Elastic Core | Network Virtualization | SDN/OpenFlow

Technology Reports

A Study to Achieve High Reliability and Availability on Core Networks with Network Virtualization

*Core networks need to efficiently handle increasing service control signals due to the spread of smartphones, while minimizing adverse effects on user services when failures occur. In response to these demands, we propose an "Elastic Core" architecture that achieves a flexible core network with high reliability and availability. The architecture utilizes and expands virtual server and SDN/OpenFlow™*1 technologies currently used in data centers for carrier networks. This article reports on evaluation tests result using an IMS test bed.*

Research Laboratories *Motoshi Tamura*
Tetsuya Nakamura
Takahiro Yamazaki
DOCOMO Innovations, Inc. *Yuki Moritani*

1. Introduction

Due to the rapid spread of smartphones in recent years, communications traffic and service control signals on networks are increasing. Therefore, core networks*2 must be equipped with sufficient service control resources that are able to handle service requests from users to provide stable services. On the other hand, some service control servers on the core network are designed to handle users in groups, according to their area of residence or their subscriber numbers. Therefore, those servers must be prepared to handle the predicted peak service requests from users.

Furthermore, these servers keep service processing conditions (state information) for each user. Thus, if a failure occurs in a server, it loses state information which affects services for all relevant users. For this reason, reliability is insured by redundant configurations with Act and Stand-by systems, or by load distribution with multiple nodes.

In contrast, data centers that offer cloud services using general-purpose servers can be configured and allocated flexibly with virtual servers or Software Defined Networking (SDN)/OpenFlow technologies. These techniques enable server resources to be assigned flexibly in response to the amount of service requests. If a failure occurs in a server, processing can be freshly assigned to another server. By providing substitute processing for new service requests in this way, services can be provided continuously and independently of the server failure.

Accordingly, applying these virtual technologies to telecommunications carrier networks in a similar manner offers an economical way to minimize the impact on services due to failure or excessive amounts of service requests beyond those estimated, and therefore holds the promise of achieving core networks with greater availability.

However, to apply the virtual technologies used in present-day data centers to carrier networks, each technology must have improved capabilities to operate in real time, have sufficient reli-

*1 **OpenFlow™**: A trademark of Stanford University in the United States.

*2 **Core network**: A network consisting of service control nodes and subscriber-information management equipment. Mobile terminals communicate with the core network via the radio access network.

ability and be able to cope with large-scale networks. Furthermore, in systems to which these are applied, user state information stored in servers has to be passed between servers when increasing, decreasing or switching server resources so that user services remain unaffected and a stable service can be provided.

In this article, we describe current state of virtual server technologies, SDN/OpenFlow technologies, and network resources management and control technologies, then discuss issues with application of these to carrier networks. We also describe an overview of an Elastic Core method that stores state information in an external database by taking the information from service control servers, as architecture for applying virtual technologies to core networks. This method increases service availability by simply passing state information between servers when resources are increased, decreased or switched. Furthermore, we describe the results of evaluation carried out on an IMS test bed[*3].

2. Virtual Technologies and Application Issues

2.1 Virtual Server Technologies

In recent years, there have been advances in the commercial application of virtual server technologies such as those used in data centers for cloud services. Virtual server technologies have freed up the binding relationship between applications and hardware so that applications can be distributed flexibly. Thus, these technologies hold the promise of delivering highly efficient resource allocation and effective reduction of operating costs. However, to apply virtual server technologies to carrier networks, new design methods for greater reliability with virtualization and real-time processing capabilities need to be established.

In particular, for CPU and memory in virtual environments and real-time processing capabilities in I/O access, applications for call processing and so forth must meet stringent requirements. Hypervisor[*4] is commonly used to mount and run multiple virtual machines on physical resources, however, the reliability of Hypervisor itself has to be raised to the level required by carrier networks.

As well as that, consideration needs to be given to moving applications in case of server replacement, switching applications with server failures, or load balancing across multiple servers with scale-out[*5], or when user processing is consolidated in a single server with scale-in[*6]. In these cases, state information for the relevant users has to be shared between the original application and the one that was migrated.

