Enhanced LISP Mapping System for Optimizing Service Path in Edge Computing Environment

KYOUNGJAE SUN AND YOUNGHAN KIM, (Member, IEEE)
School of Electronic Engineering, Soongsil University, Seoul 06335, South Korea
Corresponding author: Younghan Kim (younghak@ssu.ac.kr)

This work was supported in part by the Institute of Information & communications Technology Planning & Evaluation (IITP) funded by the Government of Korea (MSIT) (Development of Fast and Automatic Service recovery and Transition software in Hybrid Cloud Environment) under Grant 2020-0-00946, and in part by the Information Technology Research Center Support Program supervised by IITP under Grant IITP-2020-2017-0-01633.

ABSTRACT

In the evolution of mobile networks, edge computing is a key technology, which provides services with proximity to users and enables various services to be operated in a virtualized platform for mobile users. Particularly, services commonly used by multiple mobile users can be simultaneously operated at multiple edge nodes, which allows users to use the service of the closest node to their access point. However, to provide session continuity for a service when the user moves, a method for optimizing the service path along with the mobility management of the user is required. In this paper, we propose an enhanced control system using the locator/identifier separation protocol to provide service path optimization with mobility management. This was achieved by integrating the location information of the service and the session information between the service and the users. The operating cost and handoff latency were analyzed. The analysis results show that the proposed system can provide an optimal service path to mobile users through fewer message exchanges and less delay than existing approaches.

INDEX TERMS

Edge computing, locator/ID separation protocol (LISP), optimal service selection.

I. INTRODUCTION

Edge computing has brought the cloud computing infrastructure closer to users, thereby reducing service latency and unnecessary traffic to core cloud centers. Considering these advantages, mobile networks have adopted the edge computing technique, namely, multi-access edge computing (MEC) [1], [2].

In a mobile network, when a user moves at a high speed and needs to continue using the edge computing services, the service latency and optimized traffic path that are initially provided cannot be guaranteed as the user moves farther away from the original edge computing node. To solve this problem, service migration techniques have been proposed to move the service from an original edge node to an edge node that is closer to the user’s current location. During service migration, the service that is migrated to the new edge node will receive a new IP address, and the service session should be restarted; this causes a certain service interruption and delay.

To reduce service interruptions from changing the IP addresses, when considering service edge nodes as other mobile end users, general IP mobility management protocols have been proposed as a basis for the service migration method [5]–[8]. The basic principles of IP mobility management are to maintain an IP address when a node moves to another location and to redirect the traffic from the IP anchor of the migrated service to the changed location. Several methods were initially proposed based on the existing centralized IP mobility management protocols, such as Mobile IPv6 (MIPv6) [3] and Proxy Mobile IPv6 (PMIPv6) [4]. With these methods, when a service is migrated to another edge node, the traffic is redirected from the IP anchor of the migrated service to the changed location [5], [6]. However, these methods face certain challenges, such as a bottleneck of the centralized anchor and a non-optimized traffic path. Instead of applying a centralized anchor, the Distributed Mobility Management (DMM)-based solution is proposed, in which the anchor functions are distributed to various locations in the network [7]. However, even if a service is migrated to the optimal location according to the user’s movement, the actual path can be made longer if the traffic is redirected to the original anchors.

To optimize the routing path of the traffic in the DMM-based method, solutions applying the Software-Defined Networking (SDN) concept were proposed [8], [9].
However, this method increases the complexity when the number of migrations increases because the routing information of the network must be updated by an SDN controller whenever the service location is changed.

In the aforementioned studies, the traffic route is unoptimized because of the dual context of the IP address containing both the locator and the session endpoint identifier. If the context of the IP address is separated into locators and identifiers, it is possible to provide optimized traffic routes using locator values that vary depending on the location, while maintaining the session using the fixed identifier of the session endpoint. Based on the Locator/Identifier Separation Protocol (LISP), which is a representative ID/location separation protocol [10], several methods have been proposed for service migration in a distributed edge computing architecture. According to [11], when a mobile user moves and a service running on the edge is migrated to an edge closer to the user, the traffic path is optimized by updating the locator of the user ID, as shown in Fig. 1. The authors of [12] proposed a similar approach by adding a control entity to handle locator updates during service migration. In these proposals, the traffic paths are optimized using the concept of ID/location separation. However, such an approach only considers if the service needs to be migrated to a location closer to the user.

In a distributed edge computing architecture, equivalent services are usually available on different edge nodes. The authors of [13] also proposed a method for selecting the optimal service node using a service migration management system and a network control system at multiple edges. However, because of the message exchange and delay in the decision-making process across multiple entities, the service downtime increases when users move frequently or at a high speed. In this case, instead of migrating the service, it would be more effective for the user to connect from the newly moved location to the newly available optimal edge node where an equivalent service is operating. For equivalent services running across multiple edges, many studies have been proposed to select or determine the optimal service location for mobile devices, especially focusing on time-critical applications and rapidly moving devices such as vehicles. In [14], they proposed a latency-aware mashup algorithm for micro-service based mobile edge computing, which guarantees service requirements for latency with minimizing network resource consumption. For the vehicular network, authors of [15] designed a distributed routing algorithm, which can provide efficient data distribution for high-speed vehicles to give traffic information. With a federated learning algorithm, the author of [16] proposed a dynamic caching update method across the distributed edge computing node for vehicular devices. For smart cities, the author of [17] designed architecture to collaborate service management in the edge cluster, which determines appropriate location service functions requested by the user. However, despite these efforts, additional operations are necessary to inform the location of a newly selected service for a mobile user’s current location and to provide network connectivity to the changed location.

To select the optimal service among several such edge nodes, various approaches to service discovery technology have been proposed [18, 19]. According to [18], for example, the router to which a mobile user is connected determines the optimal edge node with an available service by exchanging information with the routers connected to the neighboring edge nodes and configures the traffic route with the corresponding router. Alternatively, based on Content-Centric Networking (CCN) technology, the authors of [19] proposed a method for determining an optimal edge node that has an equivalent service using an extended forwarding table implemented in a CCN-enabled router. However, for seamless service resumption at newly connected edge nodes, these methods require the user’s state information to be transferred from the previous service node to the newly serviced edge computing node when the user changes the service node according to the user’s movement.

In addition to the service which communicates between a single mobile user, other service types also can be considered in which the edge computing service is used for in-network computing during end-to-end mobile communication. In Fig. 2, it shows an example of an in-networking service in which multimedia traffic is exchanged between two mobile users with different coding technologies, using transcoding services implemented at an intermediate edge node or computing the assistance service for an Augmented Reality (AR) application running at the edge. In this case, the ID and location information of both mobile users should be provided as service state information to the newly selected edge node. The existing approaches are additionally...
implemented router functions for the management of the state information; however, all state information is managed independently within the distributed edge node and shared by a packet exchange, as required, which may increase the service delay because of the synchronization interval. Thus, for an easy and fast service resumption at the newly connected service, a logically centralized management system is necessary that manages the state information integrated with the location information of equivalent services running on the corresponding edge nodes. Basically, LISP has a logically centralized mapping system that manages the state information, such as the ID and location information; hence, it is suitable for determining the optimal edge computing node and providing the state information by accumulating the user’s state information.

