OpenFlow-Based Mobility Management Scheme and Data Structure for the Mobility Service at Software Defined Networking

Pill-Won Park, Seong-Mun Kim, and Sung-Gi Min

Department of Computer and Radio Communication Engineering, Korea University, Seoul 136-713, Republic of Korea

Correspondence should be addressed to Sung-Gi Min; sgmin@korea.ac.kr

Received 18 November 2015; Revised 5 February 2016; Accepted 10 February 2016

Academic Editor: Luis Javier Garcia Villalba

Copyright © 2016 Pill-Won Park et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The network-based mobility management is adapted to the OpenFlow architecture for mobility service at Software Defined Networking (SDN), and data structure for mobility service is proposed. SDN is a newly proposed Internet architecture which decouples the data and control planes, and mobility management is one of the most important issues in SDN. In order to provide mobility management service by utilizing the mobility scheme proposed earlier in a new network environment, the existing mobility schemes need to be modified. Particularly, the characters of the network environment need to be considered when designing the function of the network entities and data structure. Our proposed mobility management scheme is focused on the centralized control mechanism of SDN. We referred to Proxy Mobile IPv6 (PMIPv6) and OpenFlow-based PMIPv6 with a centralized mobility management controller (OPMIPv6-C). It has a merged data structure for the mobility service on SDN controller and minimizes the number of switches which need flow table modification at handover. At the performance analysis chapter, we compare the signaling cost at the registration and handover phase, packet delivery cost, and handover latency between the proposed scheme and OPMIPv6-C.

1. Introduction

Software Defined Networking (SDN) is newly proposed network architecture for the programmable network [1]. SDN separates the control plane and the data plane by the role of the network functions. The data plane is responsible for the packet forwarding and the control plane performs other functions. The control functions which were located on the distributed devices are gathered to the centralized controller. Due to the separation, standard interface is needed for the interaction between the control plane and data plane. The OpenFlow is one of the standard interfaces between the control and data plane, and it is proposed by the Open Networking Foundation (ONF).

The OpenFlow architecture consists of the OpenFlow-enabled switch, secure channel, and OpenFlow protocol. The OpenFlow-enabled switch is the switch which satisfies the OpenFlow specification. The secure channel is the path for transmission of the OpenFlow messages between the controller and switches. The OpenFlow protocol is the format of OpenFlow messages for the control, notification, and packet transfer.

The mobility management scheme is able to be categorized into two types: the network-based and host-based mobility management or micro and macro mobility management protocol. The difference of the network-based and host-based mobility management is the subject of the mobility-related signaling. The network entities are the subject of mobility-related signaling in the network-based mobility management. The MN is subject of the mobility management signaling in the host-based management scheme. The difference of the micro and macro mobility management is the mobility management area for handover control. The macro mobility management protocol handles the interdomain mobility and micro mobility management protocol controls the intradomain mobility, respectively. PMIPv6 is one of the network-based micro mobility management protocols and it is standardized by IETF. The network entities, LMA
and MAG, are responsible for the mobility control signaling instead of the MN, and PMIPv6 performs the intradomain mobility management service.

Several mobility management schemes for mobility service in SDN have been suggested [2-5]. Reference [2] suggests a brief idea for mobility service in SDN, but it does not describe any detailed mobility management procedures or data structure. References [4, 5] apply PMIPv6 to SDN. They focused on the adaptation of Proxy Mobile IPv6 (PMIPv6) to OpenFlow architecture over SDN. They keep PMIPv6 architecture and relocate the mobility control functions into SDN controller. Reference [4] proposes locating the distributed network entities, local mobility anchor (LMA), and the Mobile Access Gateway (MAG), into SDN controller. Reference [5] is similar to [4]. However they still use PMIPv6 signaling and distributed data structure. Reference [3] suggests PMIPv6-based mobility management scheme on SDN. It gathers the mobility management function into SDN controller and uses the packet rewriting mechanism instead of the tunneling mechanism. However, due to the packet rewriting mechanism, the access switch cannot handle more than one MN.

In this paper, we proposed the OpenFlow-Based Mobility Management (OMM) scheme and an integrated data structure for mobility service in SDN. The proposed scheme integrates the mobility management functions into a single mobility management entity (MME) and organizes the new data structure which is based on the LMA and MAG data structure. It eliminates all entries related to mobility management control messages and data tunnels and integrates the duplicated entries in MAG and LMA. OMM has the Flow Matrix which records the flow paths for mobile node (MN). And it adapts the flow path setup mechanism of [5, 6].

The rest of this paper is organized as follows. Section 2 summarizes the related works. In Section 3, the architecture of the proposed scheme is introduced in terms of the data structure and mobility management entity. In Section 4, the signaling flow of the proposed scheme is introduced in terms of the registration phase and handover phase. A performance evaluation is performed in Section 5. Finally, Section 6 concludes this paper.

2. Related Works

Mobility Support in Software Defined Networking [2] suggests the basic structure for the mobility management in SDN. But it proposes only the idea and does not suggest any specific method.

Mobility Management Framework in Software Defined Networks [4] and an adaptation of Proxy Mobile IPv6 to OpenFlow architecture over Software Defined Networking [5] suggest adaptation of PMIPv6 into SDN.

Reference [4] is focused on location of the mobility management entities for mobility service in SDN. It moves the LMA and MAGs to SDN controller as the applications and all mobility management follows PMIPv6. It keeps LMA and MAG functions and uses tunneling to forward data packets. Reference [5] suggests two mobility management schemes: OpenFlow-based PMIPv6 (OPMIPv6) and OPMIPv6 with a centralized mobility management controller (OPMIPv6-C). OPMIPv6 has LMA control function on SDN controller and MAGs are located at the access switches. OPMIPv6-C suggests centralized MAG at SDN controller as well as the LMA control function, but the two modules are not unified. Both schemes use the same mobility control mechanism in PMIPv6, but these two schemes eliminate packet tunneling. They use two flow entries on the OpenFlow-enabled switches on the path between the access switch and the gateway switch. One entry handles the downstream traffic and the other upstream traffic. There is no packet modification at the access and gateway switches.

A solution for IP Mobility Support in Software Defined Networks [3] also proposes the centralized mobility management function at the SDN controller. The scheme does not use tunneling to forward data packets from a CN to the MN. The GW of the SDN domain rewrites the destination address of packet to IP address of the AS, and the AS reverts its IP address to the original destination address when it receives the forwarded packet. Due to the rewriting policy, the AS cannot handle more than one MN. The AS cannot identify which destination address should be used to replace the destination address of the incoming packet as the incoming packet does not have any such indicator in the packet. In addition, the scheme uses the triangular packet forwarding but the ingress filtering problem is not considered at all.

3. An OpenFlow-Based Mobility Management

The OpenFlow-Based Mobility Management (OMM) refers to PMIPv6 to identify the requirements of the mobility management. In this paper, we assumed two things. First, the MN attachment is notified by layer-2 notification which contains the MN’s identifier. Second, the IPv6 address of MN is created by using the home network prefix (HNP). These are the same assumptions in PMIPv6.

3.1. Architecture. The proposed scheme consists of an MME on the centralized SDN controller, OpenFlow-enabled switches, and an Authentication, Authorization, and Accounting (AAA) server. SDN controller is the basement of control applications and it supports them using internal functions, such as the topology manager, database, and link discovery [7]. The OpenFlow-enabled switches provide packet forwarding within SDN domain. The AAA server authenticates MNs for the mobility service. If an MN has the right of the mobility service, the AAA provides the MN’s profile to the MME. The solid lines and dotted lines in Figure 1 indicate the control flow and the data flow.

3.1.1. OpenFlow-Enabled Switch. An OpenFlow-enabled switch is a switch which satisfies the OpenFlow specification [8]. According to their role in OMM, the switches are categorized into access switches (ASs), intermediate switches (ISs), and gateway switches (GWs). An AS provides the network access service to MNs. A GW connects the access network with the Internet. ISs are located on a flow path between the AS and the GW. A Crossing Switch (CS) is one of the ISs, and it is located at the branch point of old and new flow paths of...
an MN when a handover occurs. The CS is used to reduce the number of flow table updates on switches affected by the handover.

3.1.2. Mobility Management Entity. An MME is a control application for the mobility management and it runs over SDN controller. The MME detects the attachment of an MN and provides the mobility service for MN. After authentication of MN, it allocates the HNP for the MN and establishes downstream and upstream flow paths for the MN. It also performs the handover procedure when the MN moves into another AS.

3.2. Data Structures in the MME. The MME maintains the attached MN’s information at the binding cache. To perform the HNP allocation and flow path establishment, the MME has two additional data structures: a GW-HNP mapping table and a Flow Matrix. These additional data structures are created before the MME provides the mobility service.

3.2.1. Binding Cache. The Binding Cache is a list of Binding Cache Entries (BCEs) for each MN. A BCE is created when a new MN is attached to SDN domain and is updated when the network state or the attachment point of MN is changed. Each BCE contains MN’s information and Flow Matrix information. The format and the acquisition process of each field follow these in PMIPv6 specification. The MN information includes an MN’s identifier (MN-ID), a link-layer identifier (LL-ID), a list of MN’s home network prefixes (MN-HNPs), and an Access Technology Type (ATT), as shown in Figure 2(a). The Flow Matrix information includes the gateway index (GW_i), the AS index (AS_j), and an interface identifier of the AS to which the MN is attached. The GW_i and AS_j signify two end points of the flow path for MN, like the LMA and MAG in PMIPv6. They are used to access the specific cell at the Flow Matrix, as follows C(GW_i, AS_j).

3.2.2. GW-HNP Mapping Table. When a new MN attaches to SDN domain, the MME must allocate an HNP for the MN. If the MME may obtain the HNP from the MN’s profile, the MME must know which GW handles the HNP for the MN. If not, the MME randomly assigns a corresponding GW for the MN and allocates one of HNPs handled by the GW. The MME can know which GW manages the assigned HNP based on the GW-HNP mapping table, as depicted in Figure 2(b).

3.2.3. Flow Matrix. The Flow Matrix has the flow-related information for providing mobility service in SDN domain. It contains three pieces of information: a list of flow paths, the pairs of a previous AS (pAS) and CS, and a list of HNPs, as shown in Figure 2(c). An element in the list of

![Figure 1: System architecture.](image)

![Figure 2: The mobility-related information tables in the mobility management entity.](image)
flow paths is described as a pair of an upstream flow path and a downstream flow path, which is a Switch-ID (port index for upstream, port index for downstream). For example, GW₁(2,1) means that the upstream and downstream are forwarded through port 2 and port 1 of GW₁, respectively. The pairs of pAS and CS P(pAS, CS) are composed of each pAS and the CS related to it. The MN is able to come from every other pAS to the new AS (nAS) in SDN domain, and the CS needs to calculate per each pAS. Thus, the Flow Matrix keeps the P(pAS, CS) per each pAS. Only the flow table of switches between the nAS and the CS is updated, not every switch on the whole flow path. A list of HNPs consists of the allocated HNPs for MN.

After the GW is selected, the MME has to find the flow paths between the AS to which the MN is attached and the selected GW. Each MN needs two flow paths: one for upstream traffic and the other for downstream traffic. The two flow paths use the same switches, but their in/out ports are reversed. As the wired network topology is pretty stable, these flow paths can be calculated proactively. SDN topology information for the flow path calculation can be taken dynamically using the Link-Layer Discovery Protocol [9]. Based on the topology of each AS, the MME can calculate the best flow path between each AS and GW. The precalculated flow paths are recorded in the Flow Matrix. Each cell of the Flow Matrix is accessed by using GW and AS.

When an MN is moving into a new AS, the information of the new flow path can be found with the new AS index in the Flow Matrix. Then, the flow table entries of switches on the new flow paths must be updated. However, the switches included in both the old and new flow paths except the CS should be excluded for the flow table update. Therefore, each cell of the Flow Matrix includes CS information for all other flow paths. Only flow table entries on switches between the CS and the AS on the new flow paths need to be updated.

A cell in the Flow Matrix also includes a list of HNPs. These HNPs share the flow paths between the AS and the GW. When the flow paths in the cell are changed due to the underlying network topology change, all flow table entries of switches affected by the topology change must be updated. The MME uses the HNP list to update these flow table entries.

Figure 3 shows an example of the Flow Matrix. C(GW₁, AS₁) indicates the flow paths between GW₁ and AS₁, pairs of pAS and CS, and the HNP list for the flow path between GW₁ and AS₁. The downstream and upstream flow path based on the example topology are listed, as follows: GW₁(1,2), IS₁(1,2), IS₁(1,2), AS₁(1,null); null is replaced with the interface identifier of the Flow Matrix information of the corresponding BCE.

If the MN moves from the AS₁ to AS₂, the MME will look up a corresponding cell in Flow Matrix for GW₁ and AS₂. The MME can find C(GW₁, AS₂) and update the flow table of switch on the new flow paths. Before updating, the MME decides which switches to update based on P(AS₂, GW₁) of C(GW₁, AS₂). In this case, the CS is GW₁. Thus, the MME updates the flow tables of switches between GW₁ and AS₂.

The Flow Matrix provides several advantages. Firstly, it prevents the duplicate computation about flow paths. The MME already maintains the precalculated flow paths in the Flow Matrix. Thus, when a new MN attaches to one of ASs, the MME does not calculate flow paths and just uses the corresponding cell of the Flow Matrix. Secondly, the number of the flow paths managed by the MME can be reduced. A flow path is configured per MN in SDN. However, the flow path can be shared in the proposed scheme if MNS belong to the same cell of Flow Matrix. Thirdly, the MME can easily handle the topology change. Flow table entries affected by the underlying network topology change must be reestablished. The MME can detect the flow paths affected by the topology change by recalculating the Flow Matrix. If any flow paths are updated after recalculating, the MME must update the corresponding flow table entries on switches on the affected flow paths with the HNP list. For each HNP in the HNP list, two flow table entries on each switch on the affected flow paths are modified by the MME.

### 4. Procedures of OpenFlow-Based Mobility Management

In this section, the registration procedure and intradomain handover procedure are described in detail.

#### 4.1. Before Mobility Service

The MME creates the Flow Matrix and the GW-HNP mapping table. If SDN controller supports the flow path setup, the MME requests the flow path to SDN controller for an MN between each GW and AS. If not, the MME requests the network topology to SDN controller and computes the flow paths between each GW and AS. The MME records it on the Flow Matrix.

#### 4.2. Registration to MME

The registration procedure is shown in Figure 4 and is as follows.

- **4.2.1. Mobile Node Attachment.** When an MN is attached to SDN domain, the AS is notified by the L2 notification and forwards it to the MME. The L2 notification contains the MN-ID, LL-ID, and ATT.

- **4.2.2. Authentication.** The MME sends the MN-ID to the AAA server for authority of the mobility service. If the MN has the right of the mobility service, the AAA authorizes
and replies the profile of the attached MN. The MN’s profile includes the GW information which will be allocated to MN and the HNP assigned to the connected interface. If MN’s profile does not include HNP, the MME allocates an HNP for the MN based on the LL-ID and the GW-HNP mapping table.

4.2.3. Select the Flow Path in the Flow Matrix. At this time, the MME recognizes the indexes of the GW (GW\(_i\)) and AS (AS\(_j\)) for the MN. The MME looks up \(C(GW_i, AS_j)\) in the Flow Matrix. The MME chooses the cell in the Flow Matrix and gets the flow path for MN. Finally, the MME adds the assigned HNP for the MN to the HNP list of the cells.

4.2.4. Send the FMOD and Update the Flow Table in Switches. After selecting the flow path, the MME can know the ISs between the AS and the GW for the MN. The MME sends a Flow Modification (FMOD) message to the switches on the path as soon as possible. Then, the switches update their flow table for the MN’s upstream and downstream flows.

4.2.5. Router Advertisement. The MME sends a unicast Routing Advertisement (RA) message to the MN via the AS using POUT. The RA conveys the HNP and the other configuration parameters. The AS forwards the RA to the MN via the port to which the MN is attached. After receiving the RA, the MN configures an IP address.

4.3. Intradomain Handover. The handover procedure is shown in Figure 5 and is as follows.

4.3.1. Mobile Node Detachment and Attachment. When the MN attaches to a nAS, the L2 notification is sent from nAS to the MME.

4.3.2. Select the New Flow Path in the Flow Matrix. The MME looks up the matching BCE based on the L2 notification. If the BCE is found, a handover is performed. The MME looks up the cell \(C(GW_i, AS_n)\) in the Flow Matrix by using the indexes of the GW (GW\(_i\)) and nAS (AS\(_n\)) in the BCE and checks the \(P(AS_n, CS)\) field to find the CS. Thus, the MME can know the switch list between the CS and nAS. It adds the HNP for the MN to the HNP list of \(C(GW_i, AS_n)\) and removes the HNP from that of \(C(GW_i, AS_j)\).

4.3.3. Send the FMOD to Switches and Update the Flow Table. The MME sends the FMOD to the switches on the flow path between the nAS and CS in order to update the flow table entries. If not, the downstream packets will be forwarded to
5. Performance Evaluation

This section describes the topology and the messages which are used by OPMIPv6-C and OMM protocol for evaluation. The topology consists of a single GW and several switches to represent a SDN administrative domain. We developed the analytical cost model to evaluate the performance of OMM and the OPMIPv6-C based on [10]. The following notations are used to develop the cost models [5, 10–13].

5.1. Network Model. We exploited the similar network model and the concept of [5] for evaluations. The topology shown in Figure 6 is used to represent provisioning entities in OMM and OPMIPv6-C. We assumed that the controller is connected by the wired line directly to all switches in the domain. That means that the hop count between the controller and the switches is one. The CN is placed on the outside of a given administrative domain, and the movement of MN is limited within the domain.

The \( h_{a:b} \) means the average hop count between \( a \) and \( b \):

(i) \( h_{C:S} \): hop count between the controller and the switches.

(ii) \( h_{CN:G} \): hop count between the CN and the GW.

(iii) \( h_{G:AS} \): hop count between the GW and the access switch.

(iv) \( h_{CS:AS} \): hop count between the CS and the AS.

(v) \( h_{AS:M} \): hop count between the AS and the MN.

5.2. Mobility Model. Both OPMIPv6 and OMM are local mobility management protocols. Thus, we only consider intradomain mobility based on that presented in [5, 14]. In order to evaluate the mobility model, we assume that all access points (AP) in the given domain have a circular coverage of the same size \( S \). The coverage of the AP is calculated as \( S = \pi R^2 \), where \( R \) is the radius of the AP coverage. \( \mu_n \) is the AP crossing rate of the MN, and it is expressed as \( 2\nu/\sqrt{\pi S} \), where \( \nu \) is the average velocity of the MN [15]. Thus, the signaling cost applied to the mobility model is expressed as \( \mu_n C_S^{(1)} \).

5.3. Messages Size of the OPMIPv6-C and OMM. OpenFlow messages are carried on Transmission Control Protocol (TCP). The sizes of the messages that are used in OMM and OPMIPv6-C are shown in Table 1. We will follow the assumed attachment method of MN that is suggested in [5]. For comparison, we assumed that the L2 notification is substituted by the Extended Switch Configuration (exSC) message in extended OpenFlow message format, which sends the MN attachment event through AS to SDN controller. However, OMM uses RA instead of the Extended Port Status (exPS) message. The message sizes (bytes) used in OMM and OPMIPv6-C and the OpenFlow protocol are shown in Table 1.

Their size includes an IPv6 header and a TCP header, and we consider the TCP Acknowledgment (TCPA) generated by each OpenFlow message.

5.4. Cost Modeling. We developed the cost model which evaluates the performance of the OPMIP-C and OMM based on those presented in [10]. The following notation was used to develop the cost model [11–13, 15, 16]. \( \alpha \) and \( \beta \) each mean the weighting factor of the wired and wireless link in order to emphasize the link characteristic. \( \lambda \) indicates the average session arrival rate at the MN, and \( N(p) \) is the number of packets of a session. An MN moves from \( AS_i \) to \( AS_j \) while the MN communicates with CN. The signaling cost is calculated based on this scenario.

5.4.1. OPMIPv6-C. The signaling cost of OPMIPv6-C \( C_S^{(OPMIPv6-C)} \) consists of the binding update cost \( C_{BU}^{(OPMIPv6-C)} \) and the flow modification cost \( C_{FMOD}^{(OPMIPv6-C)} \).

\( C_{BU}^{(OPMIPv6-C)} \) is expressed as follows:

\[
C_{BU}^{(OPMIPv6-C)} = h_{C:S}(L_{exSC} + L_{exPS} + 2L_{TCPA}).
\]  

And the flow modification cost per each switch is expressed as follows:

\[
C_{FMOD} = h_{C:S}(L_{FMOD} + L_{TCPA}).
\]  

The MME sends FMOD to switches two times for the ingress and egress traffic. \( h_{G:AS} + 1 \) shows the number of

![Figure 6: The network topology.](image-url)
The parameter for handover latency is referenced from Table 2. The parameter for handover latency includes the parameters for the L2 layer handover latency $T_{L2}$, the random amount of delay before sending RS $T_{RD}$, the arrival delay of the RS message $T_{RS}$, the delay of the binding update $T_{BU}$, the arrival delay of the first packet sent $T_p^+$, the bandwidth of the wired and wireless link $B_{WDL}$, and the propagation time of the wired and wireless link $PT_{WDL}$. The propagation time of the wired and wireless link $PT_{WDL}$ can be selected from between 0 and MAX_RTR_SOLICITATION_DELAY (1000 ms) [17]. $T_{RS}$ is then calculated as

$$T_{RS} = \frac{L_{RS}}{B_{WL}} + PT_{WL}. \quad (9)$$

The RS message is then sent over the wireless link between the MN and the access switch. In this case, it is assumed that there are no wired or wireless link failures, and we also do not take any authentication procedures into consideration.

5.5.1. OPMIPv6-C Handover Latency. OPMIPv6-C does not use PMIPv6 signaling. It only uses the extended OpenFlow signaling through a secure channel over TCP. Thus, $T_{S}^{(OPMIPv6-C)}$ is expressed as

$$T_{S}^{(OPMIPv6-C)} = h_{G-S} \left( \frac{L_{exSC} + L_{TCPA}}{B_{WD}} \right) + 2h_{G-A} PT_{WD} + h_{G-S} \left( \frac{L_{exPS} + L_{TCPA}}{B_{WD}} \right), \quad (10)$$

5.5.2. OMM Handover Latency. OMM is almost similar to OPMIPv6-C. Thus, $T_{S}^{(OMM)}$ is expressed as

$$T_{S}^{(OMM)} = h_{C-S} \left( \frac{L_{exSC} + L_{TCPA}}{B_{WD}} \right) + h_{C-S} \left( \frac{L_{TCPA}}{B_{WD}} \right) + 2h_{G-A} PT_{WD}. \quad (11)$$

$T_{p}^{(OPMIPv6-C)}$ is similar to $T_{p}^{(OPMIPv6-C)}$ because they use the same packet delivery mechanism.

5.6. Cost Analysis Results. In this chapter, we explained the signaling cost analysis results for OPMIPv6-C and OMM. We set the default values of the system parameters for the cost analysis as $h_{C-S} = 1, h_{CN-G} = 5, h_{G-AS} = 5, h_{CS-AS} = [0, 5], h_{AS-MN} = 1, \lambda_{\alpha} = [0, 1], L_{\alpha} = 64$, and $N(p) = 20$. The signaling cost versus $\alpha$ is shown in Figure 7. We set the parameters $R$ as 400 m and set the parameters of $u$ to 0 m/s and 100 m/s. The signaling cost is directly proportional to $u$, and the signaling cost of OMM is almost identical to that of OPMIPv6-C. The reason of difference is only in the packet header size of the RA and exPS. If OMM used exPS instead of RA, they would have had the same signaling cost.

---

**Table 2: The parameter for handover latency.**

| Notification | Comment |
|--------------|---------|
| $T_{L2}$     | The L2 layer handover latency |
| $T_{RD}$     | The random amount of delay before sending RS |
| $T_{RS}$     | The arrival delay of the RS message |
| $T_{BU}$     | The delay of the binding update |
| $T_p^+$      | The arrival delay of the first packet sent |
| $B_{WDL}$    | The bandwidth of the wired and wireless link |
| $PT_{WDL}$   | The propagation time of the wired and wireless link |

**Switches between the GW and AS.** The signaling cost is expressed as follows:

$$C^{(OPMIPv6-C)}_{S} = C^{(OPMIPv6-C)}_{BU} + 2(h_{G-AS} + 1)C_{FMOD}. \quad (3)$$

The handover process of OPMIPv6-C needs the same procedure as the registration phase. Thus, the handover signaling cost of OMM is different from that of OPMIPv6-C. The reason of difference is only in the packet header size of the RA and exPS. If OMM used exPS instead of RA, they would have had the same signaling cost.

**5.4.2. OpenFlow-Based Mobility Management.** The basic mobility signaling of OMM, $C_{OMM}^{(OMM)}$, is similar to that of OPMIPv6-C. However, OMM uses the RA instead of the Extended Switch Configuration (exSC) message and the signaling cost of OMM at handover phase is different from that of OPMIPv6-C by the CS. The RA is forwarded to the AS, included in POUT. $C_{OMM}^{(OMM)}$ is expressed as follows:

$$C_{BU}^{(OMM)} = h_{C-G} \left( L_{exPS} + L_{POUT_{AS}} + 2L_{TCPA} \right). \quad (5)$$

The flow modification cost per switch is the same as (2). The signaling cost of OMM $C_{OMM}^{(OMM)}$ is expressed as

$$C_{OMM}^{(OMM)} = C_{BU}^{(OMM)} + 2(h_{G-AS} + 1)C_{FMOD}. \quad (6)$$

At handover, OMM controller only sends FMODP to switches between nAS and CS. Thus, $C_{OMM}^{(OMM)}$ is expressed as follows:

$$C_{OMM}^{(OMM)} = C_{BU}^{(OMM)} + 2(h_{CS-AS} + 1)C_{FMOD}. \quad (7)$$

$C_{OMM}^{(OMM)}$ is similar to $C_{PD}^{(OPMIPv6-C)}$ because they use the same packet delivery mechanism.

5.5. Basic Handover Process. We exploit the definition for the handover latency $S_{HO}$ in [16]. The parameter for handover in Table 2 is referenced from [5, 16]. The handover latency for the handover process $S_{HO}$ is able to be expressed as

$$S_{HO} = T_{L2} + T_{RD} + T_{RS} + T_{BU}^{+} + T_{p}^{+}. \quad (8)$$

The value of $T_{RD}$ can be selected from between 0 and MAX_RTR_SOLICITATION_DELAY (1000 ms) [17]. $T_{RS}$ is then calculated as

$$T_{RS} = \frac{L_{RS}}{B_{WL}} + PT_{WL}. \quad (9)$$

The RS message is then sent over the wireless link between the MN and the access switch. In this case, it is assumed that there are no wired or wireless link failures, and we also do not take any authentication procedures into consideration.
Figure 7: Signaling cost versus $v$ ($R = 400$ m).

Figure 8: Signaling cost versus $R$ ($v = 40$ m/s).

Figure 9: Signaling cost at handover versus $h_{CS-AS}$.

Figure 10: Packet delivery cost versus $\lambda_s$.

Figure 11: Total cost versus $S_{\phi}$ ($\lambda_s = 1$).

Figure 12: Handover latency versus $B_{WD}$. The packet transfer mechanism of OPMIPv6-C and OMM is similar, but the different thing is the control message length.

5.7. Handover Latency Analysis. The system parameter values for the handover latency were $T_{L2} = 45.35$ ms, $B_{WL} = 11$ Mbps, $B_{WD}/B_{WL} = 1000$ ms, $L_{DATA} = 1328$ bytes, $PT_{WL} = 2$, and $PT_{WD} = 0.5$ ms based on [18, 19].

6. Conclusion

We are proposing OMM, which is an adaptation of PMIPv6 concept to SDN architecture. In order to support mobility in SDN architecture, the mobility functions are unified from the separated PMIPv6 entities to MME. The proposed scheme only uses OpenFlow signaling and PMIPv6 concept. Also it is focused on the data structures for the mobility service, detailed procedure, and efficiency path setup mechanism.
The result of the performance evaluation indicates that the proposed scheme is efficient compared to OPMIPv6-C at the handover phase.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] Open Networking Foundation, Software-Defined Networking: The New Norm for Networks, Open Networking Foundation, 2012.
[2] D. Liu and H. Deng, "Mobility support in software defined networking-draft-liu-sdn-mobility-00," IETF Internet-Draft, 2014.
[3] Y. Wang and J. Bi, "A solution for IP mobility support in software defined networks," in Proceedings of the 23rd International Conference on Computer Communication and Networks (ICCCN '14), pp.1–8, IEEE, Shanghai, China, August 2014.
[4] K.-H. Lee, "Mobility management framework in software defined networks," International Journal of Software Engineering and Its Applications, vol. 8, no. 8, pp. 1–10, 2014.
[5] S. M. Kim, H. Y. Choi, P.-W. Park, S. G. Min, and Y. H. Han, "OpenFlow-based Proxy mobile IPv6 over software defined network (SDN)," in Proceedings of the IEEE 11th Consumer Communications and Networking Conference (CCCN '14), pp. 119–125, Las Vegas, Nev, USA, January 2014.
[6] M. Liebsch, S. Jeong, and Q. Wu, "Proxy mobile IPv6 (PMIPv6) localized routing problem statement," RFC 6279, 2011.
[7] K.-C. Wang, Floodlight User Documentation, 2012, http://www.projectfloodlight.org/floodlight/.
[8] Open Networking Foundation, OpenFlow Switch Specification Version 1.5.0, 2014.
[9] IEEE, "IEEE Standard for Local and metropolitan area networks—station and media access control connectivity discovery," IEEE 802.1AB-2009, 2009.
[10] J. Lee, E. Thierry, and T. Chung, "Cost analysis of IP mobility management protocols for consumer mobile devices," IEEE Transactions on Consumer Electronics, vol. 56, no. 2, pp. 1010–1017, 2010.
[11] J.-H. Lee, T.-M. Chung, and S. Gundavelli, "A comparative signaling cost analysis of hierarchical mobile ipv6 and Proxy Mobile ipv6," in Proceedings of the 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '08), pp. 1–6, IEEE, Cannes, France, September 2008.
[12] S.-C. Lo, G. Lee, W.-T. Chen, and J.-C. Liu, "Architecture for mobility and QoS support in all-IP wireless networks," IEEE Journal on Selected Areas in Communications, vol. 22, no. 4, pp. 691–705, 2004.
[13] Y.-H. Han and S.-H. Hwang, "Care-of address provisioning for efficient IPv6 mobility support," Computer Communications, vol. 29, no. 9, pp. 1422–1432, 2006.
[14] J.-H. Lee and T.-M. Chung, "How much do we gain by introducing route optimization in Proxy Mobile IPv6 networks?" Annales des Télécommunications—Annals of Telecommunications, vol. 65, no. 5–6, pp. 233–246, 2010.
[15] F. Baumann and I. Niemerggers, "An evaluation of location management procedures," in Proceedings of the IEEE International Conference on Universal Personal Communications (UPC '94), pp. 359–364, September-October 1994.
[16] J.-H. Lee, Y.-D. Kim, and D. W. Lee, "Enhanced handover process for Proxy Mobile IPv6," in Proceedings of the 4th International Conference on Multimedia and Ubiquitous Engineering (MUE '10), pp. 1–5, Cebu, The Philippines, August 2010.
[17] T. Narten, E. Nordmark, and W. Simpson, "Neighbor discovery for IP version 6 (IPv6)," IETF RFC 4861, 2007.
[18] S. Pack, J. Choi, T. Kwon, and Y. Choi, "Fast-handoff support in IEEE 802.11 wireless networks," IEEE Communications Surveys and Tutorials, vol. 9, no. 1, pp. 2–12, 2007.
[19] S. Pack, X. Shen, J. W. Mark, and J. Pan, "Adaptive route optimization in hierarchical mobile IPv6 networks," IEEE Transactions on Mobile Computing, vol. 6, no. 8, pp. 903–914, 2007.
