Transparent 3rd-Party Authentication with Application Mobility for 5G Mobile Edge Computing

Asad Ali*, Ying-Dar Lin*, Chi-Yu Li*, and Yuan-Cheng Lai†
*Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
†Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan

***Abstract*— Multi-access Edge Computing (MEC) is a key technology for supporting low latency applications close to the end user. Users can access application servers in MEC instead of routing to the internet by passing through a core cellular network. Few security challenges arise as the traffic does not traverse through the core network, and these can be solved by providing authentication services in the MEC. However, authentication and application mobility issues arise in the case of multiple MECs where a user is mobile and needs continuous service from application servers, without needing to establish a new session and providing authentication information repeatedly to every new MEC it connects with. We propose two solutions, namely TC3A (Token-based Cookie transfer & 3rd-party Authentication) and TS3A (Token-based State transfer & 3rd-party Authentication) for resolution of authentication and application mobility issues while achieving low latency. Experiments were conducted on a testbed which emulated the handover of a user between two MECs. The results show that TC3A and TS3A provide authentication to users at reduced latency by 4.6% and 25%, respectively, as compared to simple login method, and most importantly application service continuity.**

Keywords*— Mobile Edge Computing, Multi-Access Edge Computing, Authentication, Mobility, Latency, 3GPP Cellular Networks

I. INTRODUCTION

Mobile Edge Computing (MEC), now known as Multi-access Edge Computing, is a new computing paradigm which brings the computing powers closer to the user. It can be considered as a cloud computing facility at the edge of the network. The major advantage of MEC is the reduction in latency compared to other computing paradigms. This reduction in latency is very attractive for those applications that need real time processing. Some of the applications that can take advantage of low latency are Vehicle to everything (V2X) communications and virtual reality (VR). MEC is deployed by cellular companies and is compliant with existing cellular networks whose infrastructure supports the deployment and operation of MEC. ETSI [1] and 3GPP [2] are currently building standards for the MEC, so that low latency MEC services can be provided in existing 4G LTE, and in upcoming 5G cellular networks. They have defined uses cases, deployment methods [3], architecture, issues, and challenges for the MEC. MEC is currently deployed in 4G-LTE networks through one of four basic scenarios [3] which differ from each other in terms of the point of deployment of the MEC host, and are termed as Bump in the Wire, Distributed EPC, Distributed S/PGW, and Distributed SGW with Local Breakout (SGW-LBO). MEC avoids network traffic congestion and propagation delays by preventing the traffic from entering the core of a cellular network. This allows MEC traffic to bypass the security functions that exist in the core network, and which can create security issues. MEC is deployed by a cellular company and is as secure as the core network but, applications in MEC can be deployed by third parties which creates further security issues such as extraction of user's radio analytics [4]. Such third-party applications can leverage the existing credentials information of a client in a core network to provide authentication. A few issues arise where multiple MECs, connected to different base stations, as shown in Figure 1, are deployed by the cellular network. In such cases, the issue of mobility arises where, if user equipment (UE) has established a session with an application server on MEC and moves to another MEC for the same application, it will have to establish a new session with the application server at that MEC which will increase latency and degrade user experience.

Apart from the mobility issue, as a UE moves from the application server on the source MEC (the MEC which already has a session established with UE) to the application server at target MEC (the MEC which needs to continue the service experience for UE), as seen in Figure 1, it will have to repeat the authentication process, in order to gain access to the services. This will introduce further latency in the application access. We therefore, identify two major issues in this scenario. First, the need for the UE to authenticate itself with the application server at the target MEC goes against the purpose of MEC which is to provide low latency services. Second, the establishment of a new user session with the application server at the target MEC also introduces latency. These issues lead us towards the research questions.

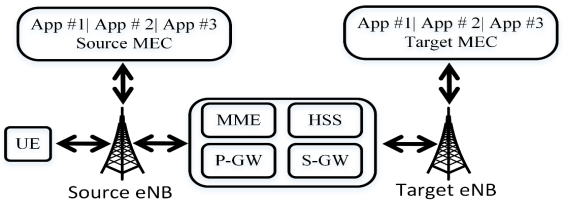

Fig. 1. MECs deployed in 4G LTE Network

978-1-7281-4355-2/20/$31.00 ©2020 IEEE

The questions that arise are: Is the seamless transition from the source to target MEC possible through using exchange of state information through UE? And, is it sufficiently secure to allow UE to carry authentication and state related information? In the literature, there are different types of web tokens available that are responsible for providing a single sign on experience for a user once they have been authenticated with the server, and do not need to provide their credentials every time. There are several types of tokens available, such as JSON Web Tokens (JWT) [5], Simple Web Tokens (SWT) [6], and Security Assertion Markup Language Tokens (SAML) [7]. These tokens can be used and modified for a scenario where a UE has to move from the source MEC to the target MEC. The session state can be stored in the server or the client, depending upon the model, and can also be transferred from the source to the target MEC. There have to be some security measures to ensure confidentiality and the integrity of the information transferred through the user.

A. Token based state and cookie transfer

In this work, we propose third party authentication of a UE application as the UE moves from the application server at the source MEC to the one at the target MEC. We apply tokens as they have already been used in current web applications to provide the single sign on experience. We propose application of an existing token to this MEC authentication problem, with a modification that includes the session state in the token. In that case, as the UE moves from source MEC to the target MEC, it carries all the necessary information with it, in order to experience the seamless transfer between the two MECs. We name the solutions TC3A (Token-based Cookie transfer and 3rd-party Authentication) and TS3A (Token-based State transfer and 3rd-party Authentication).

The use of a token between the MECs and UE solves the problem of latency, but introduces few security concerns, as it places too much information in the hands of UE; the application at the UE can be malicious and can try to change the contents of a token, steal confidential information, and send malicious packets to the core of the network. These are just a few of the security threats that can arise if we use TC3A and TS3A. We will provide necessary security measurements so that TC3A and TS3A are robustly resistant to these posed threats. We implement the TC3A and TS3A on test bed and deploy MEC servers to test the results of the TC3A and TS3A. The modules and architecture for this design will be explained in detail in the sections below. The implemented solution is evaluated for MEC handover delay. The time taken for the UE to resume the application service is also evaluated. The results show that TC3A and TS3A reduce latency by approximately 4.6% and 25% respectively, compared to a regular authentication mechanism.

The remainder of this paper is organized as follows. Section II covers the background on existing MEC integrated 4G-LTE architecture, along with the threat model and related work. We give our problem statement in section III with an example as explanation. Section IV documents the proposed design and architecture for TC3A and TS3A. The implementation of a prototype, modules and testbed is presented in Section V, and Section VI gives the results of evaluation. Section VII concludes the paper.

II. Related Work

There have been many studies in the literature which mostly look into the mutual authentication between client and server. We have looked into the studies which have provided authentication for those cases where fog nodes are connected with the cellular edge. We have also documented a few studies which support mobility in the MEC. These are closely related to our work, where we propose authentication and application mobility for the MEC. Some of these studies are documented in table I.

TABLE I. RELATED WORK

Author	Method	Authentic-ation	Transp-arency	Application Mobility
Hyeran[8]	Modified EAP-AKA	✓	X	X
Minghui[9]	Service Agent	✓	X	X
Yixin[10]	Secret Splitting	✓	X	X
Minghui[11]	Mobile-IP Handoff	✓	X	X
Sarang[12]	SDN	✓	X	X
Grasa [13]	RINA	X	X	✓
Zhang [14]	Enhanced MSS	X	X	✓
Tejas [15]	lightMEC	X	X	✓
Our Approach	TC3A, TS3A	✓	✓	✓

It can be seen from table I that, although these works focus on the authentication among the fog nodes and the cellular edge systems, they do not provide application mobility. They use different approaches like EPS-AKA, SDN, Service Agent, Mobile-IP handoff and SDN for the mutual authentication between client and server but they do not provide application mobility support and transparency in solution deployment. The studies [13-15] support the mobility in the MEC but do not provide the authentication. Our proposed TC3A and TS3A provide transparency, authentication and application mobility. A transparent solution is really important for deployment, as it does not propose any modifications to the existing underlying LTE architecture. The application mobility ensures user continuity of the application session between multiple MECs, and does not have to start the session again.

III. Problem Description

We assume we have two MECs, a source and a target MEC connected via an existing cellular network and UE is initially authenticated with and has a session established with the source MEC. We consider that the UE is mobile and it will get out of the range of the source MEC at some point, and will have to connect itself with another MEC which is target MEC. We have to transfer the authentication and application information from the application server in the source MEC to the application server in the target MEC, so as to provide a seamless handover to the UE application client, while achieving the low latency. We have assumed that the UE will use the same application in the both the MECs. The UE must access the same application

seamlessly from target MEC, and authentication information and session state information must be transferred from the source MEC to target MEC without the need of a login, while achieving low latency.

There are a few issues that that need to be solved and they are: How to authenticate the UE with target MEC? How to transfer state information from MEC-1 App server to MEC-2 App server? We will explain these issues with the help of an example. The issues of authentication and application mobility can be understood with the help of the scenario shown in Figure 1 where two MECs are connected to an LTE core network via their respective eNBs. One of the MECs is connected with the source eNB, which we call the source MEC, because we assume that the user is authenticated with this MEC and has established a connection with this MEC in order to access application 1. This application can be any generic web service, like video streaming or data processing.

We assume that client of the user equipment has established a session with the application server in the source MEC. Now, the scenario is that the user is mobile and moves out of the range of the source MEC, and approaches the target MEC. We call this the target MEC because the user now wishes to access the same application in the server installed in the target MEC. The first issue that arises here is that of authentication data transfer. If the UE needs to be authenticated again with the application server in the target MEC, it will introduce latency while accessing the application, which is not good for latency sensitive applications. The second issue is the transfer of session state. Whenever a client establishes a session with a server, the server stores some information related to that session and sends the session cookie to the client, so that client does not have to send its information every time. The server keeps record of the client through the session state and session cookie. Now, as the client has moved from the source MEC to target MEC, it will establish a new session with the target MEC and all the information stored in the source MEC application server will not be present in the target MEC application server. We need to transfer this information from source MEC to the target MEC, so that the user can resume the session with the target MEC application from the exact same position where it left the session with the source MEC application server. With this example, we have identified two issues which are authentication information transfer and application mobility from source to target MEC.

IV. Proposed Design And Architecture

In this section, we propose the solution to the two problems identified in the previous section. We used the token passing approach for transferring the authentication and state information from the source MEC to target MEC. Token passing approach is used because it is a safe way to transfer credentials from the server to client. We have modified this token passing approach to solve the problem of transferring authentication and application mobility between two MECs.

These approaches are named TC3A (Token-based Cookie transfer & 3rd-party Authentication) and TS3A (Token-based State transfer & 3rd-party Authentication). Both these approaches make use of the token, and transfer the

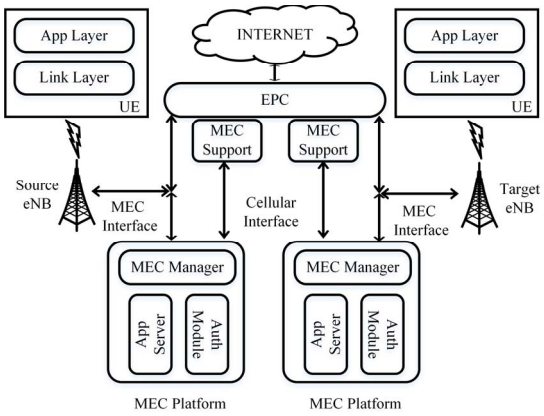

Fig. 2. Proposed MEC integrated cellular system architecture

authentication information and session state information from the source MEC to the target MEC. The token generated in both the approaches consist of the user information, token expiry time, and the source which generated the token. This token-based solution solves the problem of authentication and application mobility in the MEC transparently, as it does not involve any changes in the underlying LTE architecture for MEC deployment. The differences between these two solution approaches, TC3A and TS3A, are shown in Table II. 3-p authentication is provided by both of them while TC3A transfers the cookie from source MEC to target MEC and TS3A transfers the session state from source to target MEC. TC3A allows the UE to start the session at target MEC from the exact point where it left the session in source MEC while, TS3A might lose some state during transfer. TC3A is suitable for applications that cannot afford state loss but can compromise a little on latency while TS3A is suitable for applications that can afford a little loss in session state but have tighter latency constraints.

TABLE II. TC3A AND TS3A COMPARISON

Parameters	TC3A	TS3A
3-p Authentication	✓	✓
Cookie Transfer	✓	X
Session State Transfer	X	✓
Number of Tokens	1	N
Inter-MEC Connectivity	X	✓
Server Modification	Less	More

A. Architecture

The proposed architecture makes use of bump in the wire deployment method of MEC in cellular networks and is shown in Figure 2. There are multiple components in this architecture, which either belong to the cellular network or the MEC. We are concerned with the components which are included in the MEC platform because we have to design them. The architecture of source and target MEC platforms is similar to each other. The most important component is the MEC manager in both the MECs. The MEC manager is multi-purpose and communicates with the core cellular network and also the user equipment through the eNB. The MEC manager is responsible for checking the traffic that goes in and out of the MEC platform. It will also obtain authentication information about the user from the core network and send it to the authentication module, which will store the information. The MEC manager is also responsible for communicating with the application server in order to obtain the

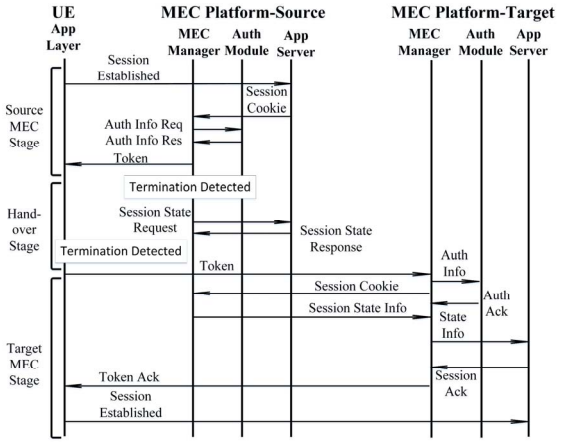

Fig. 3. TC3A message flow

cookie and state information for both the proposed solutions which are TC3A and TS3A. The MEC manager will also be responsible for generating the token that will be presented to the user equipment and also the security of the token. The messages exchanged between these modules is explained in next sub-section and is shown in Figure 3 and Figure 4.

B. Protocol

We have proposed two solution for the transfer of authentication information and session state information between a source and a target MEC, and the details of the message flow are given below.

TC3A (Token-based Cookie transfer & 3rd-party Authentication)

This approach is represented in Figure 3, and it can be divided into three stages: a source MEC stage, a handover stage and a target MEC stage. In source MEC stage, the UE establishes a session with the application server in the source MEC and the application server forwards the session cookie to the UE, which is then captured by the MEC manager as all the information between the UE application client and MEC application server goes through the MEC manager. As soon as the session cookie has been captured by the MEC manager, it sends a request to the authentication module to obtain the user's authentication information. When the MEC manager has the session cookie and the authentication information, it creates a token on the basis of this information and presents this token to the user equipment. As soon as the user moves away from the vicinity of the source MEC, both the application client and application server will detect session termination and will react accordingly. The MEC manager of source MEC will request the session state information from the application server by presenting it with a cookie, and the application server will send the information of the session to the MEC manager, where it will be saved. This is the handover stage.

For the target MEC stage, the user equipment connects to the target MEC and presents the token to it, to be received by the MEC manager. The MEC manager will then extract the authentication information (user credentials, source MEC ID, and source MEC signature) and session cookie from the token, and send the authentication information to the authentication module of the target MEC, which will then authenticate the user

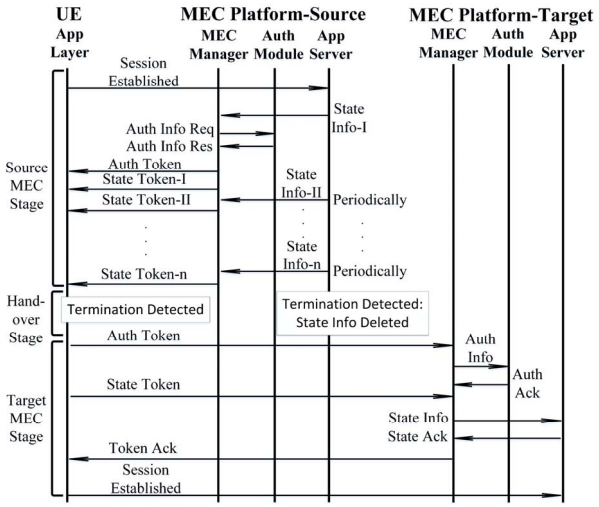

Fig. 4. TS3A message flow

on the basis of the authentication information. The MEC manager will also send the session cookie to the MEC manager of the source MEC. The source MEC manager already has the state information and when it receives the session cookie, it will send the session state information to the MEC manager of the target MEC. As the session state information is received, it is passed by the MEC manager to the application server in the target MEC, which responds with a session ACK message. After that, the MEC manager informs the application client in UE about the authentication and session ACK via token ACK, and hence a session is established between the application client and application server in the target MEC.

TS3A (Token-based State transfer & 3rd-party Authentication)

This approach is represented in Figure 4. In the source MEC stage, the application client establishes a session with the application server in the source MEC. As soon as the session is established, the application server sends the session information to the MEC manager. As soon as the session information is received by the MEC manager, it sends a request to the authentication module to obtain the authentication information about the user. The authentication module has the authentication details and responds to the MEC manager with the authentication information. The MEC manager now has the session state and the authentication information, and will create a token on the basis of the authentication information and present this token to the user equipment. Note that the MEC in this case does not include the state information in the token as it is not the complete information. The application server will keep sending the state information periodically. The MEC manager will keep generating the state token periodically until the UE becomes detached from application server in the source MEC. The application client and application server will detect the session termination and protocol will enter the handover stage. The application server will remove the state information and the user will move away from the source MEC. The next stage is target MEC stage where UE will present the authentication token and the state token to the target MEC, which will be received by the MEC manager of the target MEC. The MEC manager will extract the authentication information from the token and send it to the authentication module of the target

MEC, which will authenticate the user on the basis of that information, and will send an ACK to the MEC manager. The MEC manager will then send the state token to the application server which will respond with an ACK. After that, the MEC manager informs the application client in UE about the authentication and session state information ACK via token ACK and hence a session is established between the application client and application server in the target MEC.

C. Security Analysis

As the UE is assumed to be malicious and can attempt to attack the core network or the MEC in various ways. The UE is presented with the token which contains the authentication information and state information in the case of TS3A, and cookie information in the case of TC3A. The most important things are confidentiality and the integrity of the token. In order to provide that, we encrypt the token and also sign it by using symmetric keys and public key infrastructure respectively. In this way, if the user tampers with the token in the middle, the receiver will know that the contents of the token have been modified. The UE can also generate fake malicious tokens on its own. This can be easily detected at the target MEC by checking the signature and ID of the source. If the token was generated by the source MEC, it would have its ID and would also be encrypted and signed by the source MEC. If that is not the case, the target MEC can discard the token knowing that it was maliciously generated by the UE. This shows that TC3A and TS3A are also robustly resistant to these posed threats.

V. Implementation

The implementation was carried out assuming that we already had the underlying cellular network connecting the MECs. This is a valid assumption for experiment as TS3A and TC3A interact with the client application in UE and application server in MECs. Source and target MEC server were setup and credentials of the UE were provided to the source MEC and UE was given access to the resource sever which was chosen to be a video streaming server. We assumed that the same video streaming server is deployed in both the source and target MECs. The UE accessed the video streaming server in the source MEC and moved from the source MEC to the target MEC. The assumption was that the underlying cellular network will let the UE know that the eNB has been changed now and hence the source MEC is no longer available. The UE then established the connection with the target MEC which had the same video streaming server and authentication with target MEC was done through the token that was granted to the UE by the source MEC. The session state was transferred after the authentication was completed, and hence the session with the target MEC was established from the same place where it got disconnected with the source MEC. The modules designed for the solution implementation were an authentication module and video server together with an MEC manager. The Auth server shown in Figure 5 contains both the MEC manager and the authentication module which handles all the authentication information in both the source and target MECs. The testbed for the experiment consisted of 3 PCs which worked as UE, home MEC and foreign MEC, and two Access Points. The UE was implemented in one system and the Source and target MECs were implemented in one system and the Source and target MECs were implemented in two other computer systems. All 3 computers were Intel Core i7-8700, 16GB RAM and 64-bit operating system. The video servers were developed using node.js and authentication servers were built by using ASP.net. Angular js was used for developing the UE client. The testbed is shown in Figure 5.

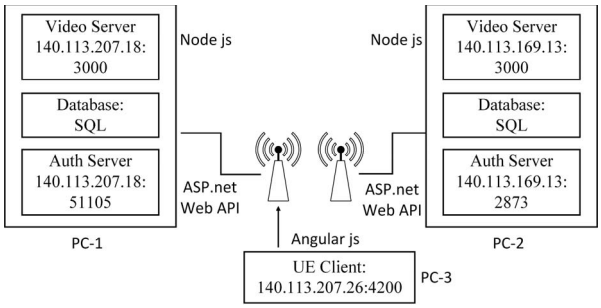

Fig. 5. Experimental Testbed

VI. Results And Evaluation

We obtained four sets of results from the experiment. The first result is the comparison between the Authentication time taken by the source MEC with and without token. When a user authenticates with the source MEC for the first time, it needs to provide its credentials for authentication. Then, server provides a token to the user and the user sends this token along with requests, and hence achieves low latency. This graph shown in Figure 6 does not involve the handing over of user from source to target MEC as it only compares the difference between the first-time authentication with credentials and authentication with token. The results show that the authentication time taken by the server while using token is considerably less than the time taken for authentication without token under different traffic loads.

Time taken by the Target MEC for user authentication and video resumption is shown in the Figure 7 where it can be clearly seen that the TS3A takes less time as compared to TC3A but, the advantage of TC3A is the video resumption from the exact same place where it was terminated in the source MEC. This feature of TC3A retains the state in its entirety and hence TC3A is useful for such applications that cannot afford the state loss. Figure 8 shows the comparison between TS3A, TC3A and simple login method. The graph shows the time taken by these approaches to authenticate the user with target MEC and resume the video. TS3A provides authentication and resumes video in less time as compared to other two approaches. Although, there is not much difference between the latency of TC3A and the simple login method, it should be noted that TC3A also provides application mobility which enables the session to be resumed from the exact same place in target MEC where it was terminated in the source MEC. This service is not available in the simple login method. We also obtained the latency profile for the three methods and assessed time taken by the different segments. The results are shown in Figure 9, where it can be seen that initially all three methods take similar time in the source MEC, but as they move from the source MEC towards the target MEC, the login method needs to provide credentials again, and hence the third segment takes more time as compared to the other segments. The Figure 9 also shows that TS3A reduces latency by approximately 25% and TC3A reduces latency by approximately 4.6% as compared to simple login method.

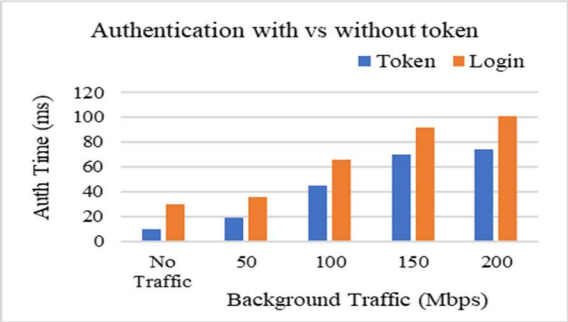

Fig. 6. Authentication with token and without token

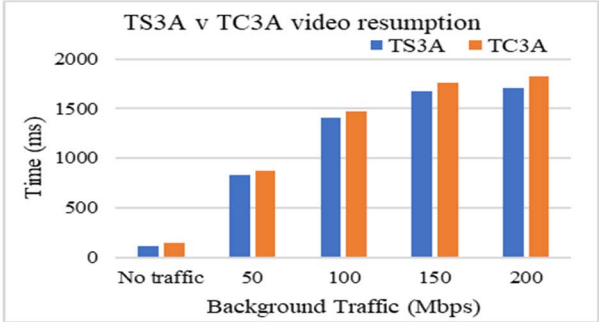

Fig. 7. TC3A v TS3A video resumption time

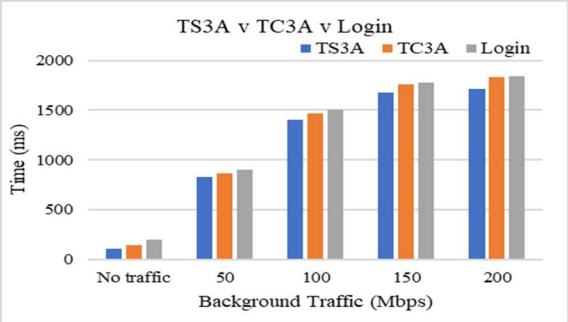

Fig. 8. TS3A v TC3A v Login time comparison

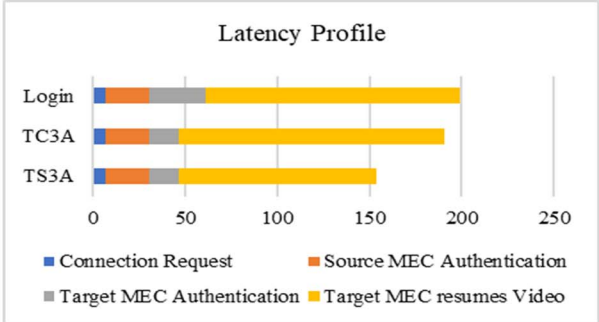

Fig. 9. Latency Profile

VII. CONCLUSION

MEC technology is one of the key technologies in 5G networks as it supports applications with low latency requirements. In the near future, there will be many users accessing application servers in an MEC instead of routing through core cellular network to the internet. Mobile users will face the issue of authentication and application mobility as they would not want to lose a session with the application server in the MEC, and would also not want to provide authentication information repeatedly. In order to address these issues, we proposed two approaches, TC3A and TS3A, which provide the seamless transfer of authentication information and application session of user from one MEC to another MEC while achieving low latency. The results show that TS3A reduces the latency by approximately 25% as compared to simple login and is suitable for the applications that have tighter latency constraints but can afford a little loss of session state. TC3A reduces the latency by approximately 4.6% as compared to simple login and is suitable for the applications that cannot afford even a small loss in the session state but have comparatively loose latency constraints.

REFERENCES

[1] ETSI, Mobile Edge Computing; A Key Technology towards 5G. ETSI White Paper No. 11, 2015.

[2] 3GPP, Technical Specification Group Services and System Aspects; System Architectuer for the 5G Systems: Stage 2 (Release 15). 3GPP Standard TS23.501 V0.4.0, 2017.

[3] ETSI, MEC Deployment in 4G and Evolution towards 5G. ETSI White Paper No. 24, 2018.

[4] Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017). On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. IEEE Communications Surveys & Tutorials, 19(3), 1657-1681.

[5] Bradley, J., Sakimura, N., & Jones, M. B.. JSON web token, 2015

[6] Hardt, D., & Goland, Y. Simple Web Token (SWT). Version 0.9, 5, 1427, 2009.

[7] Monzillo, Ronald, et al. Web Services Security: SAML Token Profile, 2006.

[8] Mun, Hyeran, Kyusuk Han, and Kwangjo Kim. "3G-WLAN interworking: security analysis and new authentication and key agreement based on EAP-AKA." 2009 Wireless Telecommunications Symposium, WTS 2009. IEEE, 2009.

[9] Shi, Minghui, et al. "A service-agent-based roaming architecture for WLAN/cellular integrated networks." IEEE Transactions on Vehicular Technology 56.5 (2007): 3168-3181.

[10] Jiang, Yixin, et al. "Mutual authentication and key exchange protocols for roaming services in wireless mobile networks." IEEE Transactions on wireless communications 5.9 (2006): 2569-2577.

[11] Shi, Minghui, Xuemin Shen, and Jon W. Mark. "IEEE 802.11 roaming and authentication in wireless LAN/cellular mobile networks." IEEE Wireless Communications 11.4 (2004): 66-75.

[12] Kahvazadeh, Sarang, et al. "Securing combined fog-to-cloud system through SDN approach." Proceedings of the 4th Workshop on CrossCloud Infrastructures & Platforms. ACM, 2017.

[13] Grasa, Eduard, Miguel Ponce de Leon, Sven van der Meer, Diego Lopez, and Miguel Tarzan. "Open multi-access edge computing and distributed mobility management with RINA." In 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 1-2. IEEE, 2017.

[14] Zhang, Ping, Mimoza Durresi, and Arjan Durresi. "Multi-access edge computing aided mobility for privacy protection in internet of things." Computing 101, no. 7 (2019): 729-742.

[15] Subramanya, Tejas, Giovanni Baggio, and Roberto Riggio. "lightMEC: A Vendor-Agnostic Platform for Multi-access Edge Computing." In 2018 14th International Conference on Network and Service Management (CNSM), pp. 198-204. ieee, 2018