In this paper, we propose an enhanced LISP mapping system that integrates the service information into the user ID and location information to quickly resume the service of mobile users when moving to another location. In the proposed system, information regarding equivalent services distributed in multiple edge computing nodes is managed by an additional service mapping table consisting of the service ID and location information of the edge nodes containing the service. Also, the service ID is recorded in the mapping table used to manage the existing user state information, allowing the optimal location for the mapped service to be found while updating the new location information when the user changes location. The architecture and process of performing service discovery by assigning an ID to services are similar to [17], but our proposal focused on improvement in constructing a service path after discovery using the LISP protocol. Through this integrated information of the user and service, the proposed mapping system can provide information on the newly selected edge node service to the mobile user associated with the service without additional procedures when the user’s location is updated, which enables a rapid resumption of the service session. The results of a performance analysis of our system demonstrate that deploying a service strategy equivalent to the proposed LISP mapping system can reduce both the operating cost for the optimal path selection and latency, allowing service to resume after the user becomes mobile.

The remainder of this paper is organized as follows. Section II details the design of the proposed LISP mapping system and mobility management using a service selection and resumption procedure. Section III presents the performance evaluation results of the proposed concept. Finally, Section IV provides some concluding remarks.

II. PROPOSED LISP-BASED SERVICE LOCATION MANAGEMENT IN EDGE COMPUTING SYSTEM

A. LISP OVERVIEW

In LISP [10], both the routing location and endpoint are represented using the IP address such that the pool of IP addresses is divided into two parts, i.e., addresses for the routing locator (RLOC) and endpoint identifier (EID). The EID is a globally unique address used to establish a session and provide connectivity between endpoints. The RLOC is the address of the access router, and temporarily represents the user’s current location. An ingress tunnel router and an egress tunnel router are used for each LISP site, which can be combined into a single entity called an xTR. For forwarding data packets between the two EIDs, the original IP packet is encapsulated in the RLOC values of the ingress and egress xTR representing the location of the endpoint EIDs and delivery in the network. When the xTR of the destination EID receives an encapsulated packet, it removes the external header and forwards the original packet to the destination EID node.

To manage the location information of all EIDs in the network, the LISP defines the LISP mapping system (LISP-MS) which is stored in the EID–RLOC mapping information. The LISP-MS is a logically centralized system that can be implemented in a physically distributed database such that xTRs can register, release, and query the EID–RLOC mapping information by exchanging LISP control plane messages. For the scalability of the LISP-MS, several distributed mapping system architectures have been standardized as various topology types [20], [21]. As a basic operation of the LISP protocol, if the RLOC value of the destination EID does not exist in the local cache table, the xTR sends a map-request message to the LISP-MS. For a response, the LISP-MS inserts the RLOC of the requested EID into a map-reply message and sends it to the xTR. When a new EID is detected at the LISP site, new mapping information with its RLOC value assigned by the xTR is registered or updated to the LISP-MS using a map-register message. In the current LISP-MS mapping table, the EID-assigned RLOC and several policy metrics (e.g., priority and weight factor) are defined.

B. ENHANCED LISP MAPPING SYSTEM DESIGN

To design the enhanced LISP mapping system (eLISP-MS), we first divide the EID into two types according to the node characteristics: user EID (UID) and service EID (SID). The UID is assigned to a mobile node (MN), which is the same as the original EID definition, that is, it is uniquely identified by the user device or device interface. It is impossible to assign the same UID to different devices or interfaces. The SID is a newly defined ID type, which is an EID assigned to a service that can be deployed and shared with equivalent services running as a VM or container instances in multiple edge nodes. Each virtualized service instance with the same SID is distinguished by an RLOC, which represents the location of the edge node running the service instance. The SID can be mapped with one or more RLOCs, and as the number of service instances increases, the number of RLOCs mapped to the SID increases. Because both UID and SID can be configured irrespective of their location, detailed strategies for an ID allocation are applied according to the policy of the network operators and service providers. For example, a UID can be configured in the form of a private IP address according to the generic mobile network operation [22].

K. Sun, Y. Kim: Enhanced LISP Mapping System for Optimizing Service Path in Edge Computing Environment

IEEE Access
The SID can be assigned using a domain IP address, which is mapped in the DNS system.

Fig. 3 shows the mapping table structure of our proposed eLISP-MS. We separated the LISP mapping system into a single UID mapping table and multiple SID mapping tables. The SID mapping table contains SID–RLOC mapping information to manage the list of its RLOCs of equivalent service instances. Each SID mapping table contains the RLOC mapping information for a single SID or multiple SIDs grouped by a service provider that can exist as a single table. According to Fig. 3, SID#1 and SID#4 manage their own SID mapping tables, whereas SID#2 and SID#3 are managed together. As an advantage of the SID mapping table separation, it improves the reliability and speed of the SID–RLOC mapping information modification by distributing access to the mapping table when modifications frequently occur owing to a dynamic instantiation and release of service instances. In addition, configuring the mapping table for each SID can enhance security through access control and limited authorization to read/write for the mapping table. In the SID table entry, additional option fields are defined along with the RLOC information for the service ID. This option field might indicate the weight parameter or priority information supported by the current LISP mapping table. For the migration of a stateful service, the option field might indicate additional information such as the objectID, which represents detailed session identification; however, examples of option fields are not discussed in detail in this paper. When a service instance of SID#1 is created at a new location and the RLOC is configured, this information is updated to the SID#1 mapping table only. When SID mapping tables (e.g., SID #2 and SID #3 in Fig. 3) contain multiple SIDs, the eLISP-MS can support the scalability and efficiency of the mapping system by integrating SID mappings for services that are managed by the same service provider.

Another enhanced feature of the eLISP-MS is an additional parameter in the UID mapping table. As shown in Fig. 3, the “Mapped SID” field is additionally defined, which represents the binding information between a UID and a SID. This field is updated while exchanging LISP protocol messages between a mobile user and the eLISP-MS, or between the service instance and the eLISP-MS. In addition, the SID value of the corresponding field includes service in the unidirectional or bidirectional traffic paths. In other words, both the UID of the source node of the traffic entering the service and the UID of the destination node of the traffic leaving the service have the same SID value in the “Mapped SID” field. For a scenario based on Fig. 2, when a multimedia session between MN#1 and MN#2 is executed by a transcoder service located at the edge node, the “Mapped SID” field in the UID table of MN#1 and MN#2 is recorded as the SID value of the same service. The advantages of this field extension in the eLISP-MS are simultaneous optimal service node discovery during mobility management procedures and fast recovery of inbound/outbound traffic states in the newly selected service. For the optimal edge node selection, when a mobile node has moved and attached a new xTR, the RLOC for the UID of the mobile node is registered by sending a map-register message from the new xTR to the eLISP-MS. At this time, using only a one-time look-up process, the mapping table updates the UID with a new RLOC value and simultaneously determines an appropriate RLOC value by looking up the SID mapping table of the value recorded in the “Mapped SID” field. Several methods about service selection and optimal service determination algorithms, as described before [14]–[16], it can be used to determine the optimal edge node, and the results can be revealed in the corresponding service mapping table.

In terms of network and edge computing management and orchestration, the proposed eLISP-MS can simplify the interworking process with other management functions for mobile edge computing. Fig. 4 shows the control functions of the edge computing environment, which contains the network management area, edge cloud infrastructure management area, and virtualized service management area. As observed in Fig. 4, each management area can be defined from different aspects, standardized in different organizations, and authorized by different operators. For example, the management of mobile nodes in the network is handled by the network operator’s control plane functions defined in [23] (such as 5G AMP and UDM), whereas service providers can provide an access interface to applications and request deployment of their service instances to the edge clouds. The resource and service management functions in the MEC platform [1] must apply not only the lifecycle of the service instances but also fault management and high availability by monitoring.
their cloud infrastructure. Using these various management functions, the network state of mobile users or services on the edge computing nodes can be changed dynamically, which impacts the complexity of the increased network management and even results in conflicting network policies. Furthermore, from the point of view of real implementation, various network functions must communicate through APIs specified for open-source based software platforms such as OPNFV, Kubernetes. For interoperating with them, a lot of resources and effort are required to develop additional plug-ins or drivers to provide compatibility.

Because the eLISP-MS can separate and manage the UID and SID management entities between the network operators and service providers, it can reduce conflict points for networking management. For example, when the service provider configures the SID and default policy for a service, the MEC orchestrator instantiates a service instance in one or more edge clouds according to their policy and updates the RLOCs of the corresponding instances to the SID mapping table. When the user gains access through a mobile network and requests a service path, the network operator not only updates the UID–RLOC mapping to the eLISP-MS using the mobility management control function it also looks up the appropriate SID mapping table for session establishment. During this process, the mobile network and edge computing management functions look up only the related mapping tables in the eLISP-MS without having to query each other for the state of the mobile user or service network. These operations in the proposed design can be operated sufficiently with LISP protocol rather than specially made API.

C. MOBILITY MANAGEMENT AND OPTIMAL SERVICE SELECTION USING eLISP-MS

In this section, we describe detailed message flows in the eLISP-MS for mobility management and optimal service selection. We consider two scenarios: mobile-to-service communication and mobile-to-mobile communication. For a mobile-to-mobile communication scenario, we consider the invocation of a transcoding service running on the edge node, as shown in Fig. 2. Transcoding invocation procedures were defined in protocols such as the session initiation protocol (SIP) [24]. However, in this paper, invocation procedures are described in general terms.

1) MOBILE-TO-SERVICE SCENARIO

For a mobile-to-service scenario, Fig. 5 shows the initial attachment and service request procedures with the update of the LISP mappings. When a service is activated in a specific edge node, the edge cloud assigns the locator with its gateway xTR address as the RLOC to the service instance. According to the figure, the instance of App#1 is assigned the RLOC of xTR#3, which acts as a gateway for the edge cloud in LISP-site#3. After sending a map-request message, including this mapping information, from xTR#3 to the eLISP-MS, the eLISP-MS adds the received RLOC to the RLOC list in the SID mapping table of App#1. When MN#1 connects to LISP-site#1, xTR#1 detects it, allocates an RLOC, and sends a map-register message with UID mapping information to the eLISP-MS. Thereafter, with additional authentication and controls handled by mobile network functions, MN#1 can communicate through the network. When MN#1 wants to initiate a session, it sends a service request to the network including the requested service (that is, App#1 in this figure). According to this request, xTR#1 must obtain the RLOC of App#1. Hence, it sends a map-request message to the eLISP-MS. When the eLISP-MS receives the request message, it first looks up the SID mapping table of App#1 to determine which RLOC is appropriate for the current location of the MN among all locations of the equivalent services. Under this scenario, the eLISP-MS chooses xTR#3 as a proper RLOC for App#1. Subsequently, the eLISP-MS adds the SID in the “Mapped SID” field to the UID mapping table and sends it to the xTR#1 using a map-reply message. As a result, once the mapping information of [App#1, xTR#3] is stored in the local map-cache entry, packets from MN#1 to App#1 are encapsulated in an outer header, which is destined to xTR#3, at xTR#1.

Mobility management and optimal service selection procedures are shown in Fig. 6. Here, it is assumed that MN#1 is previously connected at LISP-site#1 and has established a session with App#1 at LISP-site#3 according to the procedure described in Fig. 5. When MN#1 moves to LISP-site#2 and attaches to xTR#2, similar to the initial attachment procedure, xTR#2 allocates an RLOC for MN#1 and sends a map-request message with new UID mapping information to the eLISP-MS. Upon receiving the new mapping information of MN#1, the eLISP-MS updates the corresponding RLOC of MN#1 from the existing xTR#1 to xTR#2 in the UID mapping table. An update of the UID mapping information involves a handover of the MN; hence, the eLISP-MS checks the session connection status of MN#1 using the “Mapped SID” field in the UID mapping table. After confirming that the session for App#1 exists in the corresponding field, the eLISP-MS can select the optimal RLOC based on the current location of MN#1 (that is, xTR#2) by looking up the SID mapping table of App#1. The selected

FIGURE 5. Initial service request procedure of mobile-to-service scenario.
SID–RLOC mapping information (that is, xTR#4 under this scenario) is sent from the eLISP-MS to xTR#2 through a solicit map-request (SMR) message to update the information in local map-cache entry of xTR#2, and finally, the session of MN#1 with App#1 is re-directed to the optimized path toward xTR#4.

In the previous LISP mobility management approach [25], the SMR message was forwarded to the xTR of the corresponding node directly from where the MN newly connects; however, the optimal service instance location depended on the movement of the MN. In other approaches that do not separate mobile nodes and services mapping information, they need to query service locator to the mapping system after finishing the handover process so the xTR of MN should send Map-Request message for service locator. It causes increasing the number of LISP protocol message and service downtime between mobility management and service discovery procedure. With this proposal, however, using the SID mapping table that manages the RLOC list of services, the optimal equivalent service instance at the current location can be selected rapidly while registering new mapping information in the eLISP-MS. At the same time as the completion of the access procedure, the eLISP-MS based network simultaneously provides an optimal service path between the MN and service without service downtime, allowing the service requirements to be continuously guaranteed.

For stateful services, additional processes such as a state transfer or service migration are required for maintaining the user states at the new service instance. These additional procedures should be handled collaboratively with a service provider or edge computing manager that handles the service management process. During the process of a stateful service, the eLISP-MS minimizes the operational impacts and conflicts with other control functions by rapidly updating the result of the service management process to the SID mapping system and maintaining them to allow an optimal service location to be continuously selected. With this approach, the service management procedure is the same irrespective of whether it performs independently of the mobility management procedure, or according to the access event, of the MN, that is, the optimal mapping information is updated to the current xTR of the MN. As a result, it can effectively reduce the time required for path optimization, according to the handover, by minimizing the operational complexity as compared with previous approaches.

2) MOBILE-TO-MOBILE SCENARIO

For the mobile-to-mobile scenario, Fig. 7 shows the process of two devices connecting to the network and starting a multimedia session with a transcoding service. Similar to the previous mobile-to-service scenario, when a service instance on a specific edge node is activated, the RLOC information of the corresponding instance is updated into the SID mapping table of the eLISP-MS. When two mobile devices connect to the network from different LISP sites, the xTR address of each LISP site is assigned to the RLOC, and this information is updated to the UID mapping table in the eLISP-MS. When MN#1 sends a session request message to establish a multimedia communication session with MN#2, xTR#1 requests the RLOC information on the current location of MN#2 from the eLISP-MS. Under this scenario, if a transcoding is required because of the mismatch of the codec between the two devices, an application management system (e.g., a subscriber server or a session controller) for multimedia communication can invoke a transcoding service among distributed edge nodes. The method and process of invocation of the transcoding service in the application management system and the selection of the appropriate service node depend on various technologies; hence, they are not specified in this paper. Assuming that App#1 located at LISP-site#2 is selected as a node for transcoding service between the two by the application management system, as shown in Fig. 7, App#1 sends messages to inform MN#1 and MN#2 for transcoding support. For the delivery of the packet destined to MN#1 and MN#2, xTR#2 should request the location information of each mobile user from the eLISP-MS. When the corresponding request message is received by the eLISP-MS, the mapping information for App#1 is updated in the “Mapped SID” field of the MN#1 and MN#2 UID mapping tables to make an alert that both users are using this service. In addition, the mapped SID information is updated to the map-cache entry for the destination node in xTRs by receiving a message sent from App#1. Finally, multimedia packets sent from MN#1 to MN#2 are transmitted from xTR#1 to xTR#2, corresponding RLOC of SID according to the SID defined in the map-cache entry, transcoded in App#1, and transmitted to MN#2 by tunneling from xTR#2 to xTR#3.

For mobility and the optimal service selection process under the mobile-to-mobile communication scenario, Fig. 8 shows the mobility management and service resumption procedures when MN#1 moves from the existing LISP-site#1 to access the new network, LISP-site#4. When MN#1 moves to a new LISP site during communication with MN#2 and receives an RLOC value from xTR#4, xTR#4 sends a map-register message to the eLISP-MS for updating the mapping information of MN#1. Upon receiving this message, the eLISP-MS updates the UID mapping table for MN#1 and checks the value of the “Mapped SID” field to determine whether MN#1 is using the edge service. If there is
a SID value in the corresponding field, the eLISP-MS selects the optimal location among the equivalent App#1 service instances and delivers the selected RLOC value to the xTRs of each endpoint user using an SMR message. Unlike the mobile-to-service communication scenario, the destination of the packet header herein is configured with each endpoint user, and the address of the service is not visible in the packet. Hence, the eLISP-MS should provide the location information of the session endpoint nodes to the xTR of the newly selected service instance. Therefore, when the newly selected App#1 instance is in LISP-site#5, as shown in the figure, the eLISP-MS delivers the location information of MN#1 and MN#2 using an SMR message. After updating the map-cache entry information of the xTRs of each MN and the service instance, traffic may be transmitted between MN#1 and MN#2 through App#1. As a result, by managing all location information of the SID into the SID mapping table, the eLISP-MS can determine the optimal service location of the equivalent service instances with minimized message exchanges.

III. PERFORMANCE ANALYSIS

For the performance analysis of the eLISP-MS, we define a LISP-based network architecture reference model, as shown in Fig. 9. In the figure, the MN and application instances are connected to the LISP network through the corresponding xTRs. For an analysis of two scenarios, \( h_{(S-opt)} \) and \( h_{(S-nonopt)} \) are defined as the service path between the application and the MN. In the mobile-to-mobile scenario analysis, the end-to-end path was calculated as approximately double that of \( h_{(S-opt)} \) and \( h_{(S-nonopt)} \). The number of hops for each route is assumed, as shown in Table I [26].

Based on the reference architecture, we compared the operational costs of our proposal with LISP-MN [25], LISP-FMC [12], and LISP-PACAO [13]. Additionally, even though it is not the LISP-based scheme, we compare reactive intra-handover defined [9] to evaluate performance between our proposal and SDN-based scheme called SDN-MM. In the case of LISP-FMC, it is assumed that the cloud controller function is running in the same place as the mapping system. Because other approaches consider only the service migration scenario, but not the scenario with equivalent services running in a distributed manner, we consider only the operational cost related to the LISP mapping system for the network configuration, and not the costs related to the service migration or optimal service selection process. The mathematical model and all values for calculating the operational cost are referenced from [26]. For the evaluation of SDN-MM, we assume that the Edge Switch (ES) is in position the same as xTR, and SDN Controller as LISP-MS.

A. OPERATIONAL COST ANALYSIS

The total operational cost \( C_{LISP}^{Total} \) is defined as the sum of the location update cost, \( C_{LISP}^{L} \), map discovery cost, \( C_{LISP}^{C} \), and packet delivery cost, \( C_{LISP}^{D} \):

\[
C_{LISP}^{Total} = C_{LISP}^{L} + C_{LISP}^{C} + C_{LISP}^{D}
\]  

(1)

For the mobility pattern of a MN, it is assumed that the direction of the MN is uniformly distributed in \([0, 2\pi]\), based on the fluid flow model. The border crossing rate, \( \omega_c \), for the MN from one LISP site to another LISP site.
\[ \omega_c = \frac{2V}{\pi R_c}, \quad (2) \]

where \( V \) denotes the average velocity of the MN and \( \pi R_c \) denotes the radius of the circular area of a LISP site. Accordingly, the average number of handovers, \( E_{\omega_c} \), crossing the LISP site is defined as follows.

\[ E_{\omega_c} = \frac{\omega_c}{\lambda}, \quad (3) \]

where \( \lambda \) denotes the session arrival rate based on a Poisson distribution. The packet sizes for the LISP protocol are presented in Table II, and the notations for the parameters used in the simulation are listed in Table III.

### TABLE 1. Notations for path hop count.

| Symbol       | Description                                           | Value |
|--------------|-------------------------------------------------------|-------|
| \( h_{MN-xTR} \) | Hop between MN and xTR/ES                            | 1 hop |
| \( h_{App-xTR} \) | Hop between App and xTR/ES                           | 1 hop |
| \( h_{xTR-MS} \) | Avg. hop between xTR/ES and LISP-MS/SDN Controller   | 5 hops |
| \( h_{S-nonopt} \) | Avg. hop of non-optimized service path between App and MN (xTR/ES-to-xTR/ES) | 16 hops |
| \( h_{S-opt} \) | Avg. hop of optimized service path between App and MN (xTR/ES-to-xTR/ES) | 8 hops |
| \( h_{Apps} \) | Avg. hop between equivalent apps (xTR/ES-to-xTR/ES) | 13 hops |
| \( h_{MN} \) | Avg. hop between two MNs (xTR/ES-to-xTR/ES)          | 21 hops |

1) MOBILE-TO-SERVICE SCENARIO ANALYSIS

Location update procedures of the LISP-MN include exchanges of the map-register and map-notify messages between the MN and the LISP-MS because the LISP-MN includes an xTR function. For the map discovery process, the MN sends an SMR message directly to the xTR of the corresponding service instance. In the LISP-MN, there is no path optimization process because the MN always sends an SMR message to the xTR of the service instance that the MN is initially connected to. For delivering the data packets, the MN encapsulates the packet header destined to the xTR of the service instance. The costs of the LISP-MN...
TABLE 2. Notations for packet size of protocols.

| Parameter    | Description                             | Default Value |
|--------------|-----------------------------------------|---------------|
| $V$          | Average velocity of MN                  | 20 m/s        |
| $R_c$        | Radius of a single LISP site            | 1 Km          |
| $E_S$        | Average session length of packet        | 51 bytes      |
| $a$          | Weighting factor for the wired connection| 1             |
| $b$          | Weighting factor for the wireless connection| 1.5          |
| $g$          | Weighting factor for the tunneling overhead| 1.1          |
| $\lambda_s$  | Session arrival rate                    | 0.4           |
| $N_{xTR}$    | Number of edge computing xTR            | 5             |
| $q$          | Probability of wireless link failure    | 0.5           |
| $B_w$        | Wired link bandwidth                    | 30–300 Mbps   |
| $B_{wl}$     | Wireless link bandwidth                 | 10 Mbps       |
| $l_w$        | Wired link delay                        | 2 ms          |
| $l_{wl}$     | Wireless link delay                     | 10 ms         |

TABLE 3. Notation for simulation parameters.

| Message       | Description                             | Size (bytes) |
|---------------|-----------------------------------------|--------------|
| $P_{Map-reg}$ | Map-Register Message Size               | 124          |
| $P_{Map-noti}$| Map-Notify Message Size                 | 128          |
| $P_{Map-req}$ | Map-Request Message Size                | 116          |
| $P_{Map-req}$ | Map-Reply Message Size                  | 120          |
| $P_{Map-sol}$ | Solicit-Map-Request Message Size        | 124          |
| $P_{Map-up}$  | Map-Update Message Size                 | 116          |
| $P_{Map-update}$ | Map-Update Message Size               | 128          |
| $P_{RS}$      | Routing Solicit Message Size            | 52           |
| $P_{RA}$      | Routing Advertisement Message Size      | 52           |
| $P_{PBU}$     | Proxy Binding Update Message Size       | 72           |
| $P_{PBA}$     | Proxy Binding Ack Message Size          | 72           |
| $P_{Flowmod}$ | OpenFlow FlowMod Message Size           | 172          |

are calculated as follows.

\[
C_{LISP-MN}^L = E_{\omega_c} (P_{Map-Reg} + P_{Map-Noti}) \times \alpha h_{xTR-MS} (1)
\]

\[
C_{LISP-MN}^C = E_{\omega_c} (P_{Map-Sol}) (\beta h_{MN-xTR} + \alpha h_{xTR-MS}) (2)
\]

\[
C_{LISP-MN}^D = \lambda_c E_S (\beta h_{MN-xTR} + \alpha h_{xTR-MS}) (3)
\]

In the LISP-FMC, the location information of the MN is updated by exchanging map-register and map-notify messages between the LISP-MS and the xTR of the MN. When the MN moves far from its corresponding service instance, the LISP-FMC migrates the service instance to the optimized location, and the new mapping information is updated by both

xTRs of the service instance before and after migration. The data path is optimized using these procedures. The costs of the LISP-FMC are calculated as follows:

\[
C_{LISP-FMC}^L = E_{\omega_c} (P_{Map-Reg} + P_{Map-Noti}) \alpha h_{xTR-MS} (4)
\]

\[
C_{LISP-FMC}^C = E_{\omega_c} [2 (P_{Map-Up}) (\alpha h_{xTR-MS}) + (P_{Map-Reg}) (\alpha h_{xTR-MS})] (5)
\]

\[
C_{LISP-FMC}^D = \lambda_c E_S (\beta h_{MN-xTR} + \alpha h_{xTR-MS} + \alpha h_{xTR-MS}) (6)
\]

The location update cost in the LISP-PACAO includes the packet exchange process with the LISP-MS in the xTR of the MN, similar to that in the LISP-FMC. For the map discovery process in the LISP-PACAO, the xTRs of all edge clouds in the network share the mapping information of the MN that changes its location, determine the optimal location, and deliver a result as a mapping update request to the LISP-MS. That is, for an MN handover, all xTRs should send and receive the LISP protocol message to one another. The packet delivery path can be optimized using these processes. All costs of the LISP-PACAO are calculated as below.

\[
C_{LISP-PACAO}^L = E_{\omega_c} (P_{Map-Reg} + P_{Map-Noti}) \times \alpha h_{xTR-MS} (7)
\]

\[
C_{LISP-PACAO}^C = E_{\omega_c} [N_{xTR} (P_{Map-Update}) (\alpha h_{xTR-MS}) + (P_{Map-Noti}) (\alpha h_{xTR-MS})] (8)
\]

\[
C_{LISP-PACAO}^D = \lambda_c E_S (\beta h_{MN-xTR} + \alpha h_{xTR-MS} + \alpha h_{xTR-MS}) (9)
\]

In the SDN-MM, the location update procedure includes an attachment procedure for the mobile node. When the mobile node sends the RS (Routing Solicit) message to the new ES, ES exchanges PMIPv6-based signaling with the centralized SDN controller. For configuring the optimal forwarding path of the mobile node, the SDN controller should send OpenFlow FlowMod messages to all switches in the forwarding path to configure the OpenFlow flow table. After configuring all switches, data packets can be forwarded without tunneling between two endpoints. All costs of the SDN-MM are calculated as below.

\[
C_{SDN-MM}^L = E_{\omega_c} (P_{RS} + P_{RA})(\beta h_{MN-xTR}) + (P_{PBU} + P_{PBA})(\alpha h_{xTR-MS}) (10)
\]

\[
C_{SDN-MM}^C = \alpha E_{\omega_c} (P_{Flowmod}) 2\alpha h_{xTR-MS} + h_{xTR-MS} (11)
\]

\[
C_{SDN-MM}^D = \lambda_c E_S (\beta h_{MN-xTR} + \alpha h_{xTR-MS} + \alpha h_{xTR-MS}) (12)
\]

In the proposed eLISP-MS, the cost cannot be distinguished through the location update and mapping discovery process. When new mapping information of the MN is updated from the new xTR to the eLISP-MS using a map-register message, the eLISP-MS sends the result of the optimal path calculated using the SID mapping table lookup to the xTR using an SMR message. Because these two processes can
be conducted simultaneously by a single packet exchange, the eLISP-MS can significantly reduce the operational costs with an optimized path. The cost calculation formula of the eLISP-MS is thus defined by the following two formulas.

\[
C_{eLISP-MS}^{L+C} = E_{ow} (P_{Map-Reg} + P_{Map-Noti}) \times (\alpha h_{xTR-MS})
\]

\[
C_{eLISP-MS}^{D} = \lambda_d E_S (\beta h_{MN-xTR} + \alpha h_{S-opt}) + \alpha h_{App-xTR}
\]

(16)

(17)

2) MOBILE-TO-MOBILE SCENARIO ANALYSIS

Here, because both mobile endpoints can move and change their attachment point, one of the differences from the previous analysis is the average number of handovers, which increases approximately two-fold. In addition, the length of the end-to-end service path between the two endpoints doubles because packets must be forwarded after processing in the intermediate application. In other proposals, because specific information for service between two terminals is not defined in the existing LISP-MS, it is difficult to compare using a clearly defined procedure for each technique. Therefore, similar to this proposal, it is assumed that each technique can also use a network controller connected with the LISP-MS to configure the network and allow applications to pass through the communication paths between endpoints.

In the LISP-MN, when one of the MN conducts a handover, it sends the SMR message to the session endpoint, which is another endpoint MN under this scenario. At the same time, it must send the SMR message to the service instance because the path between the MN and the service instance should be changed. Similar to the mobile-to-service scenario, there is no path optimization in the LISP-MN. The costs of the LISP-MN are calculated as follows.

\[
C^{LISP-MN} = 2E_{ow} (P_{Map-Reg} + P_{Map-Noti}) \times (\beta h_{MN-xTR} + \alpha h_{xTR-MS})
\]

\[
C^{C}_{LISP-MN} = 2E_{ow} [P_{Map-Sol} (\beta h_{MN-xTR} + \alpha h_{S-notopt}) + \alpha h_{App-xTR}) + (2\beta h_{MN-xTR} + \alpha h_{MN-xTR})]
\]

\[
C^{D}_{LISP-MN} = 2\lambda_d E_S (\beta h_{MN-xTR} + \alpha h_{S-opt}) + \alpha h_{App-xTR})
\]

(18)

(19)

(20)

Because the LISP-FMC does not consider the mobility scenario for the mobile-to-mobile communication scenario, it is necessary to newly design a predictable scenario based on the mobile-to-service scenario. The main difference with the previous scenario is the cost of the map discovery, which should be an added procedure through which the LISP-MS sends mapping update results to the xTR of the session endpoint using an SMR message. When the edge node location of the in-network computing service is changed to optimize the service path, the path between the service instance and each of the two mobile nodes will be formed by signaling exchange between the xTR of each node and the centralized controller, which is tightly coupled with the LISP-MS. The cost of the LISP-FMC according to this scenario is calculated as follows:

\[
C_{LISP-FMC}^{L} = 2E_{ow} (P_{Map-Reg} + P_{Map-Noti}) \alpha h_{xTR-MS}
\]

\[
C_{LISP-FMC}^{C} = 2E_{ow} [2 (P_{Map-Up-Reg}) (\alpha h_{xTR-MS}) + (P_{Map-Reg}) (\alpha h_{xTR-MS}) + (P_{Map-Sol}) \times (\alpha h_{xTR-MS})]
\]

\[
C_{LISP-FMC}^{D} = 2\lambda_d E_S (\beta h_{MN-xTR} + \alpha h_{S-opt}) + \alpha h_{App-xTR})
\]

(21)

(22)

(23)

The LISP-PACAO does not consider the mobile-to-mobile scenario. Hence, the scenario was designed based on the existing contents proposed for the mobile-to-service scenario. Because the LISP-PACAO determines the optimal location by signaling exchange between all xTRs when the location of the MN is changed, the map discovery cost is the same as in the previous scenario, except for an increase in the average number of handovers. The cost calculation of the LISP-PACAO is as follows:

\[
C_{LISP-PACAO}^{L} = 2E_{ow} (P_{Map-Reg} + P_{Map-Noti}) \alpha h_{xTR-MS}
\]

\[
C_{LISP-PACAO}^{C} = 2E_{ow} [P_{Map-Update} (\alpha h_{Apps}) + (P_{Map-Noti}) (\alpha h_{xTR-MS})]
\]

\[
C_{LISP-PACAO}^{D} = 2\lambda_d E_S (\beta h_{MN-xTR} + \alpha h_{S-opt}) + \alpha h_{App-xTR})
\]

(24)

(25)

(26)

In the SDN-MM, the only differences from the mobile-to-service scenario are increasing the number of switches to configure OpenFlow flow rules. When the optimal service is selected running at the new location, the routing path will include the selected services from two mobile nodes. The cost calculation of the SDN-MM is as follows:

\[
C_{SDN-MM}^{L} = 2E_{w_s} (P_{RS} + P_{RA})(\beta h_{MN-xTR} + (P_{PRE} + P_{PBA})(\alpha h_{xTR-MS})
\]

\[
C_{SDN-MM}^{C} = 2\alpha E_{w_s} (P_{Flowmode}) 2h_{xTR-MS} + 2h_{S-opt} + h_{xTR-MS})
\]

\[
C_{SDN-MM}^{D} = 2\lambda_d E_S (\beta h_{MN-xTR} + \alpha h_{S-opt}) + \alpha h_{App-xTR})
\]

(27)

(28)

(29)

In the proposed eLISP-MS, an additional procedure over the existing mobile-to-service scenario is the location update process of the new service instance, which is selected as the optimal location by sending an SMR message from the eLISP-MS to the xTR of each mobile endpoint. The calculation for each cost is as follows:

\[
C_{eLISP-MS}^{L+C} = 2E_{ow} [P_{Map-Reg} + P_{Map-Sol}) (\alpha h_{xTR-MS}) + 2 (P_{Map-Sol}) (\alpha h_{xTR-MS})]
\]

\[
C_{eLISP-MS}^{D} = 2\lambda_d E_S (\beta h_{MN-xTR} + \alpha h_{S-opt}) + \alpha h_{App-xTR})
\]

(30)

(31)


### B. Handoff Latency Analysis

The handoff latency is calculated as the sum of the processing delay of the network entities and the transmission delay of a signaling message between them. However, the scope of this study does not include the specifications of the processing methods such as the service migration or the optimal service node determination, but only the definitions of the network aspects that reduce the signaling procedure and enhance the processing of the LISP mapping system. Hence, it is difficult to compare the total latency, including the processing delay. Therefore, in our study, we compared only the transmission delay of the LISP protocol messages as compared to other approaches when considering the aspects of analyzing the network efficiency and operational simplification. The transmission delay of $t_{X,Y}(s)$ between nodes $X$ and $Y$ in a mobile network for a packet of size $s$ can be generally defined as follows [27]:

$$ t_{X,Y}(s) = 1 - q \left( \frac{s}{B_{wl}} + L_{wl} \right) + (d_{X,Y} - 1) \left( \frac{s}{B_{w}} + L_{w} \right) $$

(32)

where $q$ denotes the probability of a wireless failure; $B_{wl}$ and $B_{w}$ are the wireless and wired link bandwidths, respectively; $L_{wl}$ and $L_{w}$ are the delay times of the wireless and wired links, respectively; and $d_{X,Y}$ indicates the number of hops between nodes $X$ and $Y$.

For the analysis, the network transmission delay, $D$, for each approach was calculated, including the location update and map discovery process for each scenario. For example, in the mobile-to-service scenario, the transmission delay for each technique is formulated as follows:

$$ D_{\text{LISP-MN}} = t_{\text{MN,MS}} \left( P_{\text{Map-Req}} \right) + t_{\text{MN,MS}} \left( P_{\text{Map-Noti}} \right) $$

$$ + t_{\text{MN,APP}} \left( P_{\text{Map-Sol}} \right) $$

(33)

$$ D_{\text{LISP-FMC}} = t_{\text{TR,MS}} \left( P_{\text{Map-Req}} \right) + t_{\text{TR,MS}} \left( P_{\text{Map-Noti}} \right) $$

$$ + 2t_{\text{TR,MS}} \left( P_{\text{Map-Up-Req}} \right) $$

$$ + t_{\text{TR,MS}} \left( P_{\text{Map-Req}} \right) $$

(34)

$$ D_{\text{LISP-PACAO}} = t_{\text{TR,MS}} \left( P_{\text{Map-Req}} \right) + t_{\text{TR,MS}} \left( P_{\text{Map-Noti}} \right) $$

$$ + N_{\text{TR,TR,TR}} \left( P_{\text{Map-Update}} \right) $$

$$ + t_{\text{TR,MS}} \left( P_{\text{Map-Sol}} \right) $$

(35)

$$ D_{\text{SDN-MM}} = t_{\text{MN,TR}} \left( P_{\text{RS}} \right) + t_{\text{TR,MS}} \left( P_{\text{PBU}} \right) $$

$$ + t_{\text{TR,MS}} \left( P_{\text{PBA}} \right) + d_{\text{opt}} t_{\text{TR,MS}} \left( P_{\text{FlowMod}} \right) $$

$$ + t_{\text{TR,TR}} \left( P_{\text{RA}} \right) $$

(36)

$$ D_{\text{eLISP-MS}} = t_{\text{TR,MS}} \left( P_{\text{Map-Req}} \right) + t_{\text{TR,MS}} \left( P_{\text{Map-Sol}} \right) $$

(37)

### C. Numerical Results

Based on the aforementioned mathematical models, the results of the total operational costs and the handoff latencies of the proposed eLISP-MS and three other approaches are shown in Figs. 10, 11, and 12. In each graph, the (a) represents the analysis results of a mobile-to-service scenario, and the (b) represents the results of a mobile-to-mobile scenario.

Fig. 10 shows a graph of the total operational cost with the average number of handovers while increasing the session arrival rate from 0.1 to 1. According to the figure, even if the number of handovers increases by more than 100, the proposed eLISP-MS demonstrates a significantly lower value compared to the other three approaches. This is because the eLISP-MS reduces the operational procedure by conducting a map discovery process directly with the mapping update process of the MN using the requested SID field of the UID table and SID mapping table. Further, we can clearly determine that it is possible to reduce the map discovery process in our proposed environment in which equivalent services are distributed to the edge computing environment. In the case of the LISP-PACAO, the service migration determination process is tightly coupled with the mapping discovery process when the MN applies a handover; thus, it results in larger operating costs than the LISP-MN, which does not provide an optimized path. The SDN-MM shows the largest costs even though it does not use tunneling, since it sends FlowMod message to all switches located in the forwarding path to configure OpenFlow flow table rule do it increases map discovery costs. In the case of the LISP-MN, because the MN is involved in the signaling exchanges of the LISP protocol, the transmission of the LISP protocol and the tunneling cost in a wireless section affect the operational costs.

Fig. 11 shows a graph of the total operating cost of four approaches with an increasing velocity of the MN from...
10 m/s to 100 m/s. The increasing velocity of the MN results in frequent handovers between LISP sites and results in a large operational cost that impacts the service downtime and latency. In both scenarios, the results show that the eLISP-MS can achieve both mobility management and service path optimization for an operational cost half of that of the LISP-FMC, which is the lowest operating cost among the three existing approaches. Therefore, the eLISP-MS can perform better than existing approaches, even in a service that requires fast mobility management, such as vehicular devices. This is an important factor to consider in terms of the resource efficiency of edge computing. When a service migration or a selection of the optimal edge service is triggered while the mobile user stays at the current location, and by the time the decision is completed, if the mobile user moves out of the relevant edge coverage, it may result in significantly inefficient resource management and user experience.

Fig. 12 shows a graph of the handoff latency with an increasing wired link bandwidth of 30 to 300 Mbps. Because only the LISP-MS includes a wireless link during the signaling procedure, we used fixed wireless link parameters in this analysis. In the mobile-to-service scenario, all accesses demonstrated a handoff latency of 150 ms or less at a bandwidth of up to 300 Mbps. However, in the mobile-to-mobile scenario, certain approaches still have a delay time of 200 ms or more at a bandwidth of up to 300 Mbps.

According to the graph, the proposed eLISP-MS demonstrates a lower handoff latency in the mobile-to-mobile scenario than the mobile-to-service scenario of other approaches. Because the processing latencies of the other operations are not included, the benefits of this proposal cannot be reliably explained solely by these results. However, results of our approach are clearly shown to increase the network efficiency compared to other approaches by effectively reducing the packet exchange procedure for a location update, optimal service location selection, and path update.

**IV. CONCLUSION**

In this paper, we proposed the eLISP-MS for mobility management with service path optimization in a distributed edge computing environment. In the proposed system, the service binding information is used to enable the service to be executed at the optimal edge node even when the user moves and to enable a fast reconnection between the user and changed service node. In addition, by providing location and ID information of end-users through the mapping system, mobility management and optimal service selection were effectively achieved even in the edge computing service used for the in-network computing of mobile devices. A mapping system that integrates the service information into the user ID and the location information can provide information of a newly
selected edge node service to the mobile user associated with the service without additional procedures when users update their location, thereby allowing a rapid resumption of service session without the service discovery procedures required in the other approaches. The results of the numerical analysis show that the proposed system can reduce both the operating cost for the optimal path selection and handoff latency to resume service after the user mobility. Especially based on results of operational costs with increasing velocity of the mobile node, the effect of reducing the delay time during a handover according to the minimized service search process is expected to contribute in the vehicular network where mobile nodes are moving fast between edges by providing fast reconnection of service with the optimal path. In operational terms, the proposed eLISP-MS can simplify the interworking with the edge computing resource management function by decoupling the management areas. Each edge computing node only needs to update the location information of the changed service to the eLISP-MS according to the resource management or service management processed by the edge cloud management function, and thus a network failure from a collision or asynchronous state can be minimized. To evaluate this advantage, a more detailed design of the interface between these two management functions is needed as a future study.

REFERENCES

[1] Multi-access Edge Computing (MEC); Framework and Reference Architecture, Standard ETSI GS MEC 003, 2019.
[2] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, “On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration,” IEEE Commun. Surveys Tuts., vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.
[3] D. Johnson, C. Perkins, and J. Arkko, Mobility Support in IPv6, document RFC 3775, Jun. 2004.
[4] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, Proxy Mobile IPv6, document RFC 5213, Aug. 2008.
[5] S. Kassahun, A. Demessie, and D. Ille, “A PMIPv6 approach to maintain network connectivity during VM live migration over the Internet,” in Proc. IEEE 3rd Int. Conf. Cloud Netw. (CloudNet), Krakow, Poland, Oct. 2014, pp. 64–69.
[6] Q. Li, J. Huai, J. Li, T. Wo, and M. Wen, “HyperMIP: Hypervisor controlled mobile IP for virtual machine live migration across networks,” in Proc. 11th IEEE High Assurance Syst. Eng. Symp., Nanjing, China, Dec. 2008, pp. 80–88.
[7] J. Ha, J. Park, S. Han, and M. Kim, “Live migration of virtual machines and containers over wide area networks with distributed mobility management,” in Proc. 15th EAI Int. Conf. Mobile Ubiquitous Syst., Comput., Netw. Services, New York, NY, USA, Nov. 2018, pp. 264–273.
[8] S. Garg, K. Kaur, S. H. Ahmed, A. Bradai, G. Kaddoum, and M. Atiquzzaman, “MobQoS: Mobility-aware and QoS-driven SDN framework for autonomous vehicles,” IEEE Wireless Commun., vol. 53, no. 4, pp. 12–20, Aug. 2019.
[9] Y. Bi, G. Han, C. Lin, M. Guizani, and X. Wang, “Mobility management for intro/inter domain handover in software-defined networks,” IEEE J. Sel. Areas Commun., vol. 37, no. 8, pp. 1739–1754, Aug. 2019.
[10] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, The Locator/ID Separation Protocol (LISP), document RFC 6830, Jan. 2013.
[11] P. Raad, S. Secci, D. C. Phung, A. Cianfrani, P. Gallard, and G. Pujolle, “Achieving sub-second downtimes in large-scale virtual machine migrations with LISP,” IEEE Trans. Netw. Service Manage., vol. 11, no. 2, pp. 133–243, Jun. 2014.
[12] A. Ksentini, T. Taleb, and F. Messaoudi, “A LISP-based implementation of follow me cloud,” IEEE Access, vol. 2, pp. 1340–1347, Sep. 2014.
[13] S. Secci, P. Raad, and P. Gallard, “Linking virtual machine mobility to user mobility,” IEEE Trans. Netw. Service Manage., vol. 13, no. 4, pp. 927–940, Dec. 2016.
[14] A. Zhou, S. Wang, and S. Wan, “LMM: Latency-aware micro-service mashup in mobile edge computing environment,” Neural Comput. Appl., vol. 32, pp. 1541–1545, 2020.
[15] A. Ahmad, S. Din, A. Paul, G. Jeon, M. Alokaily, and M. Ahmad, “Real-time route planning and data dissemination for urban scenarios using the Internet of Things,” IEEE Wireless Commun., vol. 26, no. 6, pp. 50–55, Dec. 2019.
[16] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, and M. S. Hossain, “Mobility-aware proactive edge caching for connected vehicles using federated learning,” IEEE Trans. Intell. Transp. Syst., early access, Aug. 31, 2020, doi: 10.1109/TITS.2020.3017474.
[17] Y. Jararweh, S. Otoum, and I. A. Ridhawi, “Trustworthy and sustainable smart city service at the edge,” Sustain. Cities Soc., vol. 62, pp. 1–11, Nov. 2020.
[18] L. Gong and P. Willis, Compute First Networking (CFN) Scenarios and Requirements, document IETF Internet-Draft, draft-geng-tngw-cfn-req-00 (Expired), Nov. 2019.
[19] T.-D. Nguyen, E.-N. Huh, and M. Jo, “Decentralized and revised content-centric networking-based service deployment and discovery platform in mobile edge computing for IoT devices,” IEEE Internet Things J., vol. 6, no. 3, pp. 4162–4175, Jun. 2019.
[20] V. Fuller, D. Farinacci, D. Meyer, and D. Lewis, Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT), document RFC 6836, Jan. 2013.
[21] V. Fuller, D. Lewis, V. Ermangan, A. Jain, and A. Smirnov, Locator/ID Separation Protocol Delegated Database Tree (LISP-DDT), document RFC 8111, May 2017.
[22] Procedures for the 5G System (5GS), document TS 23.502, 3GPP, 2019.
[23] System Architecture for the 5G System (5GS), document TS 23.501, 3GPP, 2019.
[24] G. Camarillo, E. Burger, H. Schulzrinne, and A. van Vijk, Transcoding Services Invocation in the Session Initiation Protocol (SIP) Using Third Party Call Control (3PCC), document RFC 4117, Jun. 2005.
[25] A. R. Natal, L. Jakab, M. Portoles, V. Ermagan, P. Natarajan, F. Maino, D. Meyer, and A. C. Aparicio, “LISP-MN: Mobile networking through LISP,” Wireless Pers. Commun., vol. 70, no. 1, pp. 253–266, Jun. 2012.
[26] E. Seo, V. Zalyubovskiy, and T. Chung, “Mathematical modeling of the scalable LISP-deployed software-defined wireless network,” J. Commun. Netw., vol. 14, no. 8, pp. 727–738, Aug. 2019.
[27] C. Makaya and S. Pierre, “An analytical framework for performance evaluation of IPv6-based mobility management protocols,” IEEE Trans. Wireless Commun., vol. 7, no. 3, pp. 972–983, Mar. 2008.

KYOUNGJAE SUN received the B.S. degree in electronic engineering from Soongsil University, where he is currently pursuing the M.S./Ph.D. degrees with the Department of Electronic Engineering. His current research interests include 5G networking, network function virtualization (NFV), and mobility management.

YOUNGHAE KIM (Member, IEEE) received the B.S. degree from Seoul National University and the M.Sc. and Ph.D. degrees in electrical engineering from KAIST. He is currently a Full Professor with the Department of Electronic Engineering, Soongsil University. He is also the Vice President of the Korea Information and Communications Society (KICS). His current research interests include cloud computing, 5G networking, and next-generation networks.