

Received February 15, 2021, accepted March 23, 2021, date of publication March 31, 2021, date of current version April 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3069974

LISP-Based Integrated Control Plane Framework for Service Function Chaining in Distributed Edge Clouds

KYOUNGJAE SUN^{ID} AND YOUNGHAN KIM^{ID}, (Member, IEEE)

School of Electronic Engineering, Soongsil University, Seoul 06335, Republic of Korea

Corresponding author: Younghan Kim (younghak@ssu.ac.kr)

This work has been supported by the Future Combat System Network Technology Research Center program of Defense Acquisition Program Administration and Agency for Defense Development (UD190033ED).

ABSTRACT Service function chaining, which provides a sequential set of multiple services in an end-to-end communication path, is a key technology for providing better service quality to users and efficient management of the network. Recently, as the services traditionally deployed in the data center have become more proximate to the user using a distributed cloud environment, the management of the service chain has become more complex for connecting services between clouds at a distance. In particular, when a user moves and resumes end-to-end communication at a new location, these complexities may lead to an increase in service downtime, which results in network performance degradation. In this paper, we propose an integrated control system to quickly recover the path to the service chain along with mobility management. The proposed system is based on the locator/identifier separation protocol to provide cost-effective mobility management by integrating the location information of users and services into the service chain mapping system. Based on the analysis of the operating cost and handover delay, it was determined that the proposed system can quickly restore a service chain path between distributed clouds with less signaling exchange and minimum delay when compared to previous approaches.

INDEX TERMS Service function chaining (SFC), locator/identifier separation protocol (LISP), segment routing IPv6 (SRv6), mobility management, control plane.

I. INTRODUCTION

Network service functions are deployed on the network for a variety of purposes, such as security, multimedia optimization, and computing assistance. These functions typically run in the middle of the end-to-end traffic path, and network operators can use them to provide a better quality of service and manage their networks more efficiently. With the recent advancements in cloud and edge computing technology, interest in Service Function Chaining (SFC), an ordered set of service functions that are applied to end-to-end traffic, is also increasing [1], [2]. Mobile network operators or Internet service providers generally deploy SFCs in a geographically centralized manner, such as a data center where service function chains can be easily managed and configured [3], [4]. However, the configuration and management of the service

function chain path in a distributed cloud environment consisting of edge or fog computing are complex, because careful orchestration strategies are required to discover proper services and connect them across multiple clouds operated with different resource management and network policies [5].

In a distributed cloud environment, there are two ways to set up an end-to-end service function chain. The first is a decentralized method that independently configures a part of the end-to-end chain for local service functions that is separate for each cloud [6]-[9]. This approach can provide flexibility in setting up a chain of services because service management tasks that are performed independently in each cloud do not affect the other clouds. However, in this method, packet classification and internal service function path setting are performed independently for each cloud, and this operation is repeated for each cloud; consequently, the delay time for the initial path setting may increase, and overall coordination and consistent performance of the SFC cannot

The associate editor coordinating the review of this manuscript and approving it for publication was Christian Esposito^{ID}.

be guaranteed. Conversely, a centralized way of providing a chain by setting and interconnecting the service functions of each cloud in an orchestration center can quickly configure a chain and make it efficient, taking into account all the states of the entire cloud [10]-[13]. However, even in a centralized model, the service chain path is constructed based on the network state at the initial requested point; therefore, when the user moves, the service chain path must be reconfigured based on the changed network state for user traffic at the new location of the user.

To achieve mobility management of users while maintaining continuity of the SFC, it is necessary to interoperate the mobility management system with the SFC orchestration system. Mobility management approaches based on Software-Defined Networking (SDN) [14], [15] can provide route optimization combined with chain reconfiguration by implementing the service chain management function as an application of a centralized SDN controller. However, this method requires the SDN controller to update the routing information of all switches every time the user moves; consequently, the complexity increases as the frequency of movement of the user increases.

Even if the mobility of the user increases, mobility management can be performed more effectively by separating the context of the IP address into an identifier and a locator without directly updating the routing table of all switches. Moreover, in our previous work, which was based on the Location/Identifier Separation Protocol (LISP) [16], a method was proposed that provides seamless and optimized service function relocation for end-to-end traffic even when the user moves [17]. However, the aforementioned study only considered a single service function of edge computing for end-to-end traffic. In the case of chained multiple service functions, an efficient and coordinated forwarding path setup between service functions is required. This process is composed of two steps. The first step determines the sequential location of each service function, and the second step sets up the forwarding paths between service functions.

In this paper, by extending the previous work, we propose an integrated control system framework that can configure mappings of sequential location information for the service chain path in the LISP control system and provide routing paths of the changed chain with minimal effort. In the proposed system, we define a service chain mapping system consisting of a sequential list of Service IDs (SIDs) and Service Chain IDs (SC-IDs) that are defined by the network policy; further, we implemented a process to determine the actual service chain path by interworking the service chain mapping table with a service mapping system. The actual service chain path that is finally configured by the process between mapping systems is recorded in the mapping system that manages the location mapping of the user, allowing the corresponding chain path to be determined while updating the new location information when the user changes the location. Because we focus on the rapid resumption of end-to-end communication when the user moves, the previous path is restored

first and optimization is permitted later by the additional process of the control system if optimization of the service chain path is required. For path forwarding that is compatible with the legacy routing system and for providing scalability, we applied Segment Routing IPv6 (SRv6) [18] as the routing protocol. Using the SRv6-based routing, the packet can be delivered through the legacy routing system without additional procedures by inserting the entire path information at the packet ingress node; consequently, it is significantly cost-effective for delivering a service chain that has multiple nodes in the middle of the end-to-end path. The performance analysis of this integrated approach has shown a reduction in both the operation cost and latency for the service chain path recovery, thereby allowing the service chain to quickly resume after the user moves.

The remainder of this paper is organized as follows. In Chapter II, we describe the backgrounds of SFC and LISP technologies. Chapter III describes the design and operations of the proposed control plane framework with an extended LISP mapping system. In Chapter IV, we present an evaluation of the performance of the proposed method through numerical analysis. Finally, we conclude the paper in Chapter V.

II. BACKGROUNDS

A. SERVICE FUNCTION CHAINING

The SFC architecture is separated into the control and data planes [1]. The control plane defines a service chain policy to classify the traffic of a user and create a service path, while the data plane delivers classified packets to service functions within the path. When a packet arrives at the network, the ingress classifier matches it to the proper service chain policy, selects the service path, and encapsulates the service path information into the outer header. The encapsulated packet is forwarded to the Service Function Forwarder (SFF), which is connected to service functions, and the SFF delivers the packet according to the service path information in the encapsulation header. For encapsulation of service paths, the Internet Engineering Task Force (IETF) defines the Network Service Header (NSH) [19] that includes service chain information, which is represented by the Service Path Identifier (SPI) and Service Index (SI) values. Because the NSH indicates the next-hop service with a combination of SPI and SI values rather than directly pointing to a specific service function, the SFF should resolve the next-hop service function using a preconfigured service path information or request to the control plane. However, the SFF must maintain service path state information mapped to SPI and SI values in the NSH; therefore, the performance of the SFF may become limited as the number of service paths and traffic increases. In addition, in a distributed cloud environment where the number of SFFs increases, the performance degradation of each SFF affects the scalability of the entire service chain.

To effectively manage the service chain path information to the data plane, a Segment Routing (SR)-based service

path forwarding approach was proposed [13], [20]. SR is a routing mechanism based on the concept of source routing, which provides packet forwarding and traffic steering along a specified network path and includes an ordered list of commands in the packet header [21]. In the SR-based network, traffic forwarding paths are configured as a list of sequential segments, and the list is inserted into the packet header at the ingress node. Because it is not required to maintain the forwarding state in other forwarding nodes in the path, the SR-based service chain configuration can increase scalability by reducing processing loads on the service path at the forwarding node of each cloud [22]. The SR data plane that carries the segment list in the protocol header was defined as two protocols: Multi-Protocol Label Switching (MPLS) [23] and IPv6 [18]. While comparing both protocols, SRv6, which uses an IPv6 extension header to insert a segment list, can simplify the network architecture by avoiding the use of heterogeneous protocols between the local overlay network inside the cloud server and the external IP underlay network. In addition, SRv6 can be extended to support a network programming model to implement complex service function chains with high efficiency, making it more suitable for service path management and configuration than SR-MPLS.

B. LOCATION/ID SEPARATION PROTOCOL (LISP)

When the mobile user moves to a new location and attaches to the network from a different location, there are two issues to solve. The first one is that the session information of the end-to-end flow (i.e., IP addresses of TCP/IP socket peer) is changed so that the session is broken and should be restarted with a changed IP address. Second, if the ingress point of the packet is changed according to the new location of the mobile user, the service chain path information for the traffic of the user does not exist in the newly attached node; consequently, the packet classification must be restarted. Initialization of both the end-to-end session and the service chain path leads to an interruption time during the user handover, which can be a major factor in deteriorating the end-to-end communication. LISP is a protocol that separates the dual IP context of ID and location, which can support end-to-end session continuity regardless of the location through a fixed ID used to establish a transparent session to a network location [16]. In the LISP architecture, users establish end-to-end communication using an Endpoint ID (EID) that is globally assigned as a unique ID, and packets between EIDs over the network are delivered by encapsulating the address of the LISP routers, which is called Routing Locator (RLOC). The RLOC mapping information for the EID is managed by the LISP mapping system; moreover, a LISP router called ingress/egress tunneling router (xTR) updates the mapping information for the EID or resolves the RLOC for a specific EID by exchanging the signaling message with the LISP mapping system. In [24], the authors proposed mobility management to minimize packet loss, in which the LISP mapping system informs mapping changes to the previous

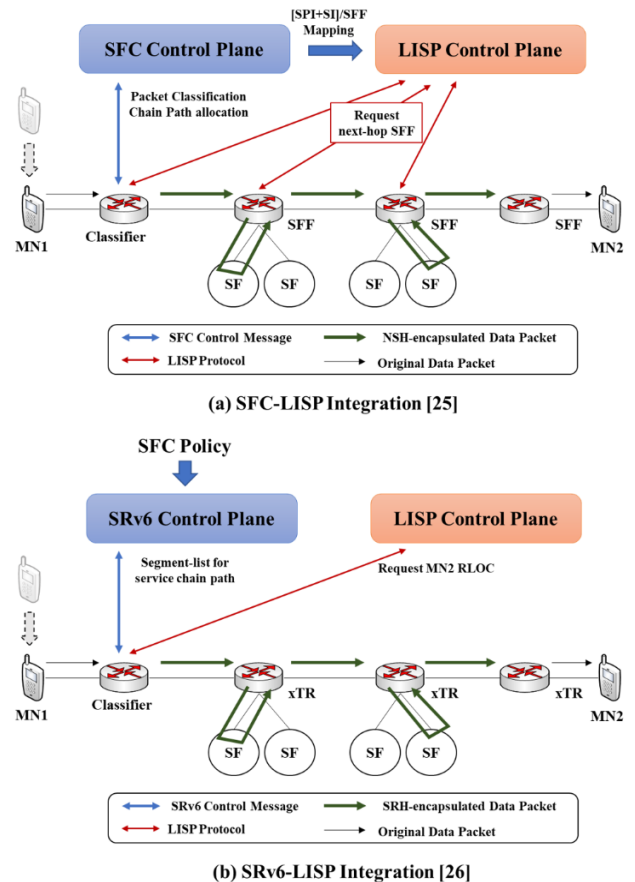


FIGURE 1. Existing locator/identifier separation protocol (LISP)-integrated service function chaining approaches.

xTR when the mapping is updated, and the previous xTR temporarily transfers the packet to the new xTR. However, this approach only considers a solution for packet loss in the backward path, and even in this case, a re-request process for mapping information in the new xTR is required for packets transmitted from the user to the correspondent node. If a service chain is included in the end-to-end path, it is required to request not only the RLOC of the correspondent node but also the service chain path, which may increase service downtime.

In previous studies, scenarios of interworking with the LISP control plane for managing and forwarding the service chain were proposed, as shown in Fig. 1. The first approach [25] is the integration of the LISP control plane with the SFC control plane for the NSH-based SFC configuration, as shown in Fig. 1(a). In this architecture, the LISP mapping system manages the mapping information of the next-hop identifier of the chain path, which combines the SPI and SI values, and its locator, which is represented as the location of the SFF. This approach is effective in resource management of the SFF by providing path state information that was originally distributed independently to each SFF node through a centralized mapping system. However, as signaling between each SFF and the LISP mapping system is

required when the service path is initialized owing to user mobility, a delay may occur until the end-to-end packet path configuration is completed. In addition, this approach does not consider mobility management, such as avoiding packet loss that occurs while delivering to a previous node during mobility. In [26], as shown in Fig. 1(b), the authors proposed an integration architecture for SRv6-based path configuration with the LISP mapping system. In this approach, the LISP mapping system is responsible for managing the ID/RLOC mapping of the endpoint user. After resolving the location of the endpoint, the Path Computation Element (PCE) of the SRv6 control plane configures the segment list for the service path between the two endpoints. Using this approach, even if each forwarding node in the service chain path does not have information about the chain, they can resolve next-hop information for the SFC path by extracting a fully configured segment list from the header. This ensures that the performance degradation of the forwarding nodes in the chain path is avoided and the service chain path at the new ingress node is easily reconfigured after the movement of the user. However, as the LISP mapping system only manages the user EID/RLOC mapping, additional procedures to determine the location of the service and service chain policy are required, which is not clearly defined in the previous study. For mobility management, packet loss was minimized by using the LISP mobility approach [24]; however, this approach still does not support the maintenance of the previous chain state; therefore, the service chain path must be reclassified and recalculated at the new receiving node. Thus, if the previous states for the service chain cannot be quickly applied to the new ingress node, a delay in establishing an end-to-end communication path is inevitable in the mobility scenario.

In a previous work [17], we proposed an enhanced LISP mapping system that allows new ingress nodes to quickly reconfigure the routing state of a moving user, including intermediate services, by extending the User ID (UID) mapping table to include services bound to traffic paths. In consideration of service equivalency, we assigned a globally unique ID to each service and defined a SID mapping system that manages the location information of the service instances of the same SID that are deployed over distributed clouds. Using them, mobility management and path reconfiguration operations could be performed with optimization of service instance locations by exchanging the LISP protocol only once, which can minimize the operation costs for mobility and handover delay. However, the previous work only considered a scenario in which a single service on the end-to-end communication path is considered. To expand to multiple services defined as a service chain, an enhancement of the previous design is required to maintain the chain states that were previously used by a user before moving and adjust it to the new ingress node with minimal effort.

III. PROPOSED SCHEME

The integrated control plane architecture for LISP-based SFC and mobility management is defined as shown in Fig. 2. In our

proposed framework, three different control planes are integrated. First, the service manager is a control plane function used to create a service chain and define classification rules for allocating service chain policy. In the Network Function Virtualization (NFV)-based distributed edge cloud environment, the NFV management and orchestration is defined, which includes the service lifecycle manager that manages the lifecycle of service instances and the VNF Forwarding Graph (VNFFG) management function [11]. The LISP control plane, which is the primary contribution of this paper, includes a mapping system for mobile users, services, and service chains. Each mapping system is running independently; the entries of each mapping system are looked up to obtain necessary information, such as the services in the required chain, location of each service, and endpoint location. The remaining functions are related to the SRv6 control plane, which is defined as the PCE function and topology management function with network topology information. The PCE function calculates the paths of segments including the service function and forwarding node in the entire end-to-end path, creates it as a segment list, and returns the list to the network ingress node, which inserts the segment list to the IPv6 extension header. The segments in the SRv6 path can be defined as not only service instances and edge gateway routers in the network but also intermediate routers between distributed clouds; therefore, the overall topology information of segments is managed by the topology management function.

Fig. 2 also shows the management flow for discovering mobile users, services, and service chains using three types of mapping systems within the LISP control plane. The service instance, which is created by the service manager based on the network policy or dynamic resource management, updates the location information of the created instance to the corresponding SID mapping system. When a service chain is created, which is represented in the service manager as a sequential list of SIDs, a specific chain ID is assigned a unique ID and updated to the service chain mapping system in the LISP control plane. Thus, the service chain mapping system is defined as the mapping of the sequential SID list for a specific SC-ID. To configure the actual forwarding path for a specific SC-ID, the service chain mapping system looks up the location information for each SID mapped to the chain ID in the SID mapping system. This system can allow various placement and path computation algorithms for optimizing service paths, which is not covered in this paper. The SRv6 control plane uses all information including SIDs, service locators, UIDs, and locators from the LISP control plane to create a segment list for the traffic destined for a specific user.

The data plane functions used for requesting the service path and forwarding to the network can be classified as the ingress classifier, service function, and forwarding node. The ingress classifier assigns a specific chain ID to the incoming packet based on the service chain policy and requests for the forwarding path of the classified packet to the control plane.

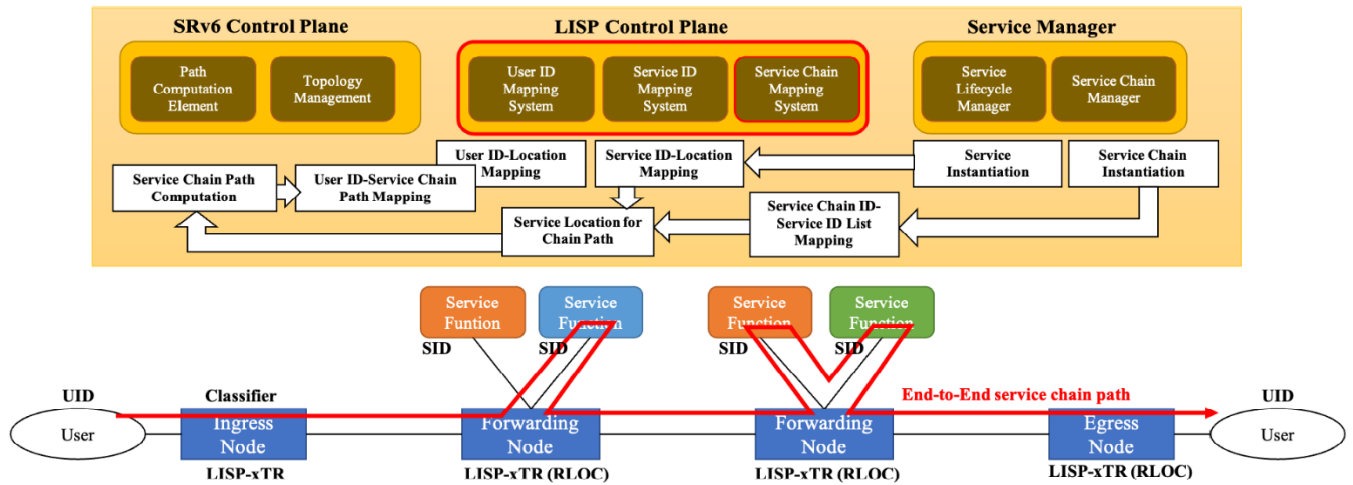


FIGURE 2. LISP-based integrated control plane framework for managing end-to-end service chain path.

From the LISP perspective, all forwarding nodes including the ingress classifiers are performing as the LISP xTR to exchange the LISP protocol with the LISP control plane for updating the user location and requesting the locator of the destination UID of the packet. From the SRv6 perspective, the ingress classifier and forwarding nodes require the capability to read and write SRv6 header information, and the service function considers both the SR-aware and SR-unaware functions.

A. ID, LOCATOR, AND SEGMENTS

Within the proposed framework, three types of IDs are defined for the end-to-end path configuration. The first is the UID, which is globally and uniquely assigned to the mobile user in the form of an IPv6 address. Because the UID is used to identify and establish TCP/IP sessions of end-to-end communication, session continuity is guaranteed as long as the UID is maintained. The Service ID (SID) is an ID uniquely assigned to the service that processes a specific function in the network and it provides service equivalence to multiple service instances operating in multiple locations for the same operation [23]. Thus, service instances instantiated in multiple locations simultaneously are considered with the same SID. Through this SID concept, it is easy to discover services in a network and provide a service connection to an optimized location. In the real environment, the SID can be defined in the form of an IPv6 address or a name in the domain name system. Finally, the SC-ID, which is newly defined in this paper, is a value assigned to a sequential list of SIDs. Even with the same SC-ID, the actual chain path may have a different path, because the SID defined in the SC-ID is independent of location. The SC-ID is defined when the service manager creates a service chain, and it is identified as the chain policy used for traffic classification at the ingress classifier in the data plane.

The locator in the proposed architecture is generally defined as the IP address of the LISP xTR, which is the

gateway of the site where the mobile user or service instance is currently connected. While locating at the mobile user side, a LISP site may include several wireless access points, and the coverage of a single LISP site may differ according to actual implementation. Within a single LISP site, the packet should be delivered internally from the LISP router to the UID, and communication between the LISP router and external network should be possible through general IPv6 routing with the locator assigned to the LISP xTR. The locator address for the SID may be the IP address of the LISP xTR function that is implemented in the gateway of the edge cloud server. Similar to the UID case, the LISP xTR provides local routing for the SID internally.

In terms of SRv6, a segment ID is classified into two types. The first type of segment ID is a locator, which is the IP address of an SR-aware router or LISP router. In the SRv6 environment, the segment ID is defined as a routable IPv6 address in IPv6 infrastructure. When the SRv6 packet is received, the SR-aware and LISP routers read the extended SRv6 header, set the next-hop segment ID as the destination of the IPv6 packet, and forward the packet to the network. In addition, in this paper, we also define the SID as a segment ID. Even though the SID is not a globally routable address, it can be used for local routing within each edge cloud. The segment list configured with SIDs can provide an easy understanding of the service chain in an abstract manner, so it can be easily managed for chain optimization and reconfiguration. Even in the case of an SR-unaware service, the SID is included in the segment list. In this case, the LISP xTR of the edge cloud where the SID is located should operate as an SR proxy. Detailed operations for this will be described later.

B. LISP-BASED SERVICE PATH RESOLUTION

Fig. 3 shows the operation between the ingress classifier and the control plane when a user (UID1) delivers a packet to the network for communication with another user (UID2).

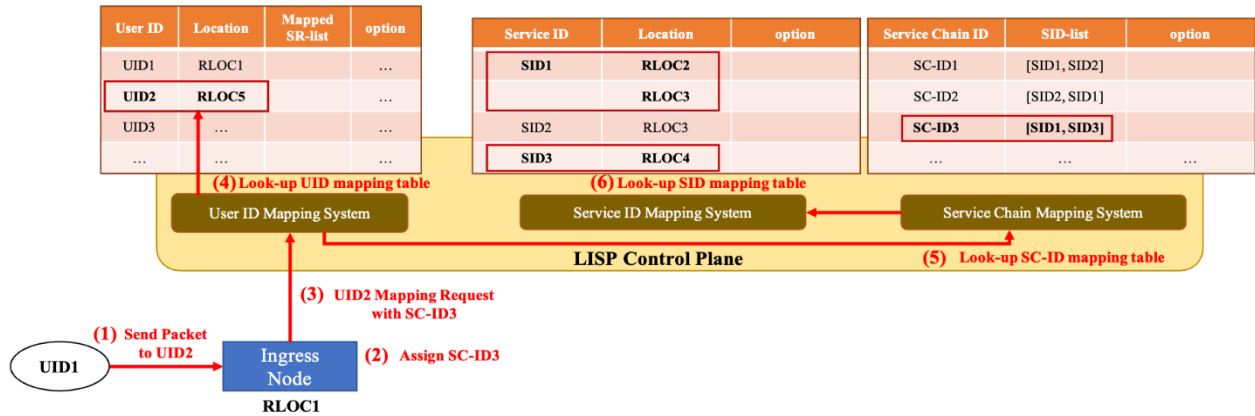


FIGURE 3. Service chain path resolution with sequential look-up process of the proposed LISP control plane.

When the user starts end-to-end communication by sending a packet destined to a UID of the corresponding node, this packet should be classified first at the LISP xTR of the sender, which is the packet ingress node in the network. If the service chain is not required, only the RLOC for the destination UID must be determined; however, when the service chain is required, the ingress node should classify packets based on not only the destination UID but also more detailed information (e.g., 5-tuples of the packet header). For packet classification, when the service chain manager creates the service chain and assigns an SC-ID to the chain, the classification rule along with the SC-ID for this chain is delivered to the ingress classifier. After determining the SC-ID for the packet sent by the user, the ingress classifier sends the Map-Request message to resolve the RLOC of the destined UID of a packet to the control plane, which includes the determined SC-ID. When a Map-Request message is received, the LISP control-plane starts to resolve the RLOC of the requested UID in the user ID mapping system. At the same time, the service chain of the received SC-ID is resolved. Using the received SC-ID, the corresponding SID list can be obtained in the service chain mapping system. This SID list represents only logical service connectivity; moreover, to configure it as a set of service instances with real geographic locations, the SID mapping system looks up RLOCs of each SID. At this time, if there are multiple locators in each SID, an additional process of determining an optimal service locator for a service chain is required. Determining the optimal location of services is required not only for the end-to-end communication distance, but also for factors such as the resource state of each service instance, and this decision algorithm is not covered in this paper.

If the list of service RLOCs in a chain is determined, this can be configured as an essential path of the segment list for end-to-end communication. Thus, with this information, the LISP control-plane requests a full segment list of the service chain to the PCE of the SRv6 control plane. Fig. 4 shows an example of an actual forwarding path with the

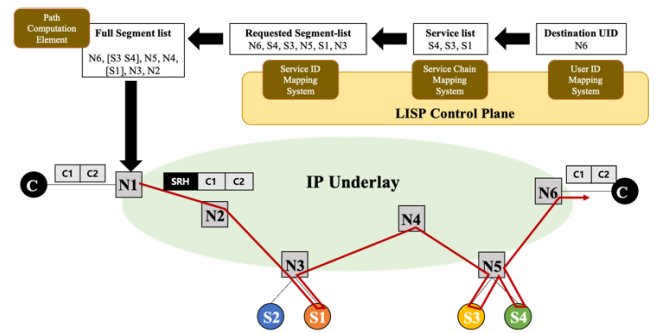


FIGURE 4. Segment routing IPv6-based path configuration and forwarding example.

SRv6 header configuration. If SIDs in the service chain and the RLOC of the corresponding SIDs are defined in the form of a segment list, the entire path of end-to-end communication calculated at the PCE is returned to the segment list with additional values for the SRv6 header information. That is, the segment list and additional values configured through PCE are used to encapsulate the SRv6 header directly at the ingress classifier so that the service chain path for the corresponding traffic is delivered using this. In Fig. 4, for the service chain between C1 and C2 going through S1, S3, and S4, in addition to N3 and N5, which are RLOCs where each service is located, N2 and N4 are additionally selected by path calculation in the PCE. While using this SRv6-based service chain path forwarding, because SID is defined as the segment ID, it supports compatibility between internal routing and external routing at the edge cloud. In the xTR function of each edge cloud server, if the next segment in the segment list is the SID value, the xTR forwards the packet to the corresponding service instance through local routing using SID. Conversely, if the next segment is RLOC, the corresponding IP address can be inserted as the destination and delivered through normal IPv6 routing. Finally, the entire chain path can be forwarded with only the IPv6 extension header without translation between heterogeneous transport protocols.

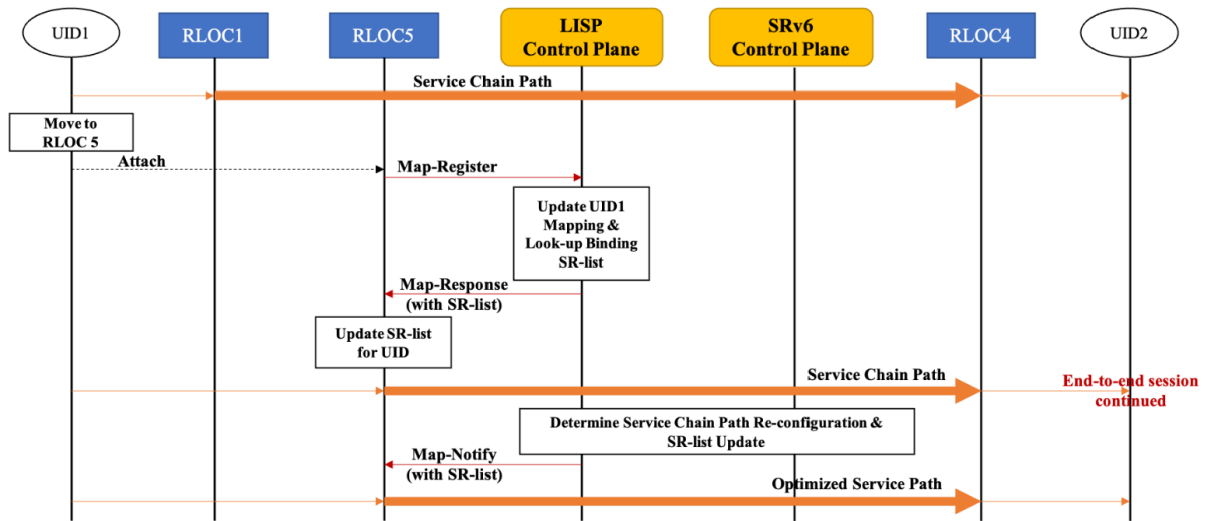


FIGURE 5. Message procedures for updating location of moving user ID and recovering service chain path.

C. MOBILITY MANAGEMENT WITH SERVICE CHAIN RECOVERY

When the user moves from the existing location and connects to the new LISP xTR, the UID-RLOC mapping should be updated to the LISP control plane by the new xTR node, and the end-to-end communication path of UID should be changed accordingly. The mobility management process should be supported for each communication direction. In particular, as a unidirectional chain can be configured, a path recovery method should consider both backward and forward paths. The existing mobility management scheme defined in [22] is intended to minimize the loss of backward packets to the mobile user. In this approach, when the new xTR updates the mapping of the UID to the LISP mapping system, the mapping system sends the Map-Notify message to the previous xTR for temporary rerouting of packets to the new xTR. At the same time, the previous xTR sends the Solicit Map-Request (SMR) message to the corresponding xTR when it receives packets destined for the moved UID, which notifies the corresponding xTR to request changed mapping for the destination UID. This method can minimize the packet loss; however, if the distance on the end-to-end communication path increases, the transmission delay of the SMR messages between xTRs increases; therefore, a longer time is required to request new mapping information of the UID. In the meantime, packets destined for the UID must be forwarded continuously to the previous xTR and rerouted to the current location, resulting in an inefficient packet path. While considering the packets sent by the moving user, the new xTR can request mapping information for the packet to the LISP mapping system only when the data packet is received in the new xTR, and the previous flow states are not maintained. In the previous study [17], this state information was extended to the UID mapping system by providing the binding information for a single service by extending a

field called “Mapped SID” so that the session state can be maintained when the user location is updated. In this paper, we define a “Mapped SR-list” field instead of the “Mapped SID” field in the UID mapping system. This field stores the segment list of the service chain path that is calculated by the PCE function of the SRv6 control plane. When the LISP control-plane receives the Map-Register for updating the UID from the new xTR, it returns the previous segment list of the service chain path together with a response pertaining to the mapping update so that it can quickly restore the states of the service chain path for the current flow. The entire chain path can be delivered to the ingress node through LISP messages with an extension of the LISP Canonical Address Format (LCAF) for segment routing, which is defined in [26].

Fig. 5 shows the user mobility procedure by extending the “Mapped SR-list” field. When the Map-Register requires the mapping update of UID1-RLOC5, the UID mapping system looks up the table entry of UID1 and extracts the SR-list if this field is not empty. By applying this segment list to the new xTR through the Map-Response message, this segment list is inserted into the corresponding packet and forwarded to the service chain path, which is configured at the previous xTR. Additionally, because the existing segment list may not be the optimized path based on the new location of the user, our proposal restores the previous path first, and the new SRv6 header information can be generated if necessary, through an optimization process. If the new segment list is configured, it records in the “Mapped SR-list” field of the user mapping entry, and it is delivered to the new xTR of the user for changing SRv6 header information. With the advantage of SRv6-based path configuration, service chain paths for the flow are fully configured only at the ingress node without other forwarding nodes; therefore, it is possible to change the path quickly and there is no impact on the

performance of end-to-end communication. Without SRv6 integration, as proposed in [25], as the number of services in the chain path increases, a large amount of time is required to reconfigure the entire path. Even when using SRv6 integrated with LISP, as proposed in [26], this scheme does not maintain previous service chain states; therefore, packet classification should restart at the new xTR whenever the user moves. Our proposed mobility management scheme that is integrated with SRv6 and with an extension in the field in the user mapping system can solve the limitations of both previous approaches. In addition, by separating the optimization process of the chain path from restoring the previous service chain path, this method can reduce the performance impact caused by the computation of the optimized chain path of end-to-end traffic, which becomes more complex as the number of services in the chain increases.

D. ADDITIONAL SUPPORTS

The proposed scheme basically considers service functions that are SR-aware. Thus, it is possible to handle the SRv6 header information in a function that is additionally implemented in the service instance, to set the next-hop segment ID as the destination of the IPv6 header in the segment list in the header, and to send the packet. In this case, even SID values that are being used simultaneously in several clouds are extracted as a next-hop segment from the xTR of each cloud and transmitted through local routing. Moreover, the packet from the service function, which can change the destination field of the SRv6 header as the next segment in the list, can be delivered whether its destination segment ID is SID or RLOC. However, among the third-party services, there may be services driven by SR-unaware services. In this case, the LISP control plane can allocate SIDs to manage the location of each service instance; however, these service instances can handle SRv6 headers. For an SR-unaware service, it is possible to implement the SR proxy function [28] in addition to the xTR function in each edge cloud server. The SR proxy function extracts the SID, which is the next-hop segment in the SRv6 header, and after removing the SR extension header, it delivers the original packet to the SR-unaware service instance. After receiving a packet from the SR-unaware service, the SR proxy encapsulates the SRv6 header for the packet by changing its destination to the next segment in the list. If the SR proxy function is implemented on the xTR in the edge cloud, the service chain path including the SR-unaware service can be supported without any additional modification or extension of the proposed architecture. An SR-unaware service can be displayed and specified using the option field of the mapping table.

An additional consideration in service chain management is the flow affinity, which requires the maintenance of the same service chain path whether the user moves or not. In this case, the flow affinity should be explicitly indicated in the process of initially creating the segment list for the service chain, so that even when the user moves, it allows us to

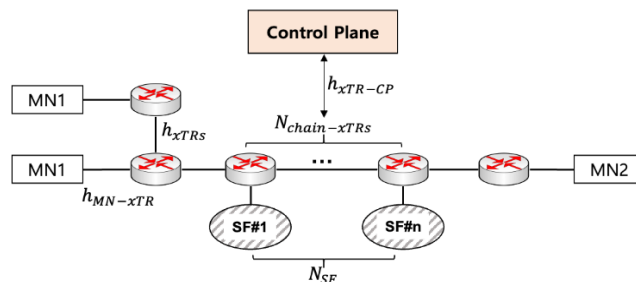


FIGURE 6. Reference architecture for performance analysis.

check the flow affinity of the previous service chain path. Fortunately, as our proposed method restores the previous service chain path first and then determines chain optimization, the flow affinity can be easily provided by marking in the option field of the user mapping system. This option field allows the scheme to avoid determining whether to additionally reconfigure the corresponding service chain that is written in the “Mapped SR-list” field. This is one of the advantages that can be obtained using the proposed scheme, which separates the optimization process from the process of restoring the service chain path.

IV. PERFORMANCE ANALYSIS

To analyze the performance of mobility management in the proposed scheme, we defined a LISP-based architecture reference model, as shown in Fig. 6. In the figure, the Mobile Nodes (MNs) at each endpoint of the end-to-end communication are connected to the LISP xTR of each network; moreover, the service functions are distributed in multiple xTRs between two MNs. As a centralized control plane function, it is assumed that the distance to access the LISP control plane from all xTRs in the network is the same. The number of hops for each route is listed in Table 1. Based on the reference architecture, we compared the operating cost and handover delay of our proposed scheme with the LISP-SFC [25], LISP-SRv6 [26], and MIPv6-based SFC (SFC-IPMM) schemes. In the SFC-IPMM-based environment, the LISP xTR accessed by the user is replaced by the mobile anchor point of each network, and the xTR in the chain path is composed of the SFF defined in the SFC. For mobility management, the SFC-IPMM scheme exchanges Binding Update/Ack messages between the previous anchor and the user, and the SFC reconfiguration is based on the SDN-based approach by exchanging the OpenFlow messages between the cloud gateway and centralized controller. The protocol messages and sizes of LISP, IP mobility, and OpenFlow used in this analysis are listed in Table 2. The mathematical model and all values for calculating the operational cost were referenced from [29] and for calculating handover delay were referenced from [30]. However, as the references did not consider the SFC environment, this study additionally assumed that a service chain including the N_{SF} , i.e., number of services, in the $N_{chain-xTRs}$, i.e., number of xTRs in the end-to-end path, was configured. The operation cost and

TABLE 1. Notations for path hop count.

Symbol	Description	Value
h_{MN-xTR}	Hop between MN and xTR	1 hop
h_{SF-xTR}	Hop between Service and xTR	1 hop
h_{xTR-CP}	Avg. hop between xTR and LISP control plane	5 hops
h_{xTRS}	Avg. hop between two xTRs	3 hops

TABLE 2. Notation for packet size of protocols.

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
$P_{Map-Rep}$	Map-Reply Message Size	120
$P_{Map-Sol}$	Solicit-Map-Request Message Size	124
P_{RS}	Routing Solicit Message Size	52
P_{RA}	Routing Advertisement Message Size	52
P_{BU}	Binding Update Message Size	72
P_{BA}	Binding Ack Message Size	72
$P_{Flowmod}$	OpenFlow FlowMod Message Size	172

handover delay are calculated based on an increase in these values.

A. OPERATION COST ANALYSIS

In [29], the total operating cost C_{LISP}^{Total} for the mobile terminal in the LISP-based environment 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 . However, in this analysis, it is difficult to accurately compare other costs because the packet delivery cost may be significantly larger than other costs with the increase in the service chain path. Therefore, we define the operation cost as the sum of the location update cost and the mapping search cost without the packet delivery cost.

$$C_{LISP}^{Total} = C_{LISP}^L + C_{LISP}^C \tag{1}$$

For the mobility pattern of an MN, it is assumed that the direction of the MN in a single LISP site is uniformly distributed in $[0, 2\pi]$, based on the fluid flow model. The border crossing rate, ω_c , for the MN from one LISP site to another LISP site is defined as follows:

$$\omega_c = \frac{2V}{\pi R_c}, \tag{2}$$

where V denotes the average velocity of the MN and πR_c denotes the radius of the circular area of the LISP site. Accordingly, the average number of handovers crossing the

TABLE 3. Notations for simulation parameters.

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
α	Weighting factor for the wired connection	1
β	Weighting factor for the wireless connection	1.5
δ	Weighting factor for the tunneling overhead	1.1
λ_α	Session arrival rate	0.4
$N_{Chain-xTRS}$	Number of xTR in the service chain path	3
N_{SF}	Ave. number of SF per each xTR in the service chain path	2
q	Probability of wireless link failure	0.5
B_w	Wired link bandwidth	300 Mbps
B_{wl}	Wireless link bandwidth	10 Mbps
L_w	Wired link delay	2 ms
L_{wl}	Wireless link delay	10 ms

LISP site E_{ω_c} is defined as follows:

$$E_{\omega_c} = \frac{\omega_c}{\lambda_\alpha}, \tag{3}$$

where λ_α denotes the session arrival rate based on a Poisson distribution. As a result, each cost can be calculated by multiplying the number of handovers, size of packets for each operation, and hop counts between entities. All notations for the parameters used in the simulation are listed in Table 3.

In LISP-SFC, the location update cost includes the exchange of Map-Register/Reply messages with the LISP control plane at the new xTR of the user. Subsequently, to discover the RLOC of the endpoint for the traffic and service chain path information, the LISP-SFC configures the NSH-based chain path in the ingress classifier and queries the LISP control plane for all SFFs where the services are located. The mapping request process for the next-hop xTR is repeatedly performed at all xTRs in the service chain path. The location update cost and map discovery cost of LISP-SFC are calculated as follows:

$$C_{LISP-SFC}^L = E_{\omega_c} (P_{Map-Reg} + P_{Map-Rep}) \alpha h_{xTR-CP}, \tag{4}$$

$$C_{LISP-SFC}^C = E_{\omega_c} \{ (2P_{FlowMod}) (\alpha h_{xTR-CP}) + (N_{Chain-xTR} + 1) \times (P_{Map-Req} + P_{Map-Rep}) (\alpha h_{xTR-CP}) \}. \tag{5}$$

In the SFC-IPMM scheme, the location update cost according to the MN mobility is defined as the signaling cost of exchanging Binding Update/Ack messages between the

moved user equipment and the previous anchor. Packets coming to the MN arrive at the existing anchor and require rerouting to the location of the current MN; however, packets transmitted from the MN can be forwarded directly to the network from the new access node. The discovery of each SFC component within the chain is handled by the SDN controller, and the cost includes the exchange of the OpenFlow message (i.e., Packet-In and Flow-mod messages) for delivering the processing rules of the NSH packet header to the classifier, SFF, and SF in the chain path. The location update cost and map discovery cost of SFC-IPMM are calculated as follows:

$$C_{SFC-IPMM}^L = E_{\omega_c} (P_{BU} + P_{BA}) (\alpha h_{xTRs} + \beta h_{MN-xTR}) \quad (6)$$

$$C_{SFC-IPMM}^C = E_{\omega_c} (N_{Chain-xTRs} N_{SF} + 1) \times (2P_{Flowmod}) \alpha h_{xTR-CP} \quad (7)$$

In the LISP-SRv6 scheme, the location of the MN is updated according to the mobility procedure defined in [27]. In this procedure, in addition to the new mapping update process of the MN, the LISP control plane additionally sends the Map-Notify message to the previous xTR of the MN, thereby providing rerouting for the traffic destined to the MN. In the SRv6-based environment, as the ingress node inserts the entire service chain path, map discovery is performed with only one LISP message exchange; therefore, additional costs for the service chain path configuration are not required at the other xTRs. The location update cost and map discovery cost of the LISP-SRv6 are calculated as follows:

$$C_{LISP-SRv6}^L = E_{\omega_c} (P_{Map-Req} + P_{Map-Rep} + P_{Map-Noti}) \times \alpha h_{xTR-CP}, \quad (8)$$

$$C_{LISP-SRv6}^C = E_{\omega_c} (P_{Map-Req} + P_{Map-Rep}) \alpha h_{xTR-CP}. \quad (9)$$

In the proposed idea, the mobility scheme defined in [27] is also applied for packets destined for the moved MN, similar to the LISP-SRv6 approach. However, the difference is that an additional map discovery process is not required, as the previously used chain information in the ‘‘Mapped SR-list’’ field of the user mapping system is extracted and delivered with the Map-Reply message, which is the response of the location update. In addition, owing to the SRv6-based path configuration, additional chain reconfiguration costs are not required in all xTRs within the path other than the ingress node. The integrated location update and map discovery costs of the proposed idea are calculated as follows:

$$C_{Proposal}^{L+C} = E_{\omega_c} (P_{Map-Req} + P_{Map-Rep} + P_{Map-Noti}) \times \alpha h_{xTR-CP}. \quad (10)$$

B. HANDOVER DELAY ANALYSIS

The handover delay is defined as the sum of the transmission delay of signaling messages from when a new xTR detects the attachment of the user to the time when the end-to-end path is completely configured, which includes the service chain path. In a real environment, when performing the reconfiguration

of the service chain path, a processing cost for processing optimization algorithms and generating service chain encapsulation is additionally considered. However, this study does not consider the processing cost because the calculation of the processing cost is a significantly complex problem. Most of the related papers that solved these problems depended on a different algorithm or real implementations, which are not covered in this paper. The propagation delay $t_{X,Y}(s)$ of a packet size s between nodes X and Y are defined as follows [30]:

$$t_{X,Y}(s) = \frac{1-q}{1+q} \left(\frac{s}{B_{wl}} + L_{wl} \right) + (d_{X,Y} - 1) \left(\frac{s}{B_w} + L_w \right), \quad (11)$$

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 of the network transmission delay D , the handover delay of MN in each mobility method was calculated, based on procedures of the location update, and map discovery process for each scenario. Based on equations (4)-(10), the handover delay of each approach is formulated as follows:

$$D_{LISP-SFC} = t_{MN,xTR} (P_{RA} + P_{RS}) + t_{xTR,CP} (2P_{Flowmod}) + (N_{chain-xTR} + 1) \{t_{xTR,CP} (P_{Map-Req} + P_{Map-Res})\}, \quad (12)$$

$$D_{SFC-IPMM} = t_{MN,xTR} (P_{RA} + P_{RS}) + (t_{xTR,xTR} + t_{xTR,MN}) (P_{BU} + P_{BA}) + (N_{chain-xTR} + 1) (2t_{xTR,CP} P_{flowmod}), \quad (13)$$

$$D_{LISP-SRv6} = t_{MN,xTR} (P_{RA} + P_{RS}) + t_{xTR,CP} (P_{Map-Req} + P_{Map-Res} + P_{Map-Noti}) + t_{xTR,CP} (P_{Map-Req} + P_{Map-Res}), \quad (14)$$

$$D_{proposal} = t_{MN,xTR} (P_{RA} + P_{RS}) + t_{xTR,CP} (P_{Map-Req} + P_{Map-Res} + P_{Map-Noti}). \quad (15)$$

C. ANALYSIS RESULTS

Fig. 7 shows the results of the operating costs derived through the mathematical analysis model that is described above. Fig. 7(a) is a graph showing the operating costs while increasing the number of xTRs in the service chain path from 1 to 10. This implies that the number of service functions that belong to the service chain will also increase accordingly. When compared to the LISP-SFC and SFC-IPMM schemes, which require reconfiguration of the service chain path by exchanging messages with the control plane at all xTRs in the path, it was observed that the SRv6-based approaches have lower operating costs. In particular, in the case of SFC-IPMM,

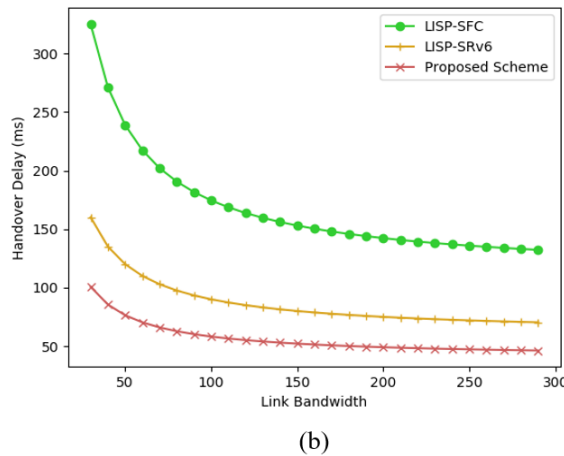
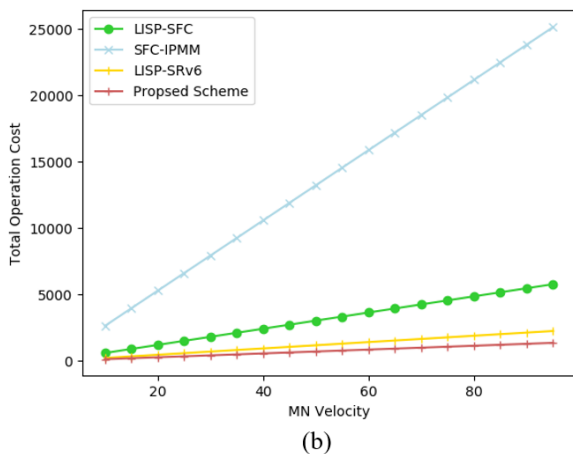
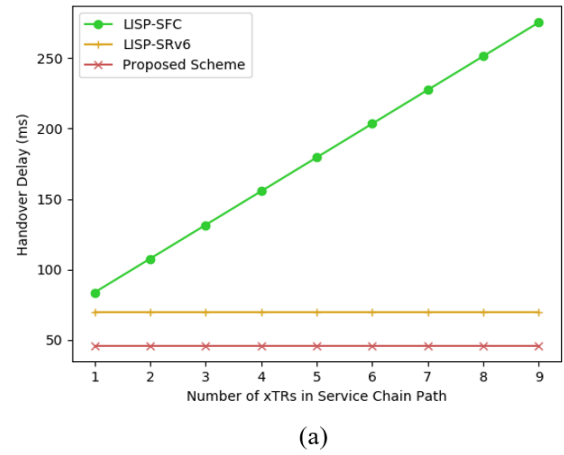
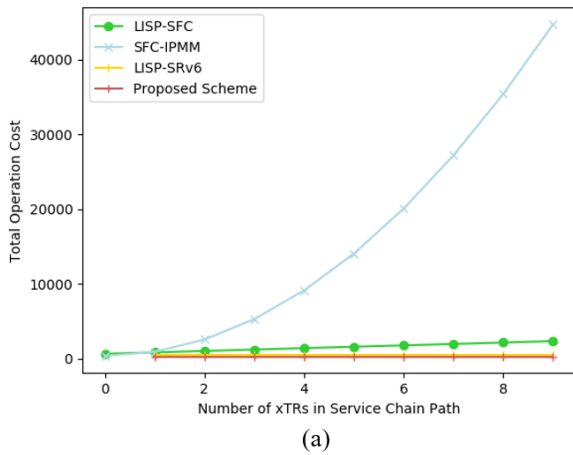


FIGURE 7. Operation costs corresponding to (a) number of ingress/egress tunneling routers (xTRs) in the service chain path and (b) different velocities of the mobile node.

the result shows that the operation cost increases exponentially corresponding to the number of xTRs because it uses an OpenFlow message that has a larger packet size. In both the SRv6-based approaches, the length of the chain and the number of xTRs do not affect the operation cost, as the other nodes do not require any additional signaling besides inserting the segment list at the ingress node. Among the two approaches, the proposed scheme achieves a lower operating cost when compared to the LISP-SRv6, which proves that integrating the location update and map discovery processes can efficiently perform network management. Fig. 7(b) shows the results of operating costs when the velocity of the MN increases from 0 to 100 m/s. When the velocity of the MN increases, handovers occur frequently, thereby increasing the cost for mobility management operations in the network and affecting the overall network performance. According to the results, the SRv6-based proposals showed relatively lower operating costs when compared to the other two proposals, and the proposed scheme has the lowest operating cost. Even at a higher MN speed, the increase in cost is less than that in other proposals; therefore, this shows that the scalability for service chain management can be ensured by efficient chain recovery and reconfiguration even with frequent handovers.

FIGURE 8. Handover delay corresponding to (a) number of xTRs in the service chain path and (b) different wired link bandwidth.

Fig. 8 shows the results of the handover delay analysis. In both results, the delay in the SFC-IPMM scheme is shown to be significantly higher than in the other schemes; therefore, its results are not completely shown in the graph. Fig. 8(a) shows the handover delay as the number of xTRs in the service chain path increases, and the SRv6 based proposals show a constant value regardless of the increase in chain length, similar to the previous results pertaining to the operation cost. The proposed scheme shows the lowest delay when compared to the other approaches, and this is because the previous service chain path states are maintained in the LISP mapping system and are recovered during the location update process, thereby omitting the additional procedures required to request for the service chain. Although not considered in this analysis, the handover delay in the LISP-SRv6 scheme is expected to increase further if the processing delay is caused by reconfiguring the initialized chain states after movement rather than a constant value depending on the length and complexity of the chain. Conversely, the proposed scheme maintains the state of the service path that was created at the previous ingress node, and then independently performs the processing for optimization and reconfiguration. Therefore, this scheme will have a constant delay value even

when considering processing cost. Fig. 8(b) shows the handover delay corresponding to an increase in the bandwidth of the wired link. It can be observed that the handover delays of all approaches decrease in all proposals according to the wired link performance. Moreover, according to the results, the proposed idea shows a delay time of less than 50 ms in a bandwidth of up to 300 Mbps. It is shown that a low handover delay enables fast recovery of end-to-end communication for the mobile user, enabling a better user experience in end-to-end communication using the proposed scheme.

V. CONCLUSION

In this paper, we proposed a LISP-based integrated control plane framework that closely cooperates with SFC management and mobility management in a distributed cloud environment. From the previous works that defined additional LISP mapping systems for services running across multiple clouds, we defined another mapping system for the service chain that is updated by the service manager, which is responsible for creating the service chain. With the sequential look-up process of mapping systems, the abstracted service chain defined only the set of SIDs that can be translated to the physical chain path. The service chain policy in the control plane for SFC creation and management is updated to the LISP control plane by additionally defining the mapping table for the service chain along with the ID/location mapping system for the distributed service proposed in the previous study. By doing so, the complexity of the path configuration through the physical location of the logical chain path is reduced. In addition, by defining an additional field in the user mapping system to bind the corresponding chain path for the traffic of the current user, the new ingress node can recover the existing chain state information quickly without requesting the service chain path again. According to the performance analysis, it was shown that integration with the SRv6-based path configuration and forwarding can reduce operation costs and handover delay efficiently by configuring the entire chain path at the new ingress node of the user traffic. In particular, based on the results of operational costs with increasing velocity of the MN, the effect of reducing the delay time during a handover according to the minimized service search process is expected to contribute to applications such as vehicular networks, where MNs move fast between edges, by providing fast reconnection of end-to-end communication with the service chain. The separation of the service chain path optimization process from the recovery of the existing path is expected to minimize the impact on the optimization process, which contributes to performance guarantee when the architecture becomes complex with the increase in the size of the chain and the number of services and distributed clouds. When services are distributed in multiple clouds and there are multiple locators for the same SID, the number of paths according to the combination of service location constituting the service chain is expected to increase significantly. Determining an appropriate service chain path within a

massive number of candidates requires a comprehensive consideration of various conditions such as resource conditions and network traffic load of each distributed service, rather than a single condition such as the minimum path. Therefore, it is necessary to formulate the calculation of processing costs in the control plane to calculate the complexity for future work.

REFERENCES

- [1] J. Halpern and C. Pignataro, *Service Function Chaining (SFC) Architecture*, document RFC 7665, 2015.
- [2] K. Kaur, V. Mangat, and K. Kumar, "A comprehensive survey of service function chain provisioning approaches in SDN and NFV architecture," *Comput. Sci. Rev.*, vol. 38, Nov. 2020, Art. no. 100298.
- [3] K. Yang, H. Zhang, and P. Hong, "Energy-aware service function placement for service function chaining in data centers," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.
- [4] D. Bhamare, R. Jain, M. Samaka, and A. Erbad, "A survey on service function chaining," *J. Netw. Comput. Appl.*, vol. 75, pp. 138–155, Nov. 2016.
- [5] G. Sun, Y. Li, H. Yu, A. V. Vasilakos, X. Du, and M. Guizani, "Energy-efficient and traffic-aware service function chaining orchestration in multi-domain networks," *Future Gener. Comput. Syst.*, vol. 91, pp. 347–360, Feb. 2019.
- [6] D. Dolson, S. Homma, D. Lopez, and M. Boucadair, *Hierarchical Service Function Chaining (hSFC)*, document RFC 8459, Sep. 2018.
- [7] Z. Zhou, Q. Wu, and X. Chen, "Online orchestration of cross-edge service function chaining for cost-efficient edge computing," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 8, pp. 1866–1880, Aug. 2019.
- [8] A.-V. Vu and Y. Kim, "An implementation of hierarchical service function chaining using OpenDaylight platform," in *Proc. IEEE NetSoft Conf. Workshops (NetSoft)*, Seoul, South Korea, Jun. 2016, pp. 411–416.
- [9] G. Sun, Y. Li, D. Liao, and V. Chang, "Service function chain orchestration across multiple domains: A full mesh aggregation approach," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 3, pp. 1175–1191, Sep. 2018.
- [10] D. Dietrich, A. Abujoda, A. Rizk, and P. Papadimitriou, "Multi-provider service chain embedding with nestor," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 1, pp. 91–105, Mar. 2017.
- [11] C. Zhang, X. Wang, A. Dong, Y. Zhao, F. Li, and M. Huang, "The intelligent multi-domain service function chain deployment: Architecture, challenges and solutions," *Int. J. Commun. Syst.*, vol. 34, no. 1, p. e4665, Oct. 2020.
- [12] Y. Liu, H. Lu, X. Li, D. Zhao, W. Wu, and G. Lu, "A novel approach for service function chain dynamic orchestration in edge clouds," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2231–2235, Oct. 2020.
- [13] C. Li, A. R. S. Hu, and Z. Li, *A Framework for Constructing Service Function Chaining Systems Based on Segment Routing*, document IETF Internet-Draft, draft-li-spring-sr-sfc-control-plane-framework-03 (on-going process), Oct. 2020.
- [14] 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.
- [15] 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. 26, no. 4, pp. 12–20, Aug. 2019.
- [16] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, *The Locator/ID Separation Protocol (LISP)*, document RFC 6830, Jan. 2013.
- [17] K. Sun and Y. Kim, "Enhanced LISP mapping system for optimizing service path in edge computing environment," *IEEE Access*, vol. 8, pp. 190559–190571, Oct. 2020.
- [18] C. Li, W. Cheng, M. Chen, D. Dhody, and R. Gandhi, *Path Segment for SRv6 (Segment Routing in IPv6)*, document IETF Internet-Draft, draft-ietf-spring-srv6-path-segment (On-going process), Nov. 2020.
- [19] P. Quinn, U. Elzur, and C. Pignataro, *Network Service Header (NSH)*, document RFC 8300, Jan. 2018.
- [20] A. Mayer, S. Salsano, P. L. Ventre, A. Abdelsalam, L. Chiaraviglio, and C. Filsfils, "An efficient linux kernel implementation of service function chaining for legacy VNFs based on IPv6 segment routing," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Paris, France, Jun. 2019, pp. 333–341.

- [21] C. Filsfils, S. Previdi, L. Ginsberg, B. Litkowski, and R. Shakir, *Segment Routing Architecture*, document RFC 8402, Jul. 2018.
- [22] P. L. Ventre, S. Salsano, M. Polverini, A. Cianfrani, A. Abdelsalam, C. Filsfils, P. Camarillo, and F. Clad, "Segment routing: A comprehensive survey of research activities, standardization efforts, and implementation results," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 182–221, 1st Quart., 2021.
- [23] A. Bashandy, C. Filsfils, B. Decraene, S. Litkowski, and R. Shakir, *Segment Routing With the MPLS Data Plane*, document RFC 8660, Dec. 2019.
- [24] M. Portoles, V. Ashtaputre, V. Moreno, F. Maino, and D. Farinacci, *LISP L2/L3 EID Mobility Using a Unified Control Plane*, document IETF Internet-Draft, draft-ietf-lisp-eid-mobility (On-going process), Jan. 2021.
- [25] V. Ermagan, P. Uinn, D. Lewis, F. Maino, and F. Coras, *LISP Control Plane Integration With NSH*, document IETF Internet-Draft, draft-ermagan-lisp-nsh (Expired), Oct. 2018.
- [26] A. Rodriguez-Natal, V. Ermagan, F. Maino, D. Dukes, P. Camarillo, and C. Filsfils, *LISP Control Plane for SRv6 Endpoint Mobility*, document IETF Internet-Draft, draft-rodrigueznatal-lisp-srv6 (On-going process), Jul. 2020.
- [27] M. Portoles, V. Ashtaputre, V. Moreno, F. Maino, and D. Farinacci, *LISP L2/L3 EID Mobility Using a Unified Control Plane*, document IETF Internet-Draft, draft-ietf-lisp-eid-mobility (On-going process), Jan. 2021.
- [28] F. Clad, X. Xu, C. Filsfils, D. Bernier, C. Li, B. Decraene, S. Ma, C. Yadlapalli, W. Henderickx, and S. Salsano, *Service Programming With Segment Routing*, document IETF Internet-Draft, draft-ietf-spring-sr-service-programming (On-going process), Sep. 2020.
- [29] E. Seo, V. Zalyubovskiy, and T.-M. Chung, "Mathematical modeling of the scalable LISP-deployed software-defined wireless network," *J. Commun.*, vol. 14, no. 8, pp. 727–738, 2019.
- [30] 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. degree/the Ph.D. degree with the Department of Electronic Engineering. His current research interests include 5G networking, network function virtualization (NFV), and mobility management.



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

• • •