OpenFlow-based Routing Mechanism for Call Session State Migration in the IMS

MANABU ITO, SATOSHI KOMORITA, YOSHINORI KITATSUJI, and HIDETOSHI YOKOTA
KDDI R&D Laboratories, Inc.
2-1-15 Ohara, Fujimino-shi, Saitama, 356-8502
JAPAN
{mn-itou, sa-komorita, kitaji, yokota}@kddilabs.jp

Abstract: - Mobile broadband and the shift to All-IP network lead the integration of mobile network services and over-the-top (OTT) services. Increasing and unpredictable traffic caused by this trend make it important for the IMS (IP multimedia subsystem), which is the call control system, to be flexible and reliable. With that goal, the authors have focused on call session state migration, which allows the IMS to distribute call sessions so as to minimize the number of servers and/or restore call sessions dependent on a halted server. This paper proposes a mechanism to enable call session state migration without modifying the standard procedure in the IMS. The proposed mechanism maps a SIP URI (uniform resource identifier), which identifies communication resources such as the user, to a flow routed by OpenFlow switches. When the call session state migrates, this mechanism changes the mapping for proper SIP message routing. This paper implements the proposal and shows that the proposal routing mechanism has insignificant impact on call session establishment unless the large number of call session states are migrated.

Key-Words: - IMS, SIP, Call session state migration, and OpenFlow.

1 Introduction

In order to provide Voice over IP service, mobile network operators (MNOs) are developing the IMS (IP multimedia subsystem) [1], which is adopted as a call control system in GSMA [2] to ensure interconnectivity between MNOs around the world. MNOs plan to launch the RCS (rich communication suite), [3] which will enable mobile phone users to use instant messaging, live video sharing, and file transfer over the IMS. In addition, through leveraging their IMS development, MNOs are now looking into the possibility of integrating over-the-top (OTT) services (e.g., Google services, Facebook, and Skype).

The support of these future services in addition to legacy services (e.g., voice and SMS (short message service)) has resulted in an increase in signaling traffic in the IMS. It is difficult to estimate the required system capacity for the increase because OTT services are created by third parties outside MNO's conscious awareness. To continue to deal with increasing and unpredictable traffic without investing in extra capacity, an enhanced flexibility is desirable in the operation of the CSCFs (call session control functions) of which the IMS is composed. Here, we define flexibility as reconfigurable functionality of CSCFs process assignments on physical servers. In case of server failure or disaster, it is also desirable to ensure reliability by continuing the processing of CSCFs on other physical servers, thus preventing service disruption.

Studies [4][5] and a concept [6] have been published on the realization of flexibility and reliability by using virtualization technologies. Virtualization can flexibly assign and distribute available hardware resources to CSCFs as required. To optimize hardware resources and cope with server failure or disaster, virtual machines (VMs) can migrate among the available physical servers. However, this approach has two shortcomings: 1) it may cause service disruption, and 2) it may produce insufficient optimization in reducing the number of physical servers. Regarding 1), VM migration takes a long time when the VM treating a CSCF maintaining a large number of sessions. Although it may be a solution to have many VMs maintain a small number of sessions in a single physical server, this requires a large overhead of computational resources and makes it difficult to minimize the number of physical servers. As a result, this leads to issue 2). There is a trade-off among these issues in the VM migration approach.

In order to realize more flexibility, we have focused on call session state migration [7][8], which allows active call sessions to continue to be processed on another CSCF. We propose a routing mechanism in the IMS for call session state
migration without extensions to the standard procedure for control of a call/service session. Our routing mechanism is realized by utilizing OpenFlow [9], which provides an open standard protocol to program flow tables (that run at line-rate to implement firewalls, NAT, QoS, and to collect statistics) in different switches and routers [10]. Furthermore, we implement our proposal on real systems and verify its behavior and impact of call session state migration.

The rest of this paper is organized as follows. Section 2 describes the IMS architecture and the requirements of call session state migration. Section 3 proposes our OpenFlow-based routing mechanism in the IMS. Section 4 shows how much this mechanism affects the call session establishment. Finally, Section 5 concludes the paper.

2 Call Session State Migration

2.1 IMS Architecture and Call Session Control

Figure 1 shows basic IMS architecture. The IMS is composed of CSCFs that control call/service sessions for UE (user equipment), an HSS (home subscriber server) that is a database server for managing subscriber information, an IBCF (interconnection border control function) that is an entry/exit point for other IMSs, and SBCs (session border controllers) that provide a variety of functions for security and connectivity (e.g., access control, topology hiding, NAT traversal, protocol interworking, and media monitoring). These functions of the SBC are often integrated into the P-CSCF (described later) and the IBCF. There are three types of CSCF: an S-CSCF (Serving CSCF) that is the representative SIP server primarily dealing with call/service control and management, a P-CSCF (Proxy CSCF) that communicates with UE directly and establishes a secure connection to it, and an I-CSCF (Interrogating CSCF) that routes signaling messages to the S-CSCF managing the terminating UE.

In the IMS, SIP (session initiation protocol) [11] is used for call control between the CSCFs and UE. First, a UE registers with the IMS before obtaining IMS services. The UE conducts its registration with the S-CSCF via the P-CSCF and the I-CSCF. The S-CSCF assigned by the I-CSCF verifies the UE based on its authentication information stored in the HSS. After that, the UE uses IMS services by sending and receiving SIP messages to and from a correspondent UE via the P-CSCF, S-CSCF, and I-CSCF. For example, when the UE makes a call, the UE sends an INVITE message to the correspondent UE and establishes an active session to communicate the required information. The exchanged SIP messages take, as shown in Fig. 1, the following path: UE1 (caller), P-CSCF1, S-CSCF1, I-CSCF1, I-CSCF2, S-CSCF2, P-CSCF2, and UE2 (callee). Note that the messages need not be forwarded through the I-CSCF (I-CSCF1) within the originating network. Basically, if the originating network operator desires to keep its configuration hidden, it forwards the messages through I-CSCF1 to an entry point (I-CSCF2) in the terminating network.

Once CSCFs begin to handle the active session in making through terminating a call of UE, it is not changed for the CSCFs to process the session because the CSCFs have the states regarding the session. However, to cope with server failure and dynamically change the number of physical servers to match the network and resource usage situation, it is desirable that the call session state can migrate to another CSCF and continue to be processed (hereinafter called “call session state migration”).

The migration of a call session state at a CSCF creates an issue. From the standpoint of the counterpart CSCF or UE, the destination IP address generally changes when a call session state migrates from its original CSCF to another. Even in this situation, SIP messages must be forwarded to the new CSCF to which the call session state is migrated. The counterpart CSCF or UE still directs SIP messages for the call session to the original CSCF. This is because the information such as IP address in SIP headers (e.g., Route header and Via Header), which is the basis of SIP message routing, cannot be changed after establishment due to the SIP specification.
2.2 Requirements of Call Session State Migration

A key requirement for producing call session state migration into the IMS is a limited modification only in the IMS of a single MNO and within the standard procedures. In other words, we avoid changing, from the standards, the procedures for interoperation between the IMS and UE, and between the MNOs. Although the SBC provides the protocol for interoperation by eliminating mismatches between parameters in SIP messages, the SBC does not always deal with mismatches between the procedures. Introducing this requirement gives our proposed IMS the interoperability needed for roaming UE and the other MNOs without any changes in them.

In addition, it is desirable to involving new functions in the IMS by utilizing technologies based on open standards rather than proprietary technologies, to gain interoperability, reliably implement the functions, simplify their management, and avoid being locked into one particular vendor’s proprietary product [12].

2.3 Related Work

Studies dealing with call session state migration into the IMS have been published. In our study [7], an efficient mutual complementarity mechanism between P-CSCF and S-CSCF was proposed. This can support IMS reconfiguration with service continuity and cope with a single CSCF failure without backup servers. However, this coping mechanism may not work well in roaming case where the call session state of roaming UE is processed on P-CSCF and S-CSCF located on different MNOs. This is because the call session states need to be mutually transferred between the S-CSCF and P-CSCF over the proprietary protocol.

