World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:4, No:11, 2010

Evaluation of Service Continuity in a Self-organizing IMS

Satoshi Komorita, Tsunehiko Chiba, Hidetoshi Yokota,
Ashutosh Dutta, Christian Makaya, Subir Das, Dana Chee, F. Joe Lin, and
Henning Schulzrinne

***Abstract*—**The NGN (Next Generation Network), which can provide advanced multimedia services over an all-IP based network, has been the subject of much attention for years. While there have been tremendous efforts to develop its architecture and protocols, especially for IMS, which is a key technology of the NGN, it is far from being widely deployed. However, efforts to create an advanced signaling infrastructure realizing many requirements have resulted in a large number of functional components and interactions between those components. Thus, the carriers are trying to explore effective ways to deploy IMS while offering value-added services. As one such approach, we have proposed a self-organizing IMS. A self-organizing IMS enables IMS functional components and corresponding physical nodes to adapt dynamically and automatically based on situation such as network load and available system resources while continuing IMS operation. To realize this, service continuity for users is an important requirement when a reconfiguration occurs during operation. In this paper, we propose a mechanism that will provide service continuity to users and focus on the implementation and describe performance evaluation in terms of number of control signaling and processing time during reconfiguration.

***Keywords*—**IMS, SIP, Service Continuity, Self-organizing, and Performance.

I. Introduction

IN recent years, the NGN, which is a standard platform for providing highly advanced network services by integrating fixed and mobile communication networks, has been attracting attention. The IMS (IP Multimedia Subsystem) [1] is its key technology, and has been standardized by 3GPP [2]. The IMS is standard specifications for a multimedia services platform operating over an all-IP network, which provides QoS controls, accounting, and various services from third-party service providers.

The IMS is currently constructed and operated on a large scale by telecommunication operators. They have their own home network, which is equipped with IMS components such as several call control servers, subscriber database servers, and various application servers. A user terminal or UE (User Equipment), connects to the IMS core network through access networks such as wireless and fixed networks. However, the IMS is a complicated system with many components. In addition, the operator needs to take care of the reliability and redundancy of the IMS. Thus cost effective implementations of the IMS are desired.

Self-organization offers an attractive way to reduce the deployment and operational costs of such a complex system. A self-organizing approach has been studied for network configurations, such as ad hoc and zero-configuration networks, and it can facilitate the incorporation of each component. Furthermore, in IMS, components are defined as functional components and can be considered separately from physical nodes. Thus, dynamic adaptation with the nodes merging and splitting IMS functional components can realize effective IMS operation in terms of resource usage. In its current form, IMS architecture and protocols do not have mechanisms that can easily help IMS components to self-organize. Thus, we have proposed additional features and mechanisms to support self-organizing capability [3].

In addition to configuration before the startup of IMS, efficiency-enhancing IMS reconfiguration while already in operation is also possible, depending on the network and resource usage situation. However, this has largely effect on many UEs that register with IMS components and establish control sessions. Thus, this self-organizing mechanism also needs to support service continuity to allow the UEs to continue their services seamlessly after the reconfiguration. In this paper, we propose a method to continue the service by effective notification to the affected UEs without large impact on the existing IMS.. This method allows the UEs register with IMS and establish the needed session again according to standard specification. Further, we implement our proposed method on real systems and verify its behavior and evaluate the performance.

The rest of this paper is organized as follows: Section II provides an overview of the basic IMS configuration and the self-organizing IMS with their call control messages. Section III describes related works dealing with self-organization. Section IV proposes a method for service continuity in a self-organizing IMS. Section V describes the experiments and their results. Finally, Section VI concludes the paper.

Satoshi Komorita, Tsunehiko Chiba, and Hidetoshi Yokota are with the KDDI R&D Laboratories, Inc., Saitama, Japan (corresponding author to provide phone: +81-49-278-7895; e-mail: sa-komorita@kddilabs.jp).

Ashutosh Dutta, Christian Makaya, Subir Das, Dana Chee, and F. Joe Lin are with Telcordia Technologies, Inc. Piscataway, NJ, USA.

Henning Schulzrinne, are with Columbia University, NY, USA

II. Overview of IMS and Self-organizing IMS

A. Basic IMS Configuration

Fig. 1 shows the basic IMS network configuration. The following IMS components are located in the IMS core network that an operator manages and operates: the HSS (Home Subscriber Server), which is a database server for managing subscribers, a S-CSCF (Serving CSCF) that is the main SIP server for call control, a P-CSCF (Proxy CSCF) that communicates with the UE directly and establishes a secure connection to it, an I-CSCF (Interrogating CSCF) that is a kind of resolver of SIP message routing and a gateway to other IMS domains, a PCRF (Policy and Charging Rules Function) that conducts QoS control, an MGCF (Media Gateway Control Function) and a BGCF (Breakout Gateway Control Function) that interconnect to the existing circuit services. Each operator has these IMS functionalities and interconnects their own IMS domain to another operator's one by a secure and quality guaranteed route (e.g., via a dedicated line). A UE connects with the IMS through the IP core network and wireless access networks such as EV-DO [4] and LTE (Long Term Evolution) [5], and a fixed access network such as a home network or an office network. All IMS components are functional entities, thus, they can run on any physical nodes except specific components that require dedicated lines.

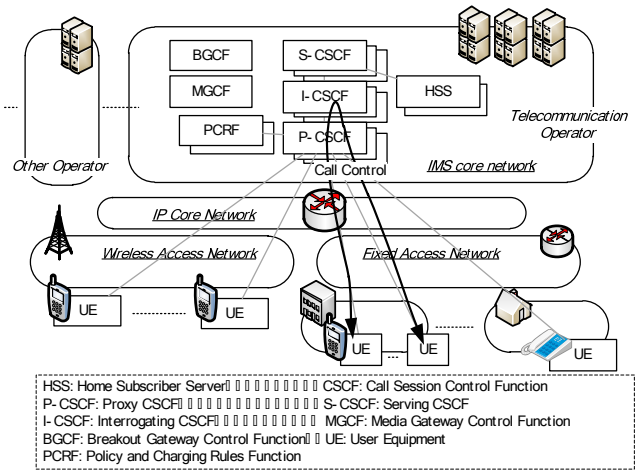

Fig. 1 Overview of basic IMS configuration

In IMS, SIP is used for call control between UEs and IMS components. First, a UE registers with the IMS before availing IMS services. The UE sends a registration message to the S-CSCF via the P-CSCF of the IMS to which the UE belongs. The S-CSCF verifies the UE based on the UE information stored in the HSS, and if successful, the S-CSCF registers the UE. After that, the UE establishes a secure IPSec connection to the P-CSCF and uses IMS services by sending and receiving SIP messages to and from a correspondent UE and ASs (Application Server) via the P-CSCF and the S-CSCF. For example, when a UE makes a call, the UE sends an INVITE message to a correspondent UE and establishes an active session to communicate the required information. CSCFs handling the active session are decided when the session is established, and cannot be changed because the CSCFs have states required for processing the session.

B. Self-organizing IMS

The basic concept of the self-organizing IMS is shown in Fig. 2. In the Resource Pool, there are physical nodes that are capable of running several IMS components. According to the load and network conditions, the nodes can adapt the components and run one or multiple instances of them. They negotiate with each other and take over missing components to maintain the IMS automatically in case of trouble. When the physical resource is not needed any more, the processing is moved to other nodes and the unnecessary node is removed. Thus this system realizes efficient redundancy and effective resource usage.

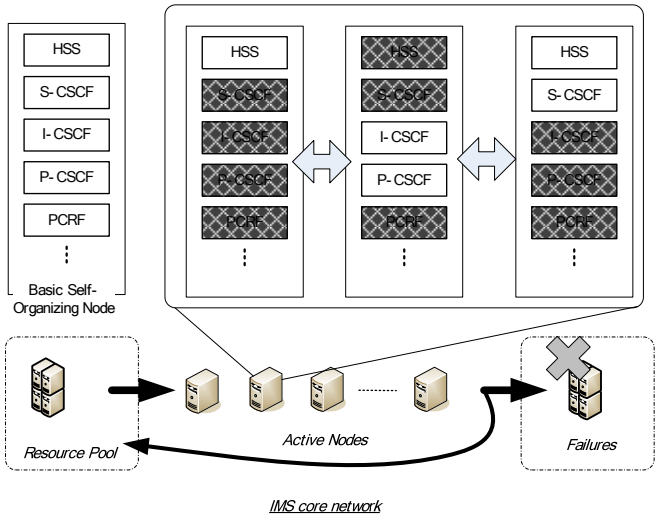

Fig. 2 Concept of the self-organizing IMS

The self-organizing IMS can be structured in one of two modes: centralized or distributed. For sake of simplicity and easy optimization of the entire system, we consider a centralized approach that has a central control node called a master node that maintains operator policy and state information for all nodes under its control. On the basis of supplied information, the master node assigns IMS components to their roles as nodes and gives instructions to them, and the nodes make the required changes. For these behaviors, additional features such as node discovery and role request and assignment, and interaction protocols between the nodes and the master node have been proposed in [3]. In addition, after the IMS configuration is changed, those IMS components affected at each node need to take care of active sessions that were established before the reconfiguration. Otherwise, services are interrupted and it degrades the user experience.

III. Related Work

With the emerging 3GPP LTE technology, self-organizing networks are envisioned as the new model for the next generation OSS/BSS (Operations and Business Support

System). The 3GPP [1] and Next Generation Mobile Networks Alliance [6] have standardized a set of capabilities known as a self-organizing network to improve OAM (Operation and Maintenance). Since IMS is envisioned as a main component for the deployment of LTE or 4G wireless networks, for example, in order to provide real-time applications and other value-added services, it is important to extend the self-organizing network concept to the core IMS networks rather than only to the LTE RAN and EPC (Evolved Packet Core).

The concept of self-organizing IMS networks has not been widely studied to the best knowledge of authors. Here, it is important to understand that the self-organizing IMS is different than the P2P-SIP concept [7] on which significant research has been done. The P2P-SIP is aim at discovering a target server quickly in distributed SIP servers. Bessis [8] describes the performance analysis and benefits of running multiple SIP servers on the same host. That paper shows how to design IMS networks to maximize IMS server co-location and explains which types of SIP calls can benefit from the co-location of IMS servers. Fabini et al. [9] describe an optimal IMS configuration with respect to architecture and QoS aspects that demonstrates the feasibility of an IMS system implementation within a single virtual device (all-in-one). In [10], a distributed IMS architecture has been proposed by representing network functional elements in DHT (Distributed Hash Tables) overlay networks. The main focus was to distribute S-CSCF functionalities by using an overlay network where these functionalities are merged in one node, which is called IMS DHT. A common issue with DHT overlay networks is related to the number of operations or message in DHT to retrieve information or data.

In a self-organizing IMS, efficiency requires that reconfiguration of the IMS is also executed while the IMS is in operation. From the perspective of users, it is important to continue their services without interruption. SCC (Service Centralization and Continuity) [11] is defined as continuing an IMS session when a UE executes a handoff by establishing new session related to the old one. In [12], a method has been proposed to use the same session by updating the SIP servers responsible for active sessions. This method sends a specific message to SIP clients, then updates session information to continue to use the same session. However, these are fundamental technologies for session continuity. In the self-organizing IMS, we need to take care of the impact on UEs and how to manage those affected. This paper deals with service continuity from the perspective of a self-organizing IMS.

IV. Service Continuity in Self-organizing IMS

A. Requirements for Service Continuity

In a self-organizing IMS, the IMS components are dynamically assigned and moved to appropriate physical nodes when IMS reconfiguration happens. However, in the IMS, each component has its own state and information for call control. A UE also keeps its connection state to the IMS and the state of the call. In particular, if an inconsistency occurs between the UE and the P-CSCF that communicates with the UE directly, and the S-CSCF that manages the registration information of the UE, the service will be interrupted and terminated.

To solve the inconsistency between those IMS components and the UE, there are generally two approaches: with the UE uninvolved and the UE involved. The former tries to hide the reconfiguration from the UE. However, that requires a hiding mechanism at IP layer, which requires new equipment such as a SIP proxy in front of the P-CSCF or a particular kind of IP routing. In addition, a handover procedure for the state registered in the IMS is needed. These incur large modification to the existing IMS including standard call flows. The latter tries to inform the UE of the reconfiguration and make the UE update information affecting the continuity. In this method, the UE that is informed about the reconfiguration registers with the IMS again, and continues the procedure of the call. Thus, processing and overhead in the IMS can be small.

In view of the overhead in the IMS, we propose a method based on the latter approach in this paper. In this approach, there are the following requirements. First, the method determines which the UEs are affected by the reconfiguration because many UEs connect to the IMS and relationships between CSCFs and UEs are dynamically assigned when UEs register. The method needs to determine which UE connects to which CSCFs, and notifies the UE if the CSCFs are changed. Second, a notification mechanism to the UE is needed. In the current IMS, there is no way to notify the reconfiguration information to UEs. Third is to minimize the service interruption time from the reconfiguration to the end of the appropriate procedure at the UEs. During the interruption time, control messages for the call cannot be processed. If a correspondent UE sends a message to the UE, the message is dropped. This failure affects service continuity. Finally, impact on the existing IMS should be minimized because IMSs are already deployed in some operators' networks.

B. Proposed Method

In this paper, we focus on the SUBSCRIBE session that is established between a UE and S-CSCF through P-CSCF in order to notify the UE's registration state to the UE. We propose a method to notify the reconfiguration to the UE by an extension of this session. In this method, the P-CSCF and S-CSCF, which are responsible for the UE, learn and store information about each other in the establishment of this session. Then, in case of any change in either of them, the other can recognize it and notify it to the UE. This allows the IMS to inform only affected UEs. The UEs execute registration and session handoff procedures [11] through the P-CSCF designated by the notification message. The P-CSCF asks HSS for S-CSCF of the UE in the same way as the basic IMS and processes the UE registration. The assignments of P-CSCF and S-CSCF to UEs are calculated and provided to CSCFs and HSS from a master node.

Fig. 3 shows an overview of this proposal. In this figure, UE#1 and UE#2 register and establish a SUBSCRIBE session with S-CSCF#2 through P-CSCF#1 and P-CSCF#2. UE#K also registers with S-CSCF#M through P-CSCF#2. In case (A) where the master node tries to move all UEs using S-CSCF#2 to S-CSCF#1, the master node sends commands to all P-CSCFs and HSS, and then P-CSCF#1 and P-CSCF#2 send a NOTIFY message to UE#1 and UE#2. HSS also updates the assignment of S-CSCF to UEs based on the master node. In case (B) where the master node tries to move all UEs using P-CSCF#2 to P-CSCF#1 and P-CSCF#N, the master node sends commands to all S-CSCFs, and then S-CSCF#2 and S-CSCF#M send a NOTIFY message to UE#2 and U#K. In the both cases, the UEs register with the indicated CSCFs and continue their services.

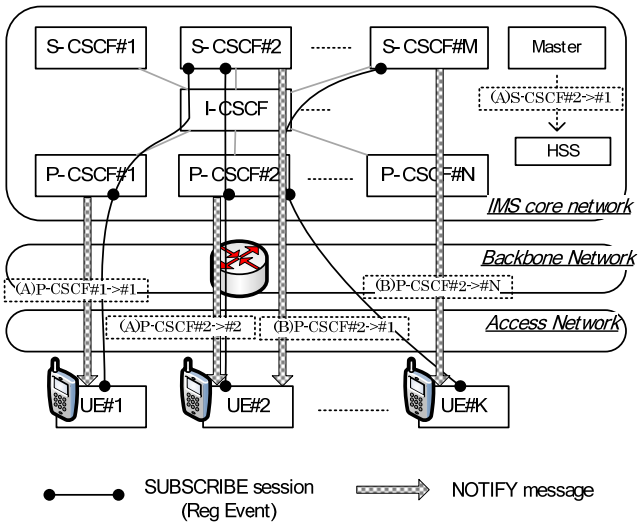

Fig. 3 Overview of Notification Method

This method is also applicable by sequential execution when both S-CSCF and P-CSCF are changed. In addition, this is valid when some failure takes place at P-CSCF and S-CSCF because the failed CSCF is not involved with this notification sequence. This method can send notifications effectively and quickly to the affected UEs and can be realized by an extension of the subscribe session and an additional master node compared with the basic IMS. Thus it satisfies the requirements described in the previous section.

C. Call Flows

In this section, the proposed call flows for notification and session continuity are described. Fig. 4 shows the first registration flow when using AKA (Authentication and Key Agreement) [13] as authentication method. This flow is almost the same as a standards register flow. However, during a standard flow, P-CSCF and S-CSCF obtain and store the information required for notification procedure in (17), (19), and (20). First, a UE sends a REGISTER to P-CSCF (1) and P-CSCF forwards it to I-CSCF (2). The I-CSCF asks HSS which S-CSCF is assigned to the UE (3) and forwards REGISTER to that S-CSCF (4). The S-CSCF asks HSS for authentication information (5) and sends back a 401 unauthorized message to the UE via the I-CSCF and the P-CSCF (6, 7, 8). The UE sends REGISTER again with authentication information to the S-CSCF (9, 10, 11). The S-CSCF verifies the authentication and gets and updates the UE's information on the HSS (12), then sends back a 200 OK message to the UE (12, 13, 14). After that, the UE sends SUBSCRIBE to the S-CSCF through the P-CSCF (15, 16). Here, the S-CSCF stores the P-CSCF of the UE (17), and sends back a 200 OK to the UE (18). The P-CSCF also stores the S-CSCF of the UE (19) and session information to ensure the P-CSCF sends NOTIFY (20), and then forwards the 200 OK to the UE.

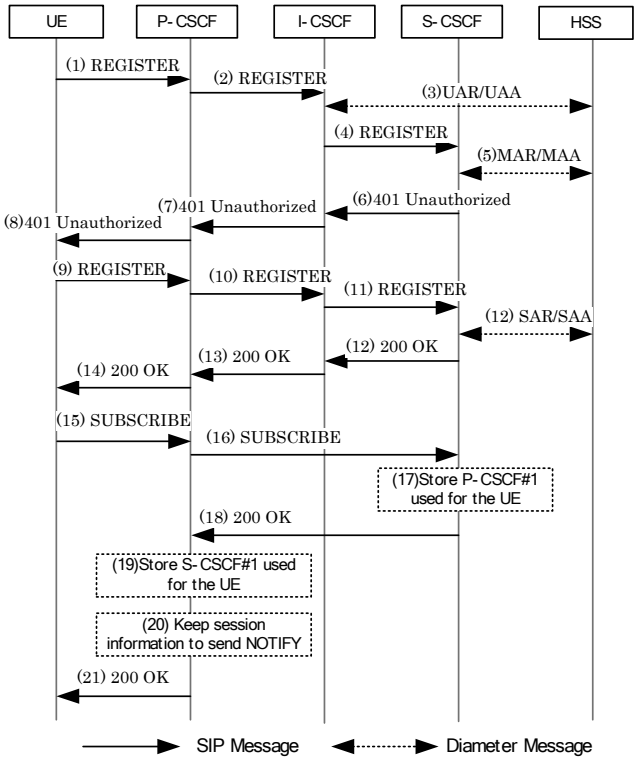

Fig. 4 Registration Flow

Fig. 5 shows the P-CSCF change flow from P-CSCF#1 to P-CSCF#2. First of all, the UE is already registered to the S-CSCF through P-CSCF#1 and establishes an active call session with an INVITE through them (1). Then the master node decides on a reconfiguration and informs all S-CSCFs of the change. If P-CSCF#2 is not running yet, the master node also sends commands to launch it at the same time. (2). The S-CSCF finds out which registered UEs use P-CSCF#1 based on the stored information, and sends the UE a NOTIFY including new P-CSCF information via the new P-CSCF (4). Here the new P-CSCF is P-CSCF#2. The P-CSCF#2 forwards the NOTIFY to the UE (5). Upon receipt of the notification from the S-CSCF, the UE sends back a 200 OK (6, 7), registers and subscribes through the P-CSCF#2 (8). The UE also takes care of the active session. The UE sends an INVITE with a session transfer identifier that associates the previous active session with the new session generated by this INVITE (9, 10,

11). The correspondent UE sends back a 200 OK (12, 13, 14) and the UE sends an ACK (15, 16, 17). Finally, the UE sends a BYE to disconnect the previous active session.

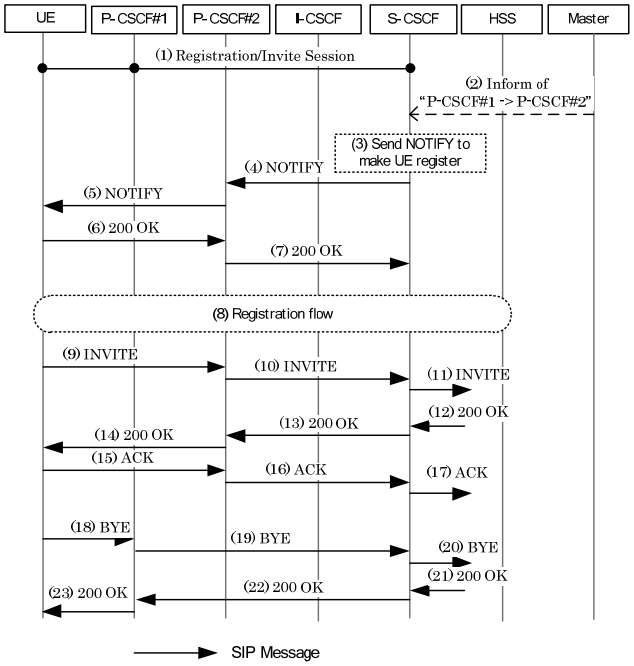

Fig. 5 P-CSCF Change Flow

Fig. 6 shows the S-CSCF change flow from S-CSCF#1 to S-CSCF#2. This flow is similar with the P-CSCF change. However, in addition to the notification to the UE, the master node needs to inform HSS of the change because HSS needs to assign other S-CSCF when the UE registers. The UE is already registered to S-CSCF#1 through P-CSCF and establishes an active call session by sending an INVITE through them (1). Then the master node decides on a reconfiguration and informs all P-CSCFs of the change. If S-CSCF#2 is not running yet, the master node also sends commands to launch it at the same time. (2).The master node also informs HSS of the change to update assignments of S-CSCF (3, 4). The P-CSCF finds out which UEs was associated to S-CSCF#1 based on the stored information, and sends NOTIFY message to the UE (6). Here, the NOTIFY indicates the same P-CSCF because the P-CSCF does not change. The UE sends back a 200 OK (7) and registers and subscribes through the P-CSCF. This time, HSS informs I-CSCF of the new S-CSCF in the registration sequence, thus the UE can register with the new S-CSCF. Here, the new S-CSCF is S-CSCF#2. Then the UE processes the session continuity of active sessions in the same way as from (9) to (23) of P-CSCF change flow.

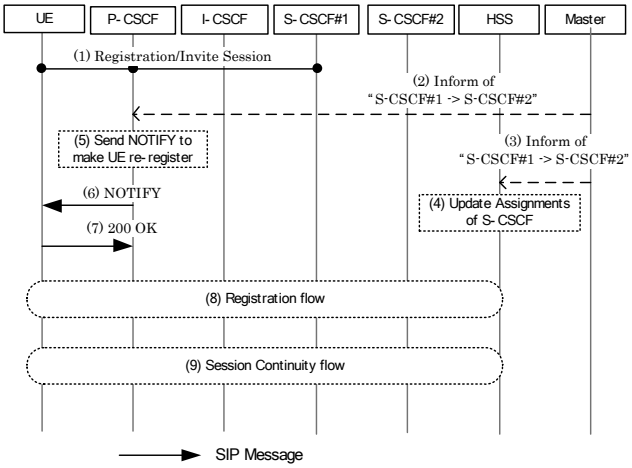

Fig. 6 S-CSCF Change Flow

V. Experiment and Results

A. Experimental Configuration

Fig. 7 shows our experimental network configuration for verifying the method's behavior and performance. In the IMS core network, there are four self-organizing nodes (node#1-node#4) and a master node. HSS and I-CSCF are running on node#1 and S-CSCFs are running on node#2 and node#4, and P-CSCFs are running on node#3 and node#4 under control of the master node. UEs and a UE Emulator connect to the IMS through hubs and a router by using Ethernet without transmission delay. A monitor machine captures packets passing through the hubs.

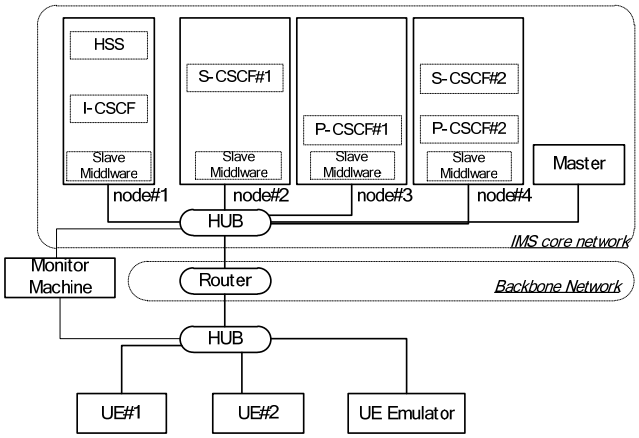

Fig. 7 Experimental Network Configuration

We implemented the required software for the CSCFs, HSS, UE, slave middleware on nodes, and the master node for our proposed method. CSCFs and HSS were built based on NIST SIP [13], which is an open-sourced SIP stack. The master node and slave middleware, which are our original software, communicate to get information about each node and dictate which IMS functions are running on the node. For a UE, we adopted and extended SIP Communicator [15], which is an open-source IMS client software. The UE Emulator, which can handle a thousand sessions but only controls signaling, is built

based on IMS bench software [16]. Each node uses Fedora 10 as its OS, and is implemented on a generic PC (CPU: Intel Atom 230 1.6 GHz, Memory: 2 GB, HDD: 80 GB, NIC Intel Pro/100).

B. Measurements and Evaluations

In the experiment, first UE#1 and UE#2 register with the IMS through P-CSCF#1 and S-CSCF#1, and then the UEs make a call. Here, P-CSCF#2 and S-CSCF#2 on node#4 are not running. During the call, the master node decides to add new P-CSCF function to increase processing capacity of P-CSCF in the IMS, and then send out commands to node#4 to launch P-CSCF#2 and also send out commands to S-CSCF#1 to change the P-CSCF of UE#1 from P-CSCF#1 to P-CSCF#2. UE#1 continues its call using our proposed method after the reconfiguration. In a similar way, the master node decides to add S-CSCF on node#4 to increase processing capacity of S-CSCF in the IMS and changes the S-CSCF of UE#1 from S-CSCF#1 to S-CSCF#2. Through the execution of the above scenario, we evaluate the performance of our proposal. By using the UE Emulator, several instances of UEs ranging from 10 through 1000 register with the IMS through P-CSCF#1 and S-CSCF#1. Then their P-CSCF is changed to P-CSCF#2, and their S-CSCF is also changed to S-CSCF#2 that is controlled by the master node.

As measurement indexes, we use processing time for service continuity which is the time from when the master node sends out commands to each node until the UE completes the registration and handoff of active sessions. For the UE Emulator, we measure the total time from when the master node sends out commands until all UEs complete their registration. These measurement indexes are computed based on packets captured at the monitor machine during the experiments.

C. Results

The desired behavior of our proposal was verified in this experiment. Fig. 8 and Fig. 9 show the traffic volume of control signaling and VoIP signaling during processing of service continuity. In both cases, UEs communicate with each other after call establishment, and then the P-CSCF and S-CSCF of UE#1 were changed. Although the CSCF is changed, VoIP communication continued without disruption and terminated correctly. In the IMS, VoIP media traffic is not transferred through CSCFs. Thus VoIP can continue seamlessly unless there is a failure of control signaling due to the change of CSCFs

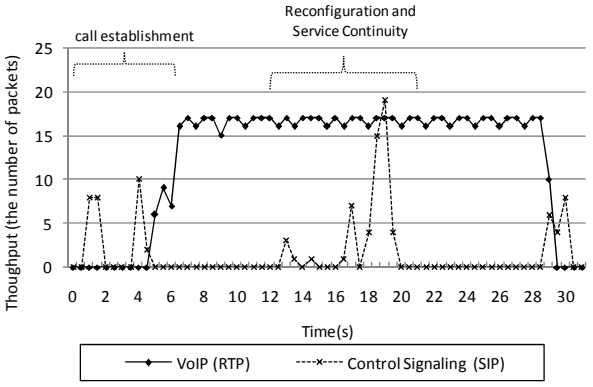

Fig. 8 Traffic when P-CSCF changed

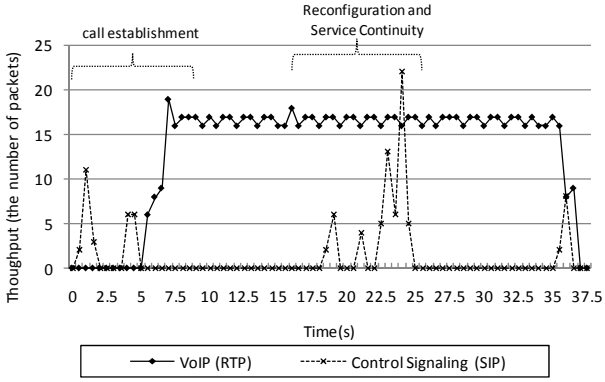

Fig. 9 Traffic when S-CSCF changed

Fig. 10 shows the details of the processing time for service continuity when the S-CSCF and P-CSCF of UE#1 were changed. It takes about 6 seconds for an S-CSCF change, and about 7 seconds for a P-CSCF change. In this experiment, these processing times are consumed by the processing and waiting at each node because there is no transmission delay and congestion in this network. In the former case, the dominant factor that affects the processing is registration because it takes several seconds to launch the new S-CSCF on node#4 and then complete the registration. In contrast, the dominant factor due to processing is Notification from the IMS to the UE in the latter case. However, this is also attributed to the start-up time of a new P-CSCF on node#4 because the notification is sent through the new P-CSCF.

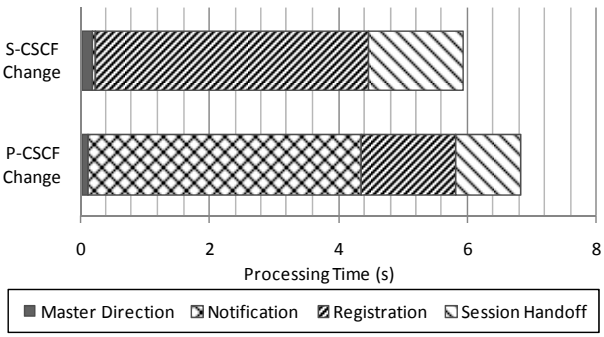

Fig. 10 Processing Time for Service Continuity

Fig. 11 shows the total processing time when P-CSCF and S-CSCF are changed by using the UE Emulator. In both cases, the processing time increases linearly, but processing time when the S-CSCF is changed takes about 6 times longer than when the P-CSCF is changed. Fig. 12 shows the success rate for service continuity. The rate when the S-CSCF changed was always 100%. However, the success rate when the P-CSCF changed decreased as the number of UEs increased. Here, the failure of service continuity means that the UE cannot complete the processing of the service continuity and its service is terminated

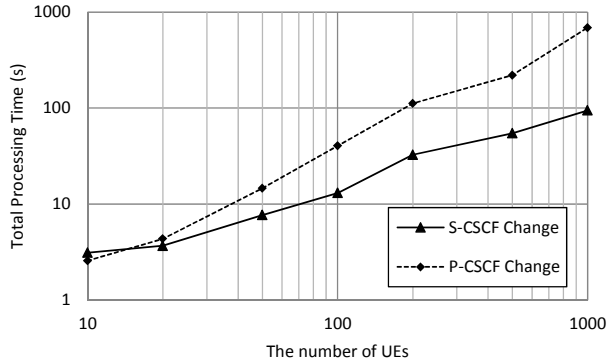

Fig. 11 Total Processing Time

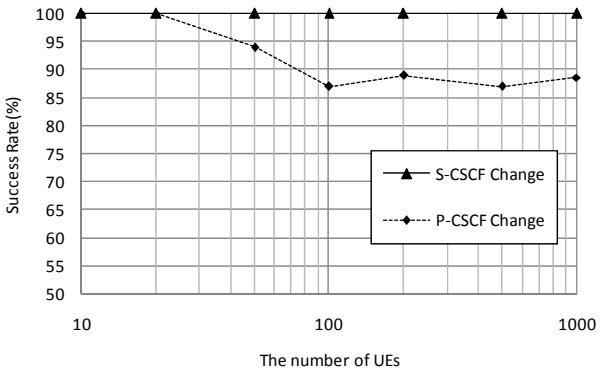

Fig. 12 Success Rate of CSCF Change

The difference between the cases seems to be due to the processing node for notification and registration. When the P-CSCF of UEs was changed from P-CSCF#1 on node#3 to P-CSCF#2 on node#4, S-CSCF#1 on node#2 sent notifications to all affected UEs through P-CSCF#2 regardless of the processing capability of other nodes such as node#4. This behavior results in the retransmission of notifications and exceeds the processing speed of registration at P-CSCF#2 on node#4. Thus, the success rate decreases. On the other hand, when the S-CSCF of the UEs was changed from S-CSCF#1 on node#2 to S-CSCF#2 on node#4, the P-CSCF#2 on node#3 sent notifications to all affected UEs directly. However, all registration comes to the P-CSCF#2 and exceeds its processing power so that it cannot send the needed notifications.

D.Consideration

In this experiment, we were able to verify service continuity due to IMS reconfiguration that used our proposed method. The overall time required for IMS reconfiguration is about 7 seconds. We think this time is fast enough to continue the service because a UE can try to continue its service if the UE recognizes the reconfiguration before re-transmission of other control signaling expires. When there are many affected UEs, quick notifications within the limited time become more important. The master node needs to take care of the capability of the processing CSCFs.

However, some open issues remain. In our method, the master node does not directly deal with the affected UEs and each CSCF sends notifications based on its stored information. This can improve the performance to send notifications quickly without the dedicated management nodes. On the other hand, each CSCF does not help with or take account of the processing capability and bandwidth of other CSCFs. This behavior results in congestion of control signaling and failure of service continuity. Thus, in future, some mechanisms are required to control the rate of sending the notifications.. For example, the master node can inform CSCFs of their capability. It is also possible that related CSCFs communicate with each other about their capability.

Security is another issue. A UE uses IPSec to connect to P-CSCF in order to secure the communication. If the P-CSCF of the UE changes, notification is sent out over the secure connection. In particular, in case that a previous P-CSCF crashes, the UE does not have any choice but to receive the unsafe notification from another P-CSCF. Thus, it is possible that a malicious attacker could send a bogus notification to the UE and redirect the UE to a fake P-CSCF. In our proposal, it is difficult to send a bogus notification because the notification includes Call-ID and tags that are unique keys generated in the secure previous registration process. However, there is still probability of some attacks, such as TCP session hijacks [17]. Thus, some security mechanism for notifications is also required.

VI. Conclusion

In this paper, we introduced self-organization capability of IMS that provides cost effective operation and makes effective usage of resources. To realize this, service continuity for UEs affected by an IMS reconfiguration is an important aspect. We proposed an approach to realize it without a large impact on specification and implementation of the IMS. The proposed method sends notifications to the UEs effectively by using the subscribe session and make the users to register and help the session handoff procedures.. Further, we implemented the method and demonstrated its behavior and performance. The performance results show the realization of IMS reconfiguration completes in seconds without any media interruption. This time seems to be fast enough considering re-transmission time of control signaling. Although the required time increases with the number of the affected UEs

and capability of CSCFs, the performance could be improved by appropriate control and distribution of the notifications according to the capability.

REFERENCES

[1] 3GPP TS 23.228 v9.0.0, “IP Multimedia Subsystem (IMS); Stage 2,” Rel. 9, Sept. 2009.
[2] Third Generation Partnership Project (3GPP), www.3gpp.org
[3] A. Dutta, C. Makaya, S. Das, D. Chee, J. Lin, S. Komorita, T. Chiba, H. Yokota, and H. Schulzrinne, “Self Organizing IP Multimedia Subsystem,” In Proc. of 3rd IEEE Int'l Conf. on Internet Multimedia Systems Architecture and Application (IMSAA'09), Dec. 2009.
[4] 3GPP2 X.S0011-001-C v3.0: "cdma2000 Wireless IP Network Standard: Introduction," 2006.
[5] 3GPP: "TS36.300 Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2," 2009.
[6] NGMN Alliance: www.ngmn.org
[7] E. Marocco, A. Manzalini, M. Sampò, and G. Canal, “ Interworking between P2PSIP Overlays and IMS Networks – Scenarios and Technical Solutions,” www.p2psip.org
[8] T. Bessis, “Improving performance and reliability of an IMS network by co-locating IMS servers,” Bell Labs Technical Journal, Vol. 10, No. 4, pp. 167-178, 2006.
[9] J. Fabini, P. Reichl, A. Poropatich, R. Huber, and N. Jordan, “IMS in a Bottle, Initial Experiences from an OpenSER-based Prototype Implementation of the 3GPP IP Multimedia Subsystem,” in Proc. of the Int'l Conf. on Mobile Business (ICMB'06) , June 2006.
[10] M. Matuszewski and M. Garcia-Martin, “A Distributed IP Multimedia Subsystem (IMS),” IEEE Int'l Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'07), pp. 1-8, 2007.
[11] 3GPP TS 23.237 v9.0.0, “IP Multimedia Subsystem (IMS) Service Continuity; Stage 2,” Rel. 9, Sept. 2009.
[12] S. Komorita, T. Kubo, T. Hasegawa, and H. Yokota, " Network-controlled SIP Server Switching Methods for Active SIP Sessions," IASTED Parallel and Distributed Computing and Networks(PDCN '2009), Feb, 2009.
[13] 3GPP TS 33.102 v9.0.0, “3G security; Security architecture” Rel. 9, Sept. 2009.
[14] NIST SIP, http://www-x.antd.nist.gov/proj/iptel/
[15] SIP Communicator, http://sip-communicator.org/
[16] IMS Bench SIPp, http://sipp.sourceforge.net/ims_bench/
[17] B. Harris and R. Hunt, "TCP/IP security threats and attack methods," Computer Communications, Vol. 22, Issue 10, pp. 885-897, 1999.