In addition, during moving, switching or executing scale-out/scale-in, traffic needs to be properly dispatched with a flow base control.

As described above, the key issues with applying virtual server technologies to carrier networks are (1) achieving real-time processing capabilities that meet the stringent requirements of network applications, (2) passing state information transparently in the cases of moving/switching/duplicating/consolidating applications and (3) properly dispatching traffic coupled to virtual server control.

To deal with the issues of applying virtual server technologies to carrier networks, the Network Functions Virtualisation Industry Specification Group (NFV-ISG)[*7] was established in the European Telecommunications Standards Institute (ETSI)[*8], and discussions have begun.

2.2 SDN/OpenFlow Technologies

In conventional networks, network devices such as routers and switches are configured individually and exchange routing information to enable data transport. Due to this principle, conventional networks operate as distributed autonomous system and have high resilience. On the other hand, conventional networks cannot utilize network bandwidth efficiently with specific configuration based on traffic, since assigning individual routes to traffic and increasing available bandwidth by using multiple routes simultaneously are very difficult to achieve.

SDN is the concept of flexibly controlling a network with software, and

*3 **Test bed**: An apparatus for experimentally evaluating the viability and performance of a technological method.

*4 **Hypervisor**: A virtual server technology that assigns and manages physical resources for applications to mount on virtual machines, and runs multiple virtual machines on physical resources.

*5 **Scale-out**: Adding and assigning new resources to reinforce processing capacity when service requests increase and there is insufficient processing capacity on the network.

*6 **Scale-in**: Releasing some assigned resources to reduce capacity when service requests decrease and there is excess processing capacity on the network.

OpenFlow is one technology that achieves this, by separating the control plane*9 from the data plane*10 and enabling control of traffic as individual flows. **Figure 1** shows an example of network control with OpenFlow. The OpenFlow controller controls the behavior of each OpenFlow switch to enable flexible routing of specific traffic (called a "flow" in OpenFlow). In Fig. 1, the OpenFlow controller assigns different routes to the flows based on their characteristics.

OpenFlow technology can change the behavior of entire networks by pushing control scenarios from OpenFlow controllers. This enables dynamic change of traffic routes in data centers shared by multiple network users, and is promising as a solution to the virtual server technology issue (3) described above. In particular, we believe this can sort processing traffic without affecting target servers or user terminals.

However, OpenFlow has a scalability issue with application to large-scale networks because a centralized OpenFlow controller controls all switches. Furthermore, there is a reliability issue with the OpenFlow controller failover scheme because it is a potential single point of failure.

To solve these problems, the Open Networking Foundation (ONF) [1], a group for standardizing OpenFlow, is studying improvements, while the Open Networking Research Center (ONRC) [2] has been established as a research organization that is engaging in joint research with corporations. NTT DOCOMO is participating in ONRC to tackle the issue of raising SDN/OpenFlow technologies to carrier grade levels.

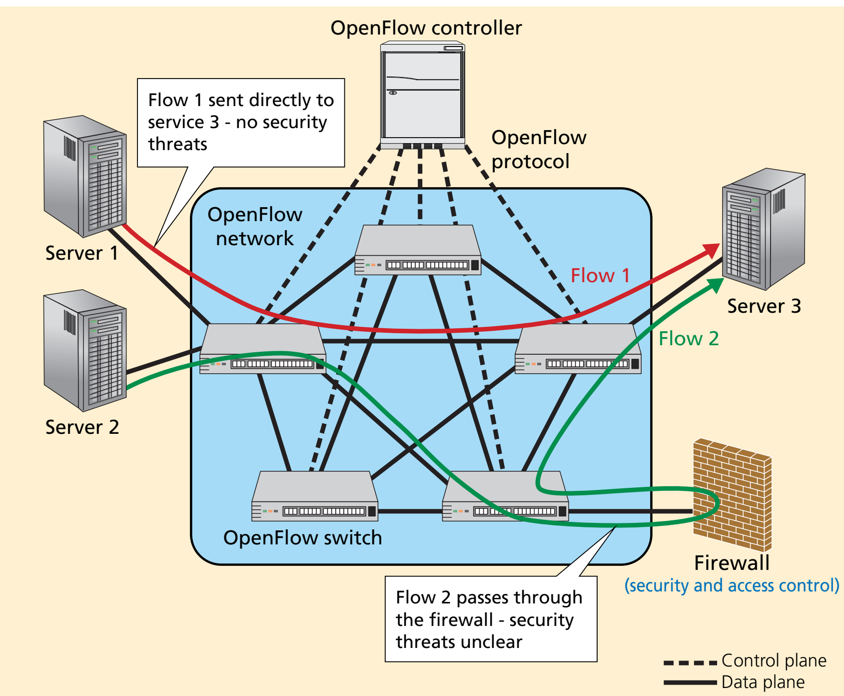

Figure 1 Network control with OpenFlow

2.3 Network Resources Management and Control Technologies

Network resources management and control technologies are elemental technologies for linking application placement and migration control*11 through the virtual server technologies discussed earlier with flow control and providing integrated management to achieve SDN (**Figure 2**). To apply these technologies to carrier networks and provide services using virtual network resources, monitoring of user service quality and monitoring of logical and physical network resources that are in use by that traffic must be available as required. Specifically, this entails monitoring the status of physical resources, understanding the status of virtual logical networks and monitoring and managing the load on resources.

Operation and control of network resources must also comply with the operational policies of the operator. These issues include the establishment of compliance with changes to operational policies in response to congested situations and logic to optimally and dynamically distribute resources, prioritized control of individual services and

*7 **NFV-ISG**: Led by network operators, this group was established in ETSI in November 2012, with the aim of summarizing general outlines and conditions related to virtualization of network functions and introducing them into groups standardizing related technological specifications (ONF and so forth).

*8 **ETSI**: The standardization organization concerned with telecommunications technology in Europe.

*9 **Control plane**: Control processes to transfer data, such as route control of data in use for communications

*10 **Data plane**: Data forwarding processes during communications users in those instances, and adjustment methods based on prioritized control between resources.

It is also assumed that network topology[*12] will change with dynamic control, for example, when a network failure is detected and routing must be changed. In those instances, depending on the type of service or network conditions, it's necessary to select routing in line with operational policies to change the topology, and to equip functions for changing the assignment of resources coupled with the altered topology and provisioning management functions[*13].

Regarding these issues, ONF recognizes the issues with Northbound API[*14] for coupling communications applications with OpenFlow controllers and so forth, and has begun discussions.

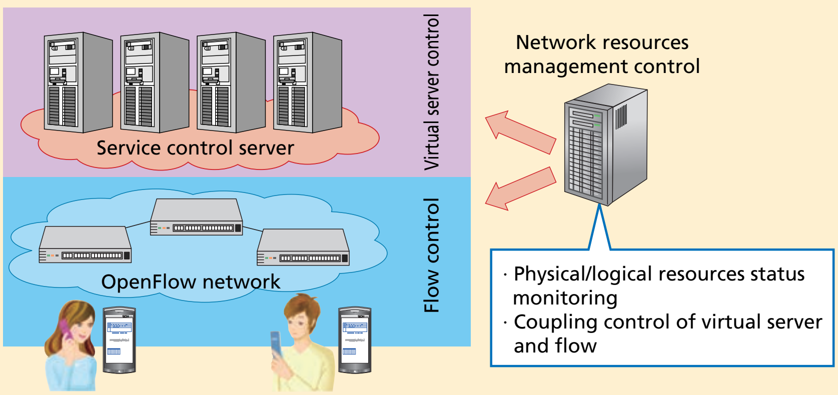

Figure 2 Network resources management and control technologies

3. Elastic Core Architecture

To solve the problems mentioned above and to improve the availability of services, we propose an Elastic Core method. **Figure 3** describes the Elastic Core architecture[3]. This architecture consists of virtual service control servers, databases, an OpenFlow network and an NW manager.