Y. Liu et al. [13] have proposed a mechanism to ensure reliability by restoring the register and session information, as backed-up in a standby server. This mechanism can reduce the load on backup servers and produce a rapid restore by storing only necessary information. They focus on call session state migration only between S-CSCFs.

In our other study [8], a restore mechanism for the entire IMS was proposed. This mechanism stores call session states in backup servers with proper timing and restores the states to other CSCFs. However, in order to maintain IP connectivity and forward SIP messages to the new CSCF after call session state migration, it is necessary to use a specific packet forwarding mechanism that requires a proprietary tunneling technology.

3 Proposal of OpenFlow-based Routing Method for Call Session Control Message

3.1 Approaches

In order to maintain IP connectivity, we consider assigning the same IP address to the same type of CSCF and forwarding SIP messages by using an identifier other than the IP address on networking equipment (e.g., switch or router), not CSCFs. When the call session state migrates to the new CSCF, the networking equipment needs to change the destination CSCF corresponding to the identifier by modifying the forwarding-tables so that the SIP messages are forwarded to the new CSCF. Alternatively, the identifier mapping of the SIP message needs to be changed. In this way, the CSCFs need not dynamically change forwarding destinations of SIP messages in accordance with call session state migration. In this paper, we propose a routing method that uses OpenFlow, which can flexibly change the flow-tables or the mapping. A point using OpenFlow is that a traffic flow is granularly distinguished and, a variety of fields in a MAC header of Ethernet through a transport header of TCP or UDP can be used to identify the flow.

3.2 Proposed Architecture for Migration of Call Session Information

Figure 2 shows the proposed architecture where new functions are introduced for routing SIP messages by using an OpenFlow switch. For the OpenFlow
switch to be able to identify the SIP messages as the flows, the SBC is extended (eSBC in Fig.2). When the eSBC receives a SIP message from a UE, it retrieves the SIP URIs of the UE included “From/To” headers and inserts the labels corresponding to the SIP URIs in specific fields recognized at the OpenFlow switch. The labels represent the combination of P/I/S-CSCFs admitting the UE. In our proposal, the labels are inserted in the MAC address fields because the original values (the MAC address) in the fields are not used as identifiers of the flows at the OpenFlow switch and the field has an ample length. Here, labels corresponding to the originating and terminating SIP URI are inserted in the destination and source MAC address fields. The OpenFlow switch forwards the SIP messages by identifying the flows with the label in the destination MAC address field and the destination IP address. In order to maintain the labels in the MAC address field when CSCFs send or receive the SIP messages, an OpenFlow Interface module (hereinafter called “OFIM”) is newly introduced in CSCFs. This module caches the labels in destination and source MAC address fields when the CSCFs receive the SIP message. Next, it inserts the same labels into MAC address fields when the CSCFs send SIP messages. A session controller has a copy of the call session states on CSCFs and restores the states to the other CSCF when that CSCF halts. When the call session state migrates, the session controller updates the label table in the eSBC and CSCFs by using the OpenFlow protocol.

The following subsections describe the label assignment solution and the detailed call flow of registration, call session establishment, and the call session state migration.

3.3 Label Assignment Solution
As described in the previous subsection, the label represents the combination of P/I/S-CSCF, as shown in Fig. 3. In this solution, the mapping information base in the eSBC and the OpenFlow switch become $O(S_p S_S S_t)$ in size, where $S_p$, $S_S$ and $S_t$ denote the number of P/I/S-CSCFs. For instance, the mapping information base in the eSBC is one million where there are 100 individual CSCFs. Our solution also specifies the field assignment of MAC addresses to distinguish the P/I/S-CSCFs. The individual CSCFs have IDs that are 13 bits or less long. The most significant nine bits are not used, to retain bits 6 and 7 for the global unique/local administrator and unicast/multicast flags defined by IEEE 802.3, respectively.

An alternative label assignment solution could be for each SIP URI to have a label. However, this increases the number of labels when the CSCFs already have a large number of UEs. With this solution, the required mapping information base becomes $O(N)$ in size, where $N$ denotes the number of SIP URIs. E.g., if each CFCS can admit the same number of UEs and more than $\sqrt{N}$ SIP URIs, our label assignment solutions have a smaller mapping information base than the naive (latter) solution. (State-of-the-art CSCF products admit five hundred thousand or more UEs.) Additionally, our solution allows the OpenFlow switch to hardly update the flow entry even when the call sessions state migrates. (When the CSCF halts and is removed from, or a new CSCF is installed in the IMS core network, the eSBC needs to update a large number of flow entries, e.g., five hundreds thousands flow entries in the case of the aforementioned CSCF product.)

3.4 Call Flow of Registration
Figure 4 shows the call flow of Registration through the OpenFlow switch. The system starts the initial settings. The session controller generates flow entries, composed of matching fields and their corresponding actions, and labels which represent
the combinations of P/I/S-CSCFs admitting the UE (step 1). The session controller adds the flow entries to the flow table in the OpenFlow Switch and the labels to the label table in the eSBC by using the OpenFlow protocol (steps 2 and 3).

When a REGISTER message enters the eSBC (step 4), it first tries to retrieve the label corresponding to the SIP URI, included in the “From” header of the message. If there is none, the eSBC assigns a label that represents the combinations of P/I/S-CSCFs (step 5). The eSBC notifies the mapping information to the session controller (step 6). This is done only if the label is newly assigned. Here, the session controller prepares to store the new call session state coming later (at steps 26 and 28). The eSBC inserts the label into the destination MAC address field and sends the REGISTER message to the I-CSCF (step 9), it inserts the same label into the destination MAC address field. The IP header has P-CSCF and I-CSCF as the source and destination IP addresses, respectively. In this way, the REGISTER message and the response message (401 Unauthorized) are forwarded to CSCFs through the OpenFlow switch. When the 401 Unauthorized message returns to UE1, the eSBC removes the label from the message (step 17) and forwards it toward UE1 through IP routing (step 18). UE1 sends the REGISTER message again with the authentication information (step 19), which is included in the 401 Unauthorized message, and the eSBC inserts the same label as at step 7 (step 20). The second REGISTER message and the response message (200 OK) are forwarded to CSCFs through the OpenFlow switch in a same way (steps 21-29). Finally, the eSBC removes the label from the 200 OK and forwards it towards UE1 (step 30).

3.5 Call Flow of Call Session Establishment

After registration, the UE establishes a call session in Fig. 5. In this figure, UE1 makes a call to UE2. First, UE1 sends an INVITE message to UE2 (step 1). The SBC inserts two labels corresponding to the originating (UE1) and terminating (UE2) SIP URIs, which are retrieved from the “From” and “To” headers, to the destination and source MAC address fields, respectively (step 2). Note that the label
represents a combination of the P/I/S-CSCF admitting the UEs. When the INVITE message is processed by the originating CSCFs (steps 3-6), the label corresponding to the caller’s SIP URI should be referred by the OpenFlow switch, but when the INVITE message is processed by the callee-side CSCF servers (steps 8-13), the label corresponding to the callee’s SIP URI is referred instead. For this purpose, the I-CSCF1 in Fig. 4 reverses the labels in the destination/source MAC address field (step 7). In the response message (steps 16-27), I-CSCF2 reverses the labels in the same manner as the request messages when the messages are forwarded from terminating-side to originating-side (step 21). The call session state is stored in the response with proper timing [8]. After sending ACK, they can communicate with each other like a VoIP.

3.6 Procedure of Call Session State Migration

Call session state migration is triggered when availability monitoring detects a CSCF failure or burden. Figure 6 shows the procedures by which the call session states on P-CSCF2 migrate to P-CSCF1. The migration takes the following steps:

1. The session controller chooses the available CSCF (e.g., P-CSCF1) to which the call session states are restored.
2. P-CSCF1 restores the call session states from the provided data.
3. The session controller notifies the new labels corresponding the restored session’s SIP URLs to P-CSCF1 and the eSBC.

After the migration, the message (e.g., 200 OK) related to the call session state is forwarded by the insertion of the new labels on P-CSCF1. The message (e.g., BYE) from UE is on the eSBC.

There are various monitoring methods to monitor the availability of CSCFs, including periodic workload querying, network availability check, and monitoring error and/or warning logs. The workload and/or number of admitted sessions of each CSCF can be used in choosing the available CSCF. This depends on what is monitored at the session controller according to network operator policy.

4 Implementation and Evaluation

4.1 Experimental Configuration

Figure 7 shows our experimental network configuration for verifying the behavior of the proposal on real system. In the IMS network, CSCFs and an eSBC are connected via an OpenFlow switch. A UE simulator connects to the IMS through the eSBC. A session controller connects to the CSCFs, OpenFlow switch, and eSBC via a switch. An HSS also connects to the CSCFs via the switch. Table 1 shows the network components.

The P/I/S-CSCFs and HSS were built based on the Open IMS core [14], which is an open-source SIP server. We implemented the required modules of the CSCFs for our proposed mechanism and the required software for the session controller and eSBC. We simulated a large UE load by using SIPp [15], which is also open-source load testing software for the SIP.

4.2 Measurement method

We verified how much the routing mechanism based on labels and call session state migration affect call session establishment. Multiple UEs randomly registered with four S-CSCFs through four P-CSCFs, and The UEs made a given number of call initiations per second. In the first experiment, we increased a call initiation rate and measured the CPU usage of the eSBC and CSCFs. In the second experiment, we increased the number of call session states which migrated from a P-CSCF (P-CSCF1) and measured the time for updating the labels at eSBC. After call session establishment, we performed call session state migration by halting P-
CSCF1. Regarding the call session state migration strategy, all the sessions in the halted P-CSCF1 were randomly distributed over the remaining (three) P-CSCFs.

4.3 Results and Discussion

Figure 8 shows the one-second average of P-CSCF1 CPU usage while increasing the call initiation rate. Note that the CPU usage of P-CSCFs was higher than the other types of CSCF and four P-CSCFs were similar in behavior. The CPU usage of eSBC (solid line) was lower than P-CSCF1 (dotted line). This indicates that introducing eSBC does not become a bottleneck at the call session establishment.

Figure 9 shows the amount of time for updating the label table caused by call session state migration. This indicates that call session state migration gives some influence on the call session establishment when the large number of call session states migrates. SIP messages sent from the UE before this update of label table are forwarded to the old CSCFs, which the call session states migrate from. This results in retransmissions of SIP messages and the call session establishment time will increase.

In the above experiments, we verified the behavior and showed the effect of call session state migration. When the large number of states migrates caused by server failure, the eSBC takes a long time to update of the bulk labels. It is necessary to distribute the processing across multiple eSBCs to prevent the eSBC from becoming a bottleneck. In the case of multiple eSBCs deployment, all eSBCs are required to synchronize the labels. To do this without overload of the session controller or eSBC, it is considered that multicast or broadcast functions in OpenFlow switches, which are added to the version 1.1.0 specification, offers a solution to synchronize all eSBCs. When an eSBC receives information of new labels from the session controller, it sends the same information to an OpenFlow switch. The OpenFlow switch clones the packets and sends them to each eSBC using the group table which is one component of flow table.

5 Conclusion

We have proposed a mechanism that can dynamically change the forwarding destinations of SIP messages by using OpenFlow together with call session state migration into the IMS. This provides the IMS to involve greater flexibility and reliability without modifying the standard procedure in the IMS. By implementation and actual measurement, we showed that the proposed routing mechanism has insignificant impact on the call session establishment and how much call session state migration gives influence on the call session establishment. Regarding a large number of routing label updates occurred at an eSBC, we also discussed how to distribute a large amount of its processing workload among multiple SBCs, and indicated that it is doable.

Acknowledgment: - This work has been supported by the Japanese Ministry of Internal Affairs and Communications funded project, “High-Reliability Cloud Services Control Platform Technologies.”

References:


