modification in the existing network elements, so it can easily be installed in the current 4G networks.

In [28], the authors perform the joint optimization for computational tasks of-

floading on Fog in an energy-efficient manner. Energy consumption is considered as a minimization problem with respect to the delay constraint. In order to perform energy optimization, joint optimization of offloading probability and transmission power is done. A queuing model is employed in order to analyze the performance of the *MN* and Fog node in terms of response time. *M/M/1* queuing model is used to estimate the local execution time of the computational task on *MN*. *M/M/S* queuing model is used to estimate the time each request has to wait in order to get served on the Fog node.

The authors in [29] formulate a multi-objective optimization problem to achieve minimal delay, energy consumption, and payment cost. This is achieved by finding the appropriate value of transmit power and offload probability. In addition, a queuing model is employed by the authors for a thorough examination of offloading process in the Fog system.

The problem of optimal deployment of micro-service-based applications in MEC architecture is investigated within the context of deployment cost in [30]. The cost optimization is performed by taking into consideration the resource constraints of MEC servers and application response time. The authors in [31] provide a service placement technique within the Edge environment for micro-service-based applications. This scheme performs service placement by considering the heterogeneity of Edge servers and the uncertainty of end-users. The proposed work is different as it is aimed at reducing handover authentication latency using MEC.

The study in [32] addresses the issue of load evacuation within the Edge environment. In order to ensure good QoS in case of burst load, the strategy migrates the load to other Edge servers. It offloads the service requests to other Edge servers when the service request load exceeds the capacity of a certain Edge server.

In [33], the authors highlight the high-complexity and privacy-leakage issues of

network-slicing algorithms. In order to address these problems, the authors introduce an online algorithm, namely "DPoS (Decentralized, Privacy-Preserving, Low-Complexity Online Slicing)". DPoS encourages tenants to make their own decision based on their genuine preferences rather than providing any personal information to the operators or other tenants.

The concept of dynamic horizontal offloading for distributed stream processing in Edge computing paradigm is introduced [34]. Horizontal offloading refers to compute-intensive task-offloading to the peer Edge devices. The basic idea of the approach is the deployment of stream processing entirely on Edge, thus, completely liberating it from the centralized control.

The authors in [35] introduce an Edge-Cloud-based stream processing approach (SpanEdge) for the reduction of latency and bandwidth consumption. In the approach, the stream processing application is deployed in a distributed way that spans from Edge to Cloud data centers. The latency is reduced by reducing the transmission data size by utilizing the Edge nodes for partial computations.

The optimization framework for the optimal placement of data stream processing operators in the Edge-Cloud environment is described in [36]. The framework models the operator placement scenario as a constraint satisfaction problem by taking into consideration the computing and power consumption requirements of operators.

In the second category of related literature, different techniques for reducing the response time of jobs are discussed using *MEC*. The proposed approach (HALR-ECF) is different as it exploits the resources at the Edge to process the handover authentication requests of low-mobility *MNs* while processing the requests of high-mobility *MNs* using Cloud.

The proposed approach reduces both the networking and processing delay of the

whole system. The networking delay is reduced by processing the handover authentication requests on one hop distance (on Edge) from the *MN*. The processing delay is reduced by only processing the handover authentication requests of low-mobility *MNs* on the Edge. Processing all the authentication requests of low and high-mobility *MNs*) on the Edge increases the load of the authentication requests on the system, resulting in increased response times. Moreover, the proposed approach is designed in such a way that it takes into account the computing capacity of Edge servers. As Edge servers are equipped with limited computational resources, it is infeasible to process all handover authentication requests on the Edge. In case of moderate or high mobility, the *MN* frequently performs handovers as compared to a low-mobility *MN*, which would increase the load of handover authentication requests on the Edge system and that can result in higher response times. Hence, it is suitable to process the handover authentication requests of high-mobility *MNs* on the Cloud.

3 Proposed Approach, HALR-ECF

The proposed approach, Handover Authentication Latency Reduction using Edge-Cloud Framework (*HALR-ECF*), uses the resources at the Edge to reduce the handover authentication latency. We use the mobility pattern of an *MN* and employ *MEC* for the reduction of handover authentication latency of the *MN*.

We categorize the mobility of *MN* as no, low, moderate, or high based on its mobility pattern. The mobility pattern is associated with the sequence of residence times of *MN* in different Access Nodes (*ANs*). Cell Residence time is the attachment time of *MN* with an *AN* before it moves to the next *AN* as a result of handover. No, low, moderate, and high mobility are determined through handovers per unit time (handover rate). Low, moderate, and high-mobility handover rates are chosen as 0.3, 1, and 3 per hour, consistent

with the typical handover rates chosen in the related literature. For a moderate or high mobile *MN*, handover authentication is done on Cloud, whereas authentication of a non or low mobile *MN* is done on an Edge server. For a specific duration of time, we categorize *MN* into one of the four classes namely: high mobility (HM), moderate mobility (MM), low mobility (LM), and no mobility (NM). The categorization is based on the mobility pattern of the *MN*. Mobile Edge servers are used for the handover authentication of those *MNs* that have either low or no mobility.

The first step is mobility-related data generation of *MN*. The related literature [37] indicates that the residence times of *MN* is an Exponentially distributed stochastic variate. It is evident through literature that the cellular data is not available due to the privacy issues [38]. So, we generate the residence times of *MN* using Exponential distribution. During the data generation phase, we randomly choose the frequency of handovers for random periods of time. We generate the mobility data of one year for *MN*. After the data generation phase, *MN* categorization (based on its mobility pattern) is done in any of the four categories.

3.1 Categorization Technique

In this section, we discuss the proposed categorization algorithm. The algorithm performs the categorization of the mobility class of an *MN* by utilizing its mobility pattern.

Algorithm 1 takes as input the simulation time (Θ), handover rate threshold values, and the sample duration (S_d). The total simulation time is 1 year. The symbols represent the threshold values defined for each mobility class (α_{HM}, α_{MM}, α_{LM}, α_{NM}) (line 2). The proposed algorithm employs a weighted average to estimate the mobility class of the *MN*. First, the mobility class of the *MN* is checked after a specific time interval t, which is known as S_i duration (i-th sample duration). Then, the weighted average of the

handover rate, Ψ_i, of the i-th sample is calculated. The handover rate Ψ_i of the sample is then further used to classify the *MN* in one of the mobility classes and is calculated by using Equation (1).

$$\Psi_i = (1 - \beta)\Psi_{i-1} + \beta\Psi_i \tag{1}$$

In order to calculate Ψ_i, we assign a weight $\beta = 0.10$ (usually used for the latest sample) to the recent sample value Ψ_i and $(1-\beta)$ weight to the previous sample value, Ψ_{i-1} (line 8). The previous sample is given more weight because a decision to categorize *MN* in an MC (mobility class) should be done considering the history of the mobility pattern of the *MN*. Then, Ψ_i is checked against all mobility class ranges to assign a particular mobility class to the *MN* (lines 9-21). This process is repeated for each sample after the time interval 't'. We further categorize non or low mobile *MN* as low-mobility *MN* and moderate or high mobile *MN* as high-mobility *MN*.

After the mobility class assignment phase, the authentication request of a low-mobility *MN* is sent to the *MEC* server, and that of a high-mobility *MN* authentication request is sent to the Cloud (lines 22-26).

3.2 Reduction of Handover Authentication Latency

After categorizing *MN*, based on its mobility pattern, the handover authentication phase starts. Whenever the *MN* crosses the boundary of its *AN* and gets attached to a new *AN*, it undergoes the process of handover. Before the *MN* gets services from the new *AN*, it undergoes the process of authentication. During authentication, some messages are exchanged between the *MN*, *AN*, and authentication server. The number of message exchanges and the way of authentication depend on an authentication mechanism/protocol.

Algorithm 1 Mobility Categorization of *MN*.

Total Simulation time: Θ
Handover rate thresholds: α_{HM}, α_{MM}, α_{LM}, α_{NM}
Sample duration = S_d
$t = 0$, $i = 1$, $\Psi_0 = 0$, $\beta = value$ {Initialization}
BEGIN
while $t < \Theta$ **do**
  Calculate Ψ_i of S_i {handover rate of i-th sample}
  $\Psi_i = (1 - \beta)\Psi_{i-1} + \beta\Psi_i$ {weighted average of handover rate of i-th sample}
  if Ψ_i=0 **then**
    MC = MC_{NM} {no mobility}
  else
    if $\Psi_i \in (0,\alpha_{LM}]$ **then**
      MC = MC_{LM} {low mobility}
    end if
  else
    if $\Psi_i \in (\alpha_{LM},\alpha_{MM}]$ **then**
      MC = MC_{MM} {moderate mobility}
    end if
  else
    MC = MC_{HM} {high mobility}
  end if
  if MC=MC_{NM} OR MC=MC_{LM} **then**
    Handover authentication request of *MN* is sent to *MEC* server
  else
    Handover authentication request of *MN* is sent to Cloud
  end if
  t = t + S_d
end while
END

We employ the *MEC* framework/architecture in our approach in order to do handover-authentication of the *MN*. The proposed scheme is illustrated in Figure 1.

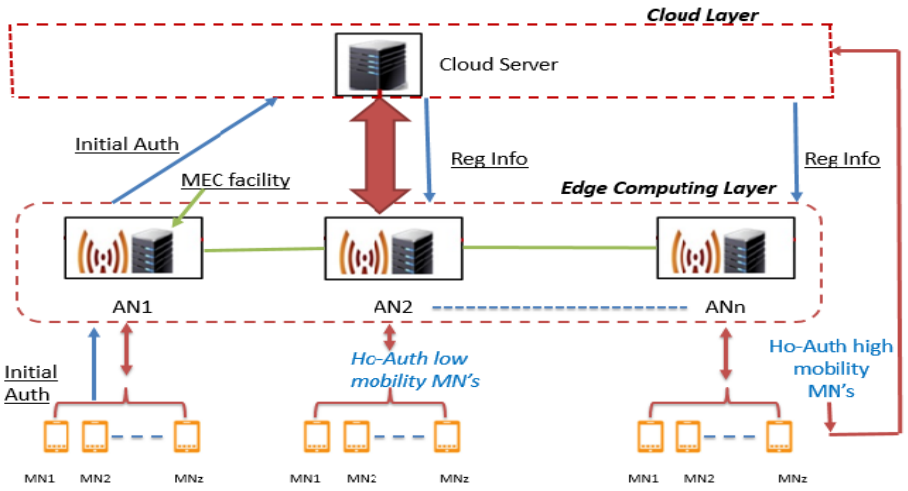


Figure 1: Architecture of the Proposed Approach.

The proposed system consists of *MN*, Edge server, and the Cloud server. There is a total of N number of *ANs*, where each *AN* consists of z number of *MNs*. Each *MN* is connected to an *AN* through a wireless channel. In the proposed scheme, each *AN* has *MEC* facility, which consists of *C* number of Edge servers. The Edge layer is further connected to the Cloud over multiple hops.

Authentication is a two-fold process, which consists of initial authentication and handover authentication. Initial Authentication occurs when *MN* boots up or access the network for the first time. Handover authentication occurs when the *MN* gets attached to the new *AN*.

There are two cases in our scenario: low-mobility *MN* handover authentication and high-mobility *MN* handover authentication. We use Cloud authentication service for initial authentication of the low-mobility as well as high-mobility *MNs*. When the

low-mobility *MN* is about to get connected to a new *AN*, its registration information is transmitted to the Edge server of the new *AN* from the Cloud Authentication Server. Here, it is stored in the local Edge server database and serves the purpose for low-mobility *MNs* authentication as long as it is connected to that *AN*. After that, all authentication of low-mobility *MNs* is locally done on the Edge server. After some time, when the *MN* again undergoes the process of handover and is about to get attached to the next *AN*, its registration information is transmitted from the Cloud Authentication Server to next *AN* Edge server and omitted from the previous Edge server database.

For high-mobility *MNs*, we utilize Cloud servers for handover authentication. Whenever the high-mobility *MN* undergoes handover, its authentication messages are exchanged between the *MN* and the Cloud Authentication Server. All authentication messages are exchanged between the *MN* and Cloud Authentication Server, which is multiple hops away from the access network. We use authentication latency as a performance metric for the proposed scheme. The efficiency of the proposed scheme is checked by calculating the overall latency (response time) for *MN* that occurred in the handover authentication process. The authentication Latency is measured as a sum of networking and processing delay (in the queuing system of an authentication server). We explain the calculation of the processing delay in the next section.

3.3 Calculation of Average Processing Delay

We apply a queuing model to calculate the average processing delay of an authentication request of a low-mobility *MN* on an Edge authentication server. The processing delay includes the waiting time of authentication requests in the queuing system. Because of the huge processing capability available on Cloud, it is realistic to consider the processing time of an authentication request of a high-mobility *MN* to be negligible as compared to the overall authentication latency, and hence it can be ignored. Each *AN* has *MEC*

facility that consists of a C number of Edge servers. Authentication Requests' (Auth-Reqs) arrivals on an *MEC* server follow a Poisson distribution with an average arrival rate λ_{LM}. The *MEC* facility employs Round Robin (*RR*) scheduling to equally divide the load on each i-th Edge Server (ES_i), where $i \in \{1, 2, ..., c\}$. After passing through the *MEC* load-balancing facility, the authentication requests arrive at ES_i with effective arrival rate λ^e_{LM}, where λ^e_{LM} is calculated using Equation (3). We model ES_i as *M/D/1* queuing system. In *M/D/1*, *M* represents the Exponentially distributed arrivals with arrival rate λ^e_{LM}, *D* represents the deterministic service time with service rate (μ), and *1* represents a single server [39]. In our case, each Auth-Req takes a deterministic service time. The proposed queuing system is illustrated in Figure 2.

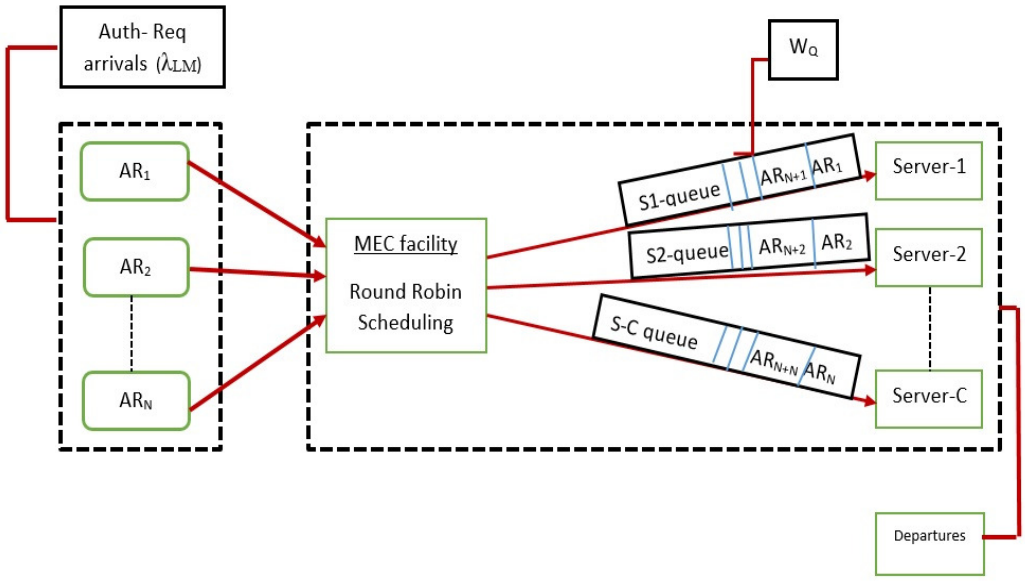


Figure 2: Queuing System of ES_i: AR stands for authorization request.

Figure 2 shows the process of arrival of Auth-Reqs (Authorization requests) on an *MEC* facility. The first block of flow model shows the Auth-Req's arrivals (AR_1, AR_2,...,AR_N) at the *MEC* facility. The next block represents the process of scheduling on the *MEC* facility in which AR_1 is destined to the ES_1, AR_2 is sent to the ES_2, and so on.

In this way, the load of authentication requests is equally divided among C *MEC* servers. We assume that every ES_i is identical with the same service rate μ, where $\mu = \mu_i$ and $i \in \{1, 2, ..., c\}$. Each ES_i has a separate queue and serves the authentication requests on First-Come-First-Served (*FCFS*) basis. The arrival rate λ_{LM} of authentication requests is calculated by using Equation (2) and the effective arrival rate λ^e_{LM} on an *MEC* server using Equation (3) as follows:

$$\lambda_{LM} = \lambda_{LM-HO} + \lambda_{LM-CO} \tag{2}$$

$$\lambda^e_{LM} = \frac{\lambda_{LM}}{C} \tag{3}$$

where λ_{LM-CO} is the Auth-Req arrival rate of those low-mobility *MNs* that are connected to the Edge server and send an authentication request after every time t and λ_{LM-HO} is the Auth-Req arrival rate of those *LM-MNs* that arrive from other *ANs* as a result of handovers. In Equation (3), λ^e_{LM} is the effective Auth-Req arrival rate on each ES_i. In order to find the processing time of *MN* in the *MEC* queuing system, we use Equation (4) as follows:

$$W_q^{ES} = \frac{\lambda^e_{LM}}{2\mu_{ES}(\mu_{ES} - \lambda^e_{LM})} \tag{4}$$

where $\mu_{ES} = \frac{1}{D}$.

$$\rho = \frac{\lambda^e_{LM}}{\mu_{ES}} \tag{5}$$

In Equation (4), W_q^{ES} represents the average processing time of the authentication query of *MN* in the queue, λ^e_{LM} is the effective arrival rate of authentication requests

arriving at ES_i. In Equation (5), ρ is the average utilization of ES_i. We evaluate the proposed mechanism in the next section.

4 Performance Evaluation

In this section, we describe the evaluation and present the results of the proposed Handover authentication latency reduction Edge-Cloud framework (*HALR-ECF*) approach. We perform the simulations using python3 with the anaconda framework, on Intel Core i3-4010 having a processing speed of 1.70 GHz and 4 GB main memory. The anaconda framework consists of pre-installed packages which are used for python programming [40].

The cellular network topology used for the simulations consists of 19 *ANs* as shown in Figure 3. We assume that on average, there are 1 million *MNs* per *AN*. Each *AN* has an *MEC* facility consisting of C Edge servers, where C ranges from 100 to 200. The *MNs* are connected with the *ANs* over the air interface in a radio access network. Furthermore, every *AN* is connected with the Cloud through wired Internet. We vary the number of *MEC* facility Edge servers from 100 to 200 in order to address the authentication requests in a reasonable time. Figure 3 shows the wrapped-around topology of a cellular network that is used in the simulations.

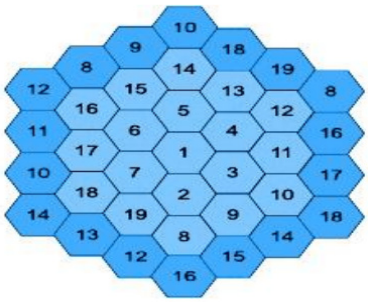


Figure 3: Topology of the Cellular Network.

The proposed approach for handover authentication latency reduction is evaluated by considering the EAP-Transport Layer Security (*EAP-TLS*) protocol [41] authentication messages over the Edge-Cloud framework. Extensible Authentication Protocol (*EAP*) is a standard authentication protocol and it has various authentication methods namely: *EAP-TLS*, *EAP-TTLS*, *PEAP*, *EAP-FAST* [42, 43, 44]. The most widely deployed *EAP* authentication method is *EAP-TLS*, as it is deployed by various wireless technologies i.e. IEEE 802.11, IEEE 802.16 [45]. We use *EAP-TLS* for the Edge-Cloud environment. Figure 4 illustrates the authentication message sequence when applied according to the *HALR-ECF* approach for Handover Authentication.

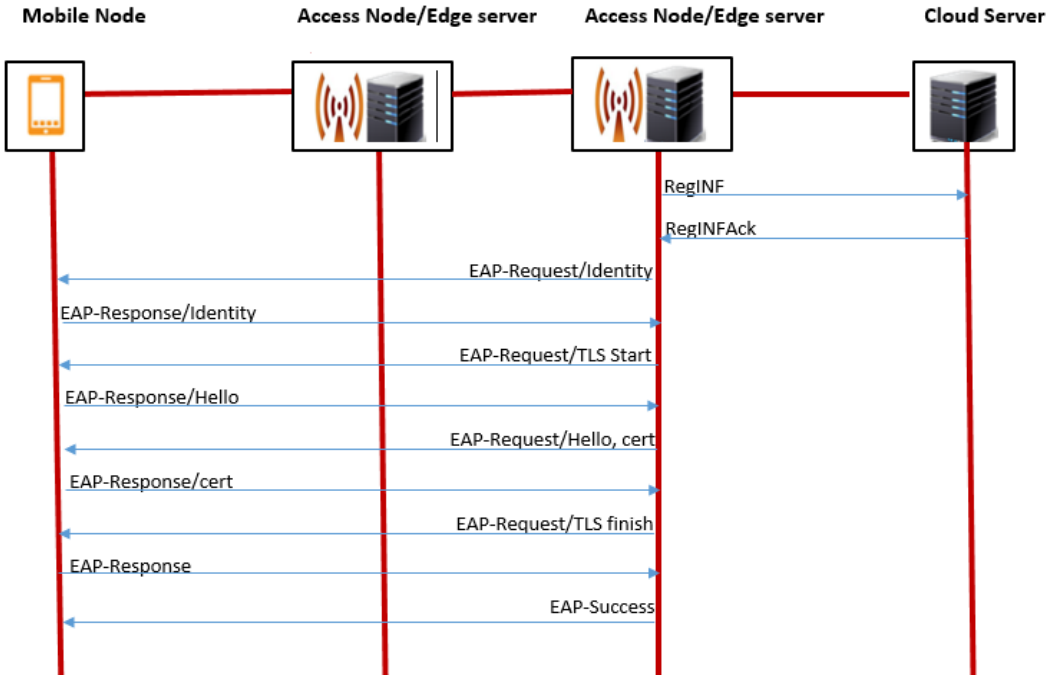


Figure 4: EAP-TLS Flow of Authentication Messages.

4.1 Simulation Parameters

We consider the parameter settings of [9] regarding the number of hops and average transmission delay per hop. Let $n \in [5, 9]$ be the number of hops between the Edge server and the Cloud. We assume *MN* to be one hop away from Edge servers. We consider one-hop transmission delay $ht_{delay} = 20$ ms [9]. For generating the network residence times of *MN* according to Exponential distribution, we use $\alpha_{LM} = 0.1$, $\alpha_{MM} = 0.3$, and $\alpha_{HM} = 0.5$. The simulation time is one year.

4.2 Handover Authentication Latency

We measure the handover authentication latency as a sum of the networking delay between *MN* and the authentication server and the processing delay of the authentication request at the authentication server. The processing delay includes the waiting time of an authentication request at an Edge server.

4.2.1 Networking Delay of an Authentication Request

In *EAP-TLS*, a full authentication message exchange is required between *MN* and the authentication server during handover authentication. Every time the *MN* gets connected to the new *AN* or boots up, the full authentication messages are exchanged between the *MN* and AS [46]. An *AN* is known as an authenticator in *EAP* terminology and acts as a relay between the *MN* and authentication server. It just passes the messages between the *MN* and *AN*, which results in longer authentication delays because the authentication server resides in the centralized Cloud. *EAP-TLS* is used as an authentication mechanism by various wireless technologies. In *EAP-TLS*, 4 round trips are executed between the *MN* and the Cloud Authentication Server. The handover authentication latency of *EAP-TLS*

in *HAC* (handover authentication using Cloud AS) approach is expressed in Equation (6) as follows:

$$L_{HO-Auth_{EAP-TLS}} = T_{MN-AN} + 8T_{MN-CAS} \tag{6}$$

Equation (6) shows the Handover authentication delay when the authentication server resides in the centralized Cloud. It is the networking delay involved in full *EAP* message exchange. For example, T_{MN-AN} shows the average networking delay between *MN* and *AN* and T_{MN-CAS} shows the average networking delay between *MN* and Cloud Authentication Server.

In the proposed *HALR-ECF* scheme, the initial authentication of low-mobility *MN* is done on the Cloud authentication server. During handover authentication, when the *MN* gets connected to the new *AN*, the *MN* registration information is transmitted from the Cloud Authentication Server to the new *AN*. This results in just one round trip between Cloud Authentication Server and the new *AN*. After that, all *EAP* authentication message exchange is locally done between the *MN* and the Edge Authentication Server (*EAS*). So the handover authentication latency of *EAP-TLS* by employing *HALR-ECF* approach is expressed by Equation (7) as follows:

$$HALR - ECF_{HO-Auth} = T_{MN-AN} + 8T_{MN-EAS} \tag{7}$$

Equation (7) shows the authentication latency when full *EAP* message exchange is locally done between the *MN* and the *EAS*. T_{MN-AN} represents the mean transmission delay between *MN* and *AN*; likewise T_{MN-EAS} represents the mean transmission delay between the *MN* and *EAS*.

4.2.2 Processing Delay of an Authentication Request

We calculate the processing delay of the authentication request using Equation (4) at the Edge server using the queuing model described above. It is the time that *MN* has to wait before its authentication request gets served. Figure 5 shows the impact of C (number of servers) on the processing delay. We vary the value of C in order to investigate its effect on queuing delay. A minimal queuing delay is obtained at $C = 200$. It is clear from Figure 5 that processing delay has an inverse relation with C. Increased value of C results in a decreased processing delay. So, overall the handover authentication latency using the proposed approach is calculated as follows:

$$HALR - ECF_{HO-Auth}(Total) = Networking_{delay} + Processing_{delay} \tag{8}$$

The processing delay of an authentication request on Cloud is negligible compared to the Edge, and hence, it is not considered. Edge has less computational resources than Cloud and authentication request has to wait for certain time t depending on the authentication request load on Edge server.

4.3 Results and Discussion

In this section, we discuss the results and present our analysis of the proposed approach. The Handover authentication latency of *MN* is observed by varying n, i.e., the number of hops between the access/Edge network and the core/Cloud network. The handover authentication latency is measured as a networking and processing delay (including the waiting time at a server) incurred during the authentication process between the *MN*, *AN*, and authentication server. We run the simulations by varying the hop count between the

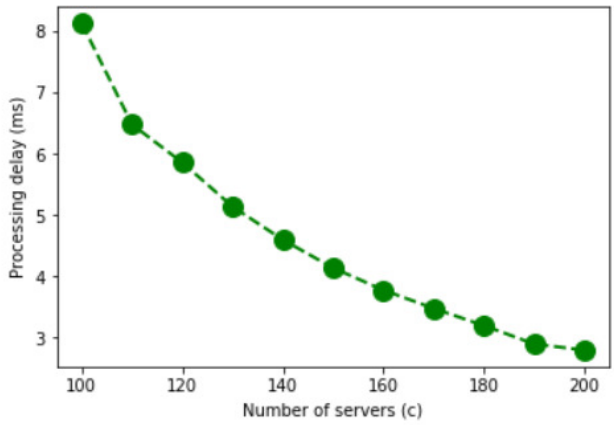


Figure 5: Impact of **C** (servers) on Processing Delay.

Access network and the Cloud to measure the networking latency involved in handover authentication of the *MN*.

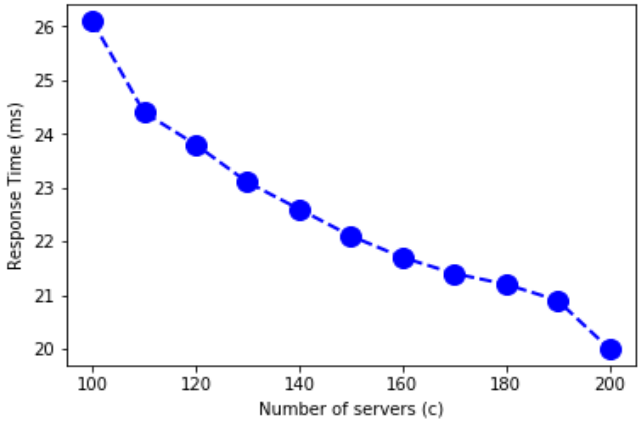


Figure 6: Impact of **C** (servers) on Response Time.

Now, we discuss and analyze the results of the proposed *HALR-ECF* approach as compared to the traditional Cloud-Based Approach (*CBA*) for handover authentication. The results have shown that the proposed approach reduces *EAP-TLS* latency up to a great extent. Figure 5 shows the impact of the number of servers on the processing delay

of an authentication request.

Figure 6 shows the impact of the number of servers (C) on response time. For this result, we vary the value of C from 100 to 200. Figure 6 shows that the response time of an authentication request has an inverse relation with the value of C. As the value of C increases, the load of authentication requests on each Edge server decreases. The load on each Edge server decreases because, in the proposed HALR-ECF approach, the MEC facility employs round-robin scheduling to distribute the load on each Edge server equally. The decrease in load results in reduced waiting time of an authentication request on an Edge server. Figure 6 shows that there is a rapid decrease in the response time when the value of C exceeds 190. This is because the queuing delay of the system (Edge server) sharply decreases, resulting in a sharp decline in response time.

Figure 7 shows the impact of the arrival rate of authentication requests on response time. We observe its effect on response time by varying the arrival rate (λ). The figure shows that the response time has a direct relationship with λ. It decreases with the decrease in authentication request arrival rate and vice versa. The response time is a sum of queuing delay and service time. Queuing delay is greatly affected by the arrival rate. If the arrival rate (authentication requests per unit time on *MEC* facility) increases or decreases, the waiting time per authentication request also increases or decreases.

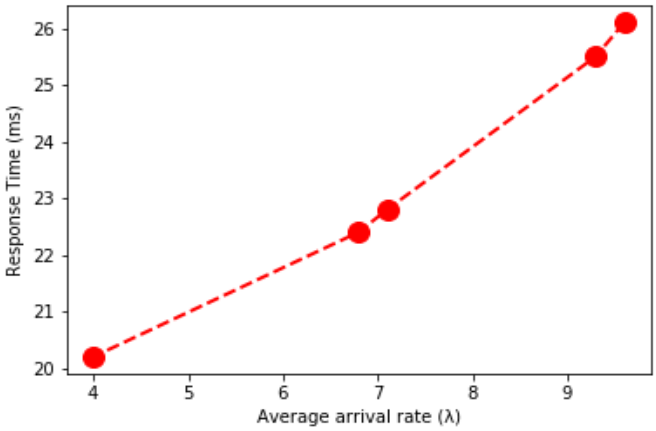


Figure 7: Impact of λ (arrival rate) on Response Time.

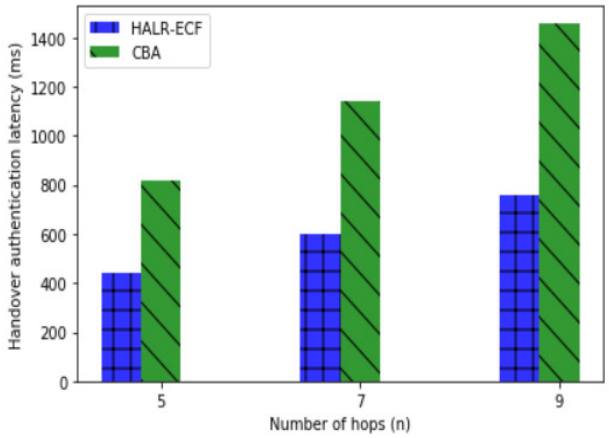


Figure 8: Handover Authentication Latency of *MN* by varying n from 5-9.

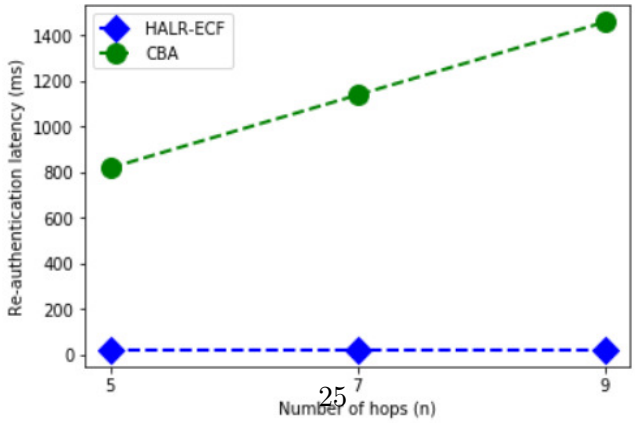


Figure 9: Re-authentication Latency of *MN* by varying n from 5-9.

Figure 8 shows the handover authentication latency over different values of n. As it is clear from the results that the authentication latency of *EAP-TLS* is significantly reduced by executing the message flow according to the proposed *HALR-ECF* approach. Whereas in *CBA*, *EAP-TLS* authentication latency is highly influenced by the increased value of n because it involves *MN* and the Cloud Authentication Server for handover authentication. Figure 9 shows the re-authentication latency. It can be seen that there is a significant difference in the results of our approach and the conventional Cloud-based approach. The main reason behind this improved re-authentication latency is that there is no need to do authentication signaling with the Cloud authentication server. Instead, an Edge server is utilized to locally perform the handover authentication in the access network, which is just one hop away from *MN*. As it can be seen in the figure 9, re-authentication latency increases with the increased number of hops between *MN* and Cloud Authentication Server. In contrast to this, *HALR-ECF* scheme is not affected by this increased hop count because *MN* is just one hop away from EAS.

5 Conclusions and Future Work

In this paper, we proposed a handover authentication latency reduction approach using Edge computing framework and mobility patterns of *MNs*. We categorized the *MNs* into low or high-mobility classes based on their mobility pattern. Then, we used a weighted average method to estimate the mobility class of the *MN*. For low-mobility *MN*, we utilized the Edge authentication service, and for high-mobility *MN*, we used the Cloud authentication service.

After categorizing the *MN* into above-mentioned mobility classes, handover authentication of *MN* is done either on Edge server or Cloud server according to its mobility class. The efficiency of the proposed approach is validated by calculating the overall latency

of the *MN* that occurs during the handover authentication phase. The improvement in results has been demonstrated by conducting experimental simulation, which measures the handover authentication latency. The proposed approach has outperformed the traditional Cloud-based approaches in terms of handover authentication latency by up to 54%.

The proposed work can reduce the latency of handover authentication requests in cellular networks to provide better services and connectivity during handovers. In future, we will extend the work to other applications and further explore the benefits of Edge/Fog computing in cellular networks.

Acknowledgements

This work received support from the DataCloud project funded by the European Union under the Horizon 2020 Programme (grant number 101016835), Kärntner Fog 5G Playground project funded by Carinthian Agency for Investment Promotion and Public Shareholding, and Ernst Mach-Nachbetreuungsstipendium (reference number ICM-2017-08089) by the Austrian Agency for International Cooperation in Education & Research (OeAD-GmbH).

References

[1] N. D. Tripathi, J. H. Reed, and H. F. VanLandinoham, "Handoff in cellular systems," *IEEE personal communications*, vol. 5, no. 6, pp. 26–37, 1998.

[2] Q.-A. Zeng, D. P. Agrawal, *et al.*, "Handoff in wireless mobile networks," *Handbook of wireless networks and mobile computing*, pp. 1–25, 2002.

[3] K. E. K. M. Vivek, "Efficient handover authentication scheme for mobile nodes in wireless networks," *International Journal of Engineering Research and Technology*, vol. 2, 2013.

[4] D. He, S. Chan, and M. Guizani, "Handover authentication for mobile networks: security and efficiency aspects," *IEEE Network*, vol. 29, no. 3, pp. 96–103, 2015.

[5] Y. Xie, L. Wu, N. Kumar, and J. Shen, "Analysis and improvement of a privacy-aware handover authentication scheme for wireless network," *Wireless Personal Communications*, vol. 93, no. 2, pp. 523–541, 2017.

[6] G. Li, J. Ma, Q. Jiang, and X. Chen, "A novel re-authentication scheme based on tickets in wireless local area networks," *Journal of Parallel and Distributed Computing*, vol. 71, no. 7, pp. 906–914, 2011.

[7] J. Kim and J. Song, "A public key based pmipv6 authentication scheme," in *2014 IEEE/ACIS 13th International Conference on Computer and Information Science (ICIS)*, pp. 5–10, IEEE, 2014.

[8] J.-H. Lee, Y.-J. Han, and T.-M. Chung, "Local authentication scheme based on aaa in mobile ipv6 networks," in *International Conference on Multimedia Modeling*, pp. 552–559, Springer, 2007.

[9] J.-H. Lee and J.-M. Bonnin, "Hota: Handover optimized ticket-based authentication in network-based mobility management," *Information Sciences*, vol. 230, pp. 64–77, 2013.

[10] L. Xu, Y. He, X. Chen, and X. Huang, "Ticket-based handoff authentication for wireless mesh networks," *Computer Networks*, vol. 73, pp. 185–194, 2014.

[11] A. K. Tripathi, J. Lather, and R. Radhakrishnan, "Secure and optimized authentication scheme in proxy mobile ipv6 (soas-pmipv6) to reduce handover latency," *Inter-*

national Journal of Computer Network and Information Security, vol. 10, no. 10, p. 1, 2017.

[12] X. Sun and N. Ansari, "Edgeiot: Mobile edge computing for the internet of things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, 2016.

[13] B. P. Rimal, D. P. Van, and M. Maier, "Mobile edge computing empowered fiber-wireless access networks in the 5g era," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 192–200, 2017.

[14] M. ETSI, "Mobile edge computing-introductory technical white paper," *etsi2014mobile, no. Issue*, 2014.

[15] U. Puetzschler, "Lte and car2x: Connected cars on the way to 5g," *Mobile Broadband SIG*, vol. 6, 2016.

[16] 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, no. 11, pp. 1–16, 2015.

[17] N. Sprecher, "Mobile edge computing an enabler for enhanced car2x communication," *ETSI white paper*, 2016.

[18] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[19] E. Ahmed and M. H. Rehmani, "Mobile edge computing: opportunities, solutions, and challenges," 2017.

[20] B. H. Allah and I. Abdellah, "Mec towards 5g: A survey of concepts, use cases, location tradeoffs," *Transactions on Machine Learning and Artificial Intelligence*, vol. 5, no. 4, 2017.

[21] M. Wan, Y. Liu, H. Zhou, and H. Zhang, "A chord-based handoff authentication scheme under id/locator separation architecture," 2010.

[22] E. A. Arif Ahmed, "A survey on mobile edge computing," 2016.

[23] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.

[24] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 26–35, 2017.

[25] S. Yi, Z. Hao, Q. Zhang, Q. Zhang, W. Shi, and Q. Li, "Lavea: Latency-aware video analytics on edge computing platform," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, p. 15, ACM, 2017.

[26] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile edge computing potential in making cities smarter," *IEEE Communications Magazine*, vol. 55, no. 3, 2017.

[27] C.-Y. Li, H.-Y. Liu, P.-H. Huang, H.-T. Chien, G.-H. Tu, P.-Y. Hong, and Y.-D. Lin, "Mobile edge computing platform deployment in 4g {LTE} networks: A middlebox approach," in {*USENIX*} *Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[28] Z. Chang, Z. Zhou, T. Ristaniemi, and Z. Niu, "Energy efficient optimization for computation offloading in fog computing system," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1–6, IEEE, 2017.

[29] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 283–294, 2017.

[30] S. Deng, Z. Xiang, J. Taheri, K. A. Mohammad, J. Yin, A. Zomaya, and S. Dustdar, "Optimal application deployment in resource constrained distributed edges," *IEEE Transactions on Mobile Computing*, 2020.

[31] H. Zhao, S. Deng, Z. Liu, J. Yin, and S. Dustdar, "Distributed redundancy scheduling for microservice-based applications at the edge," *IEEE Transactions on Services Computing*, 2020.

[32] S. Deng, C. Zhang, C. Li, J. Yin, S. Dustdar, and A. Y. Zomaya, "Burst load evacuation based on dispatching and scheduling in distributed edge networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 8, pp. 1918–1932, 2021.

[33] H. Zhao, S. Deng, Z. Liu, Z. Xiang, J. Yin, S. Dustdar, and A. Zomaya, "Dpos: Decentralized, privacy-preserving, and low-complexity online slicing for multi-tenant networks," *IEEE Transactions on Mobile Computing*, 2021.

[34] R. Dautov and S. Distefano, "Stream processing on clustered edge devices," *IEEE Transactions on Cloud Computing*, 2020.

[35] H. P. Sajjad, K. Danniswara, A. Al-Shishtawy, and V. Vlassov, "Spanedge: Towards unifying stream processing over central and near-the-edge data centers," in *2016 IEEE/ACM Symposium on Edge Computing (SEC)*, pp. 168–178, IEEE, 2016.

[36] G. Amarasinghe, M. D. de Assunçao, A. Harwood, and S. Karunasekera, "A data stream processing optimisation framework for edge computing applications," in *2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC)*, pp. 91–98, IEEE, 2018.

[37] Y.-B. Lin, M.-F. Chang, and C.-C. Huang-Fu, "Derivation of cell residence times from the counters of mobile telecommunications switches," *IEEE Transactions on Wireless Communications*, vol. 10, no. 12, pp. 4048–4051, 2011.

[38] K. Munir, E. Zahoor, W. Shahzad, and S. J. Hussain, "Intelligent reduction in signaling load of location management in mobile data networks," *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 8, no. 11, pp. 23–31, 2016.

[39] N. U. Prabhu, *Foundations of queueing theory*, vol. 7. Springer Science & Business Media, 2012.

[40] W. contributors, *Anaconda (Python distribution)*, 2020 (accessed July 31, 2020).

[41] D. Simon, B. Aboba, R. Hurst, *et al.*, "The eap-tls authentication protocol," *RFC5216, IETF, March*, p. 1, 2008.

[42] N. Cam-Winget, D. McGrew, H. Zhou, and J. Salowey, "The flexible authentication via secure tunneling extensible authentication protocol method (eap-fast)," 2007.

[43] P. Funk and S. Blake-Wilson, "Extensible authentication protocol tunneled transport layer security authenticated protocol version 0 (eap-ttlsv0)," 2008.

[44] A. H, J. S, Securit, Z. G, and A. B, "Protected extensible authentication protocol (peap)," 2001.

[45] J.-H. Lee, J.-H. Lee, and T.-M. Chung, "Ticket-based authentication mechanism for proxy mobile ipv6 environment," in *2008 Third International Conference on Systems and Networks Communications*, pp. 304–309, IEEE, 2008.

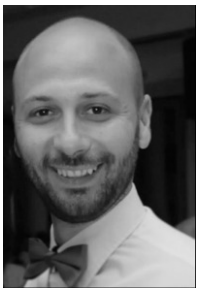
[46] T. Clancy, M. Nakhjiri, V. Narayanan, and L. Dondeti, "Handover key management and re-authentication problem statement," tech. rep., 2008.

Authors' Biographies

Fatima Abdullah

Fatima Abdullah did her BS (Telecommunication Systems) from Bahauddin Zakariya University and MS (Computer Science) from National University of Computer and Emerging Sciences, Islamabad. Currently, she is pursuing her Ph.D. from Kyungpook National University, South Korea. Her research interests include Edge/Fog/Cloud computing and mobile networks.

Dragi Kimovski



Dragi Kimovski is a tenure-track staff member at the Institute of Information Technology, University of Klagenfurt, Austria. He earned his doctoral degree in 2013 from the Technical University of Sofia. He was Assistant Professor at the University for Information Science and Technology in Ohrid and Senior Researcher at the University of

Innsbruck. He authored more than 40 articles in international conferences and journals. His research interests include parallel and distributed computing, Fog computing, multi-objective optimization, and distributed processing for bioengineering applications. He was a WP leader in the H2020 ENTICE project and currently acts as a Scientific coordinator in the H2020 ASPIDE project.

Radu Prodan

Radu Prodan is Professor in distributed systems at the Institute of Information Technology, University of Klagenfurt, Austria. He received his Ph.D. in 2004 from the Vienna University of Technology and was Associate Professor until 2018 at the University of Innsbruck, Austria. His research interests include performance, optimization, and resource management tools for parallel and distributed applications. He participated in numerous national and European projects as is currently principal coordinator of the H2020-ICT project ARTICONF (smART socIal media eCOsystem in a blockchaiN Federated environment). He authored over 100 publications and received two IEEE best paper awards. He is a member of IEEE.

Kashif Munir

Kashif Munir received his Ph.D. degree from the University of Innsbruck, Austria, in 2009. He was a post-doctoral researcher at IMT Atlantique (formerly known as Telecom Bretagne), France, from February 2011 to December 2012. He is working as a Head of Department and Associate Professor in the department of Computer Science at National University of Computer and Emerging Sciences (NUCES), Islamabad, Pakistan. His areas of research include admission and congestion control, quality of service for bulk data transfers, performance modeling of computer and communication systems, high performance computing, and mobility cost analysis. He is a reviewer of IEEE Transactions on Communications, Computer Networks, Telecommunication Systems, Cluster Computing, Concurrency and Computation, Journal of Network and Computer Applications, and International Journal of Computer Mathematics. He is an author of numerous peer-reviewed conference and journal publications.