The Elastic Core method solves the server virtualization issue (2) related to passing state information before and after migration/switching/replication/aggregation of communication service applications, by storing state information in an external database and separating state information from call processing. This method makes it simpler to pass user processing states between different servers and thus increase service availability. Furthermore, the Open-

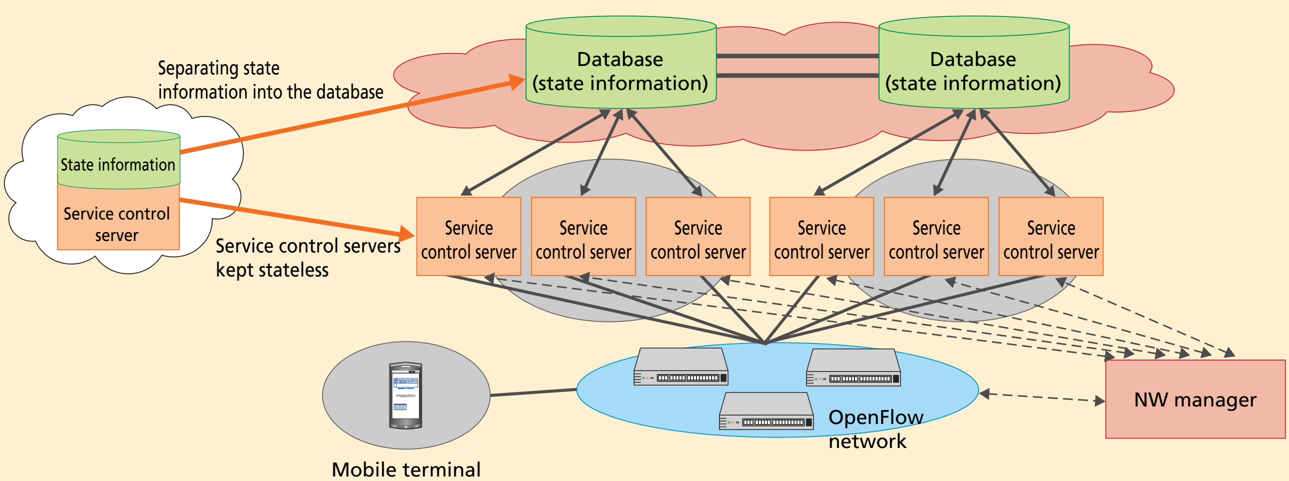

Figure 3 Elastic Core architecture

*11 **Migration control**: Migrate an application mounted on a certain virtual machine to a different virtual machine to reshuffle physical resources in virtual server technologies.

*12 **Topology**: Logical relationship of devices, network configuration, etc.

*13 **Provisioning management functions**: Network resources management control commands logical functions to be assigned to physical resources on the network, and achieves performance status management.

*14 **Northbound API**: An interface for controlling OpenFlow controllers (SDN controllers), etc from network resources management control applications.

Flow network is used as a data transport network to control user flows corresponding to virtual service control servers provisioned dynamically on the core network. The NW Manager achieves integrated network management by controlling the OpenFlow switches and virtual service control servers.

The IP Multimedia Subsystem (IMS)[*15] service control network consists of session control functions (Proxy/Serving/Interrogating-Call/Session Control Function (P/S/I-CSCF)[*16-18]) and user data storing functions (Home Subscriber Server (HSS)[*19]), etc. It processes service requests from mobile terminals (User Equipments (UE)) and provides IP-based multimedia communications services. **Figure 4** describes the registration[*20] sequence for UE location registration with the implementation of the Elastic Core method for S -CSCF in the IMS.

UE sends a register message as a registration signal to its IMS (Fig. 4 (1)). P-CSCF selects I-CSCF and relays the message (Fig. 4 (2)), and I-CSCF authenticates the user with HSS (Fig. 4 (3) (4)), selects S-CSCF to register the UE, and relays the register message to the selected S-CSCF (Fig. 4 (5)). S-CSCF acquires the authentication vectors[*21] needed to authenticate the user from HSS (Fig. 4 (6) (7)), and authenticates the UE's user using those authentication vectors (Fig. 4 (8)-(15)). When S-CSCF successfully authenticates the user, it reads the user registration status from the database (Fig. 4 (16) (17)), and determines whether the UE is already registered. At this time, the registration data that is read is stored in local memory, user information is acquired from HSS (Fig. 4 (18) (19)), and registration data is updated and written into the database (Fig. 4 (20) (21)).

By writing registration data into the database in this way, communications processes can be continued once registration is complete if the S-CSCF managing the UE fails. This is achieved by the OpenFlow network transferring signals from the opposing device to another arbitrary S-CSCF, and the S-CSCF that receives the signals then reading the registration data from the database. This achieves high service availability and improves server resource efficiency by sharing backup servers.

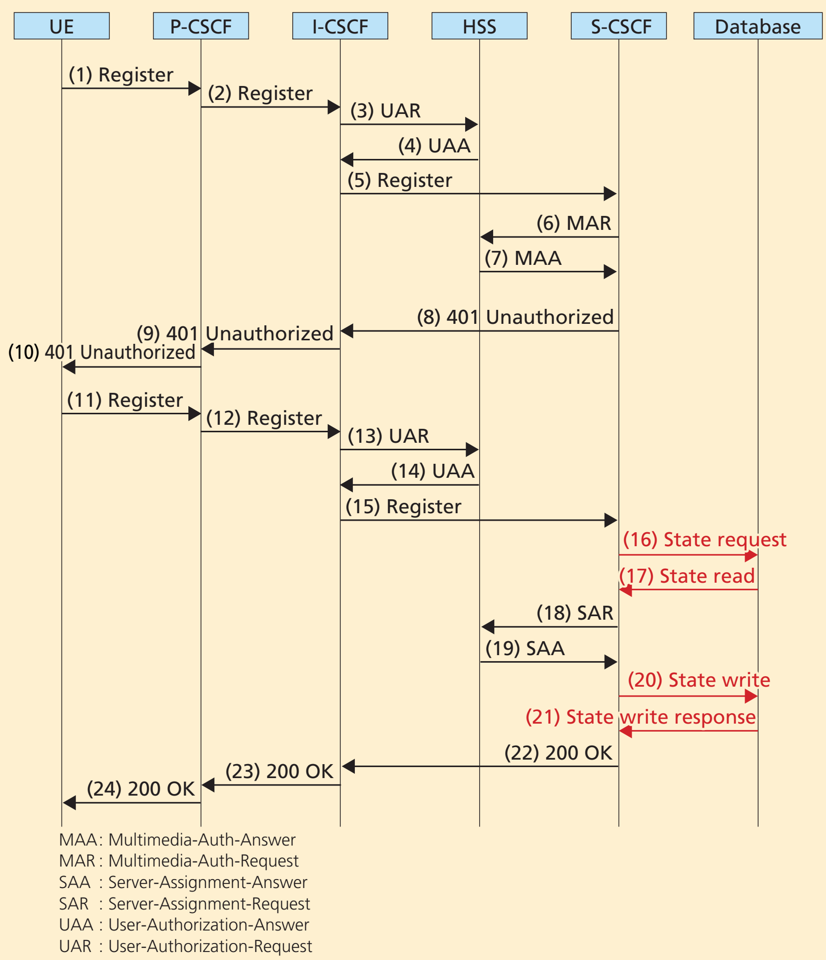

Figure 4 Registration sequence

*15 **IMS**: A 3GPP standardized call control procedure that realizes multimedia communications by consolidating fixed and mobile network communication services with Session Initiation Protocol (SIP), which is a protocol used on the Internet and with Internet phones.

*16 **P-CSCF**: Positioned at the connection points between mobile terminals, S-CSCF (see *17) and I-CSCF (see *18), to relay SIP signals between them.

*17 **S-CSCF**: A SIP server performing terminal session control and user authentication.

*18 **I-CSCF**: A SIP gateway server that is the first connection point from remote networks when interconnecting or roaming. It identifies the S-CSCF and relays messages.

By similar procedures, the calling state in a session set up signaling (Invite[*22]) sequence can also be passed by storing S-CSCF processing status data (dialogue data etc) in the database.

4. Experimental Evaluation of Elastic Core Deployment in S-CSCF

We prototyped Elastic Core deployment in S-CSCF in IMS to evaluate the impact on services.

4.1 Evaluation of the Impact of Switching S-CSCF with Communications Failures

Figure 5 describes the S-CSCF switching scenario with a communications failure. We tested this scenario with the conventional method and the Elastic Core method, and evaluated the effects on services. In this scenario, user group 1 has been registered to S-CSCF1, and user group 2 has been registered to S-CSCF2. User group 1 makes 400 calls per second to user group 2 during the test. At a specified time, we caused a communications failure in S-CSCF1. When the failure occurs, user group 1 is switched from S-CSCF1 to S-CSCF backup. **Figure 6** shows the measured results.

1) Conventional Method

In the initial state, calls from user group 1 to user group 2 are established with processing in S-CSCF1 and S-CSCF2 (dotted line in Fig. 5 (A)). At

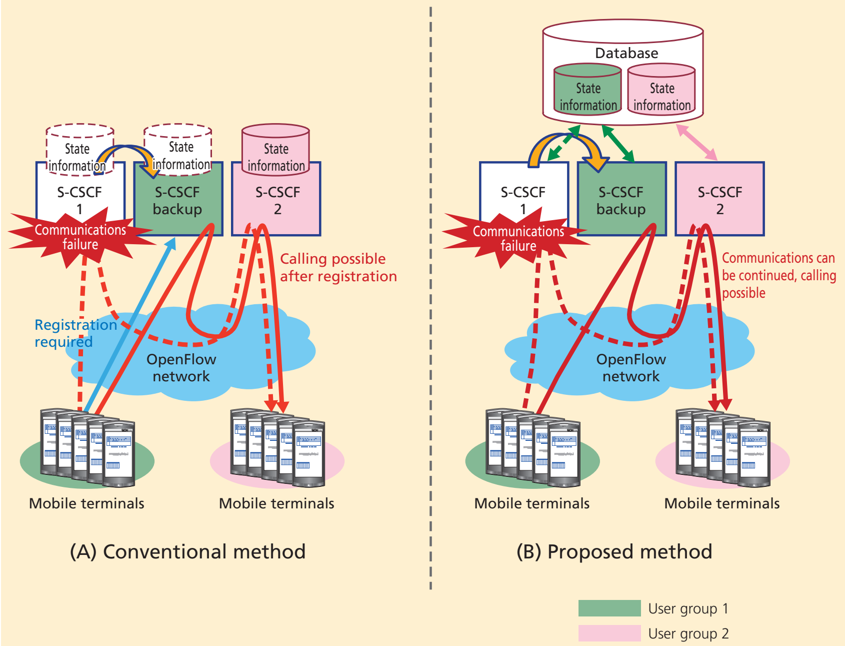

Figure 5 S-CSCF switching scenario with communications failure

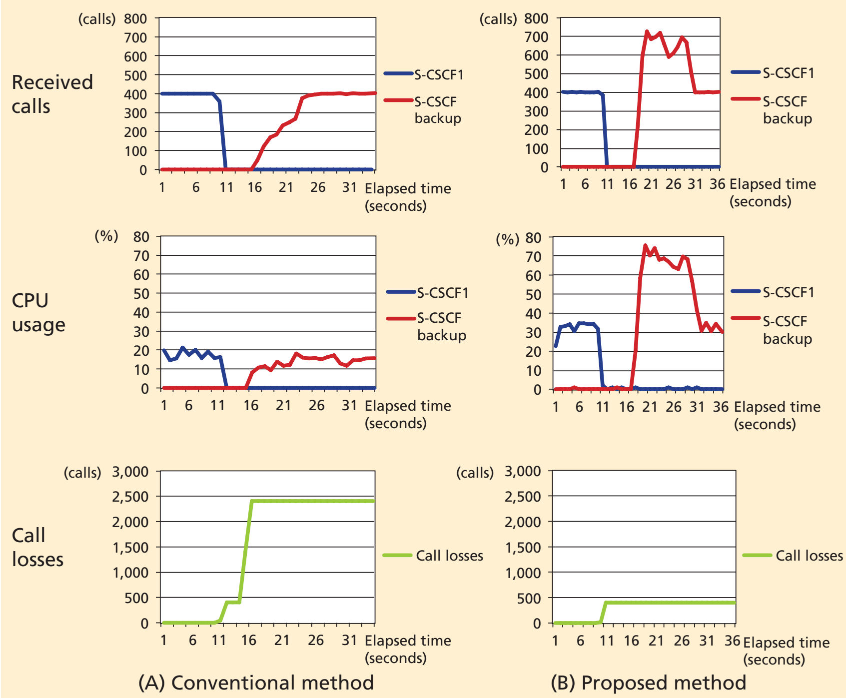

Figure 6 Switching scenario measured results (conventional method, proposed method)

*19 **HSS**: A subscriber information database in a 3GPP mobile network that manages authentication and location information.

*20 **Registration**: In IMS, mobile terminals register current location data in HSS with SIP.

*21 **Authentication vectors**: Data consisting of random numbers, authentication tokens, private keys and integrity keys used to ensure that parties communicating are legitimate between the network and terminals.

*22 **Invite**: A SIP signal requesting a session set up.

this time, user state information (registration state, calling state, etc) is stored in each S-CSCF. When we caused a failure in S-CSCF1, S-CSCF1 is switched to S-CSCF backup as shown by the solid line in Fig. 5 (A). In this instance, because user group 1 state information is stored in S-CSCF1 and the information cannot be passed to S-CSCF backup, it is necessary to start over from the registration process to make calls, which means that call loss builds up with server switching, as shown in the call loss graph in Fig. 6 (A). Users in user group 1 are able to call once registration is redone after switching to the other server, and number of received calls is gradually recovered as shown on the received calls graph in Fig. 6 (A). In this way, when a failure occurs in a service control server that stores user state information, the conventional method requires re-synchronization processing because state no longer matches user terminals. Thus, when the server processing load is high, and this re-synchronization processing occurs concurrently, it can cause processing congestion[*23] in the communications processing node.

2) Elastic Core Method

In contrast to the above, with the Elastic Core method, user state information is successively written and stored in the external database. For this reason, even if a communications failure occurs in S-CSCF1 and switching to S-CSCF backup occurs, S-CSCF backup can read user group 1 state information from the database to continue the service processing that was being performed by S-CSCF1. In the call losses graph in Fig. 6 (B), about 400 calls were also lost with our proposed method. These losses were for calls made prior to writing state information to the database in the server that experienced the failure. Thus, active calling which had been established before the failure or new calls that were made did not experience call loss, nor did re-synchronization due to unmatched states with user terminals occur. After switching servers with the proposed method, CPU usage on the backup server temporarily rises due to rescue processing to continue communications for calls made during switching.

4.2 Evaluation of the Proposed Method with a Scale-out Scenario

As shown in **Figure 7**, with the proposed method, we evaluated the impact on services with aggregating processing in one server (dotted line in Fig. 7) or distributing processing to a new server (S-CSCF b) which is newly provisioned to handle the increase in

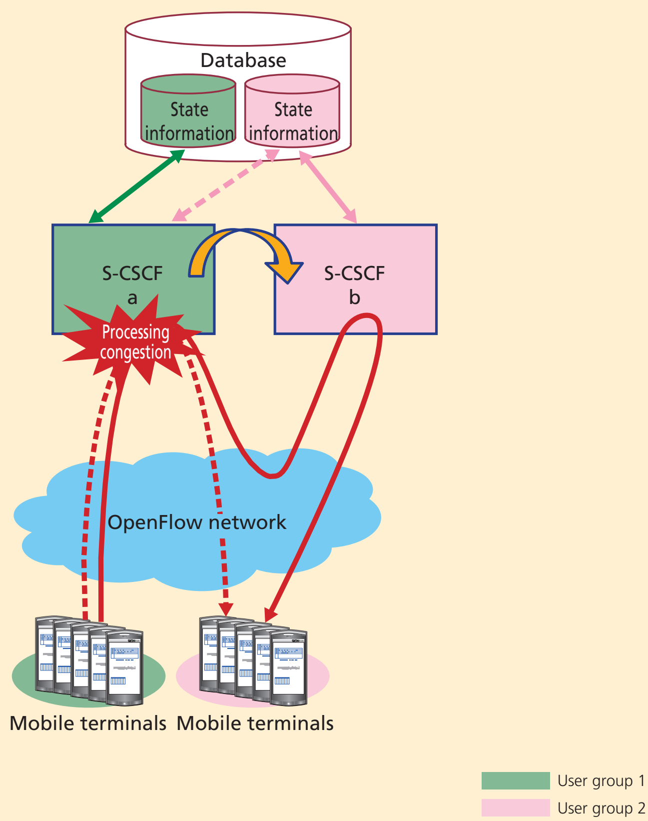

Figure 7 S-CSCF scale-out scenario with processing congestion

*23 **Congestion**: Impediments to communications services due to communications requests being concentrated in a short period of time and exceeding the processing capabilities of the service control server.

call requests (solid line in Fig. 7).

As can be seen in the number of calls received and CPU usage graph in **Figure 8**, service processing stability was achieved by distributing the processing concentrated in S-CSCF a with the addition of S-CSCF b. With this scale-out, similar to the failure switching described previously, service processing can be continued because S-CSCF b acquires state information for relevant users from the external database. Scale-out/in is not the same as failure switching, and because it is possible to sort call transfer to the newly established S-CSCF b in a planned manner, call loss does not occur, as shown in the call loss graph in Fig 8.

5. Conclusion

This article has described trends in virtual server and SDN/OpenFlow technologies in the data center market, and has discussed issues with their application to carrier networks. We have also described an overview of an Elastic Core method in which those technologies are applied, and how state information is separated from servers and stored on an external database to perform user service processing. Applying this method to IMS S-CSCF and prototyping it on a test bed, we have confirmed that it can achieve high availability and continuous service on core networks in cases of server exchange failure and increases and decreases in processing capacity among servers.

Into the future, we intend to study

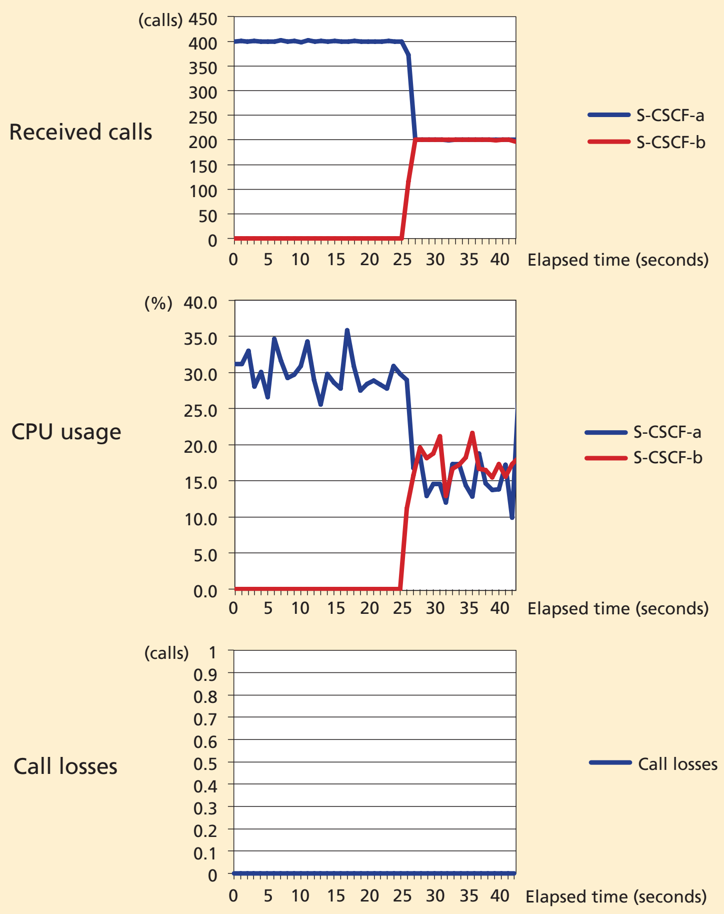

Figure 8 Scale-out scenario measured results (proposed method)

expanded application of these technologies to other core network nodes, and further clarify network resources control optimized for network conditions.

References

[1] Open Networking Foundation homepage. https://www.opennetworking.org/

[2] ONRC homepage. