Data Structure and Switch Categorization for Mobility Management Service in Software Defined Networking

Pill-Won Park¹, Seong-Mun Kim¹, and Sung-Gi Min¹,

¹ Department of Computer and Radio Communication Engineering Korea University
Seoul, South Korea.
{pillwon79,sgmin}@korea.ac.kr

Abstract. In this paper, the Data Structure and Switch Categorization are suggested for the network-based mobility management which is adapted to the OpenFlow architecture at Software Defined Networking (SDN). SDN is newly internet architecture which decouples the data and control planes, and the mobility management is one of the most important issues in SDN. The several network based mobility schemes for SDN are proposed, but they lack consideration about SDN. Our proposed data structure and switch categorization are focused on the centralized mechanism of SDN. It is referenced Proxy Mobile IPv6 (PMIPv6) and OpenFlow-based PMIPv6 with a centralized mobility management controller (OPMIPv6-C). We proposed the merged data structure and switch categorization for the mobility service on the SDN, and minimizes the number of switches which need flow table modification at handover.

Keywords: Software Defined Networking (SDN), Mobility Management, Data Structure for mobility.

1 Introduction

Software Defined Networking (SDN) is newly network architecture for the programmable network. 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 was located on the distributed devices are gathered to the centralized controller. By the separation, standard interface is needed for the interaction between control plane and data plane. The OpenFlow is one of the standard interface between control and data plane, 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 switches which are satisfied the OpenFlow specification. The secure channel is path for transmission of the control messages from controller to switches. The OpenFlow protocol is format about OpenFlow message for sharing the control, notification, and packet transfer.
Several mobility management schemes for mobility service in SDN have been suggested. But they do not describe the detail mobility management procedures nor data structure.

We propose the data structure and switch categorization for the network-based mobility management service for SDN. The proposed data structure refers PMIPv6, and it is merged and lighten by removing the duplicate field. It is used on the integrated mobility management functions into a single mobility management entity (MME). It eliminates all entries related to mobility management control messages, data tunnels, and unifies duplicated entries in MAG and LMA.

2 Related Works

Mobility Support in Software Defined Networking [2] suggests the basic guideline for the mobility management service in SDN. But it only proposes the mobility concept. Mobility Management Framework in Software Defined Networks [3] proposes the simple adaptation of PMIPv6 in SDN. It relocates the LMA and MAGs from network entities to the centralized SDN controller as the applications and it keeps PMIPv6 mobility management signal follows. It keeps LMA, MAG, and tunneling mechanism for forwarding data packets. An Adaptation of Proxy Mobile IPv6 to OpenFlow Architecture over Software Defined Networking [4] suggests two mobility management schemes: OpenFlow-based PMIPv6 (OPMIPv6) and OPMIPv6 with a centralized mobility management controller (OPMIPv6-C). OPMIPv6 is similar [3]. OPMIPv6-C centralizes LMA and MAGs into single MAG at the SDN controller, but LMA and MAG are not unified. A Solution for IP Mobility Support in Software Defined Networks [5] proposes the centralized mobility management function to the SDN controller. It modified the packet header for packet forwarding. But, due to the rewriting policy, the access switch cannot handle more than one MN.

3 The Data Structure and Switch Categorization

It borrows the MN attachment assumption of PMIPv6. When the MN attachment is notified by layer-2 notification message and it contains the MN’s identifier.

3.1 System Architecture

The proposed structure assumes default SDN environment which consists of the centralized SDN controller, OpenFlow-enabled switches, and an Authentication, Authorization, and Accounting (AAA) server. The SDN controller is the basement of control applications and it supports network functions [6]. The MME is located on the SDN controller. An OpenFlow-enabled switch is able to satisfy the OpenFlow specification [1]. The AAA server authenticates the MN and provides the MN’s profile to the MME, if an MN has the right of the mobility service. The solid line and dotted line indicate the control flow and the data flow in Fig. 1.
3.1 Switch Categorization

According to the role of switches in SDN domain, they are able to categorize into Access Switches (ASs), Intermediate Switches (ISs), and Gateway switches (GWs). An AS is access point when MN is attached. A GW connects the Internet. ISs are located on a flow path between the ASs and the GWs. A Crossing Switch (CS) is specific switch which is located on the branch point between old flow path and new flow path at the handover.

3.2 Data Structures in the MME

The MME manages the information for MN at the Binding Cache. The Binding Cache is able to categorize two things: HNP information and Flow path. To provide them, the MME has two data structures: a GW-HNP mapping table and a Flow Matrix. These data structures are created before the MME provides the mobility service.

1) Binding Cache: The Binding Cache is a list of the Binding Cache Entries (BCEs). Each BCE has the MN’s information and Flow Matrix information for each MN. The MN information includes an MN’s Identifier (MN-ID), a Link-Layer Identifier (LL-ID), a List of MN’s Home Network Prefix (MN-HNP), and an Access...
Technology Type, as shown in Fig. 2 (a). The Flow Matrix information includes the Gateway number ($GW_i$), the AS number ($AS_j$) and an AS side Interface Identifier to which the MN is attached. The $GW_i$ and $AS_j$ mean two end points of the flow path for MN in SDN domain. They point the specific cell of the Flow Matrix.

2) $GW$-$HNP$ Mapping Table: For the mobility service for MN, the MME must allocate an HNP for the MN. If the MME get the HNP by the MN’s profile from the AAA, the MME selects the GW which handles the HNP, which based on the $GW$-$HNP$ mapping table, as depicted in Fig. 2 (b).

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

An element in the list of flow paths is described as a pair of upstream flow path and downstream flow path, as follows $Switch-ID(port$ $index$ $for$ $upstream, port$ $index$ $for$ $downstream)$. The pairs of pAS and CS, $P(pAS,CS)$, shows the switch lists which must be updated. The CS is intersection between the old path including pAS and the new path which the MN is currently attached. Thus, flow table of switches between the new AS (nAS) and the CS must be updated for packet forwarding, not between GW and nAS.

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 upstream flow path and downstream flow path. In this paper, we assume that two flow paths use the same switches, but there in/out ports are reversed.

The MME calculates two things: the best flow path between each AS and GW, and pAS and CS. The flow path is pre-calculated based on the network topology. The MME compares the flow paths and infer the CS. Each cell of the Flow Matrix is accessed by using GW index and AS index.

At the handover, MME is able to find the flow information of the new flow path which is based on the new AS index in the Flow Matrix. The flow table entries at the switch which between CS and nAS must update. Therefore, each cell of the Flow Matrix includes CS information for all another 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. It shows the list of MN which using the flow path. When the flow paths in the domain 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.

Fig. 3 shows an example of Flow Matrix. $C(GW_i,AS_i)$ indicates the flow paths between $GW_i$ and $AS_i$, 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), ISs(1,2), ISs(1,2), ASs(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 updates the flow table of switch on the new flow paths. Before updating, the MME decides which switches are updated 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 AS1.

The Flow Matrix provides several advantages. Firstly, it prevents the duplicate computation about same flow paths. The MME already calculates the flow paths between every ASs and GWs on the domain. Thus, when a new MN attaches to the AS, the MME does not calculate flow paths. MME just uses the corresponding cell of the Flow Matrix. Secondly, the MME can easily handle the topology change. Flow table entries affected by the under layer network topology change must be re-established and re-calculate. After re-computation, the MME records the flow path and updates the corresponding flow table entries at switches on the affected flow paths.

5 Conclusion

We are proposing the merged data structure and switch categorization for the mobility service in SDN. It is an adaptation of PMIPv6 concept to the SDN architecture. In order to support mobility in the SDN architecture, the mobility functions are unified from the separated PMIPv6 entities to MME. The proposed structure only uses OpenFlow signaling and PMIPv6 concept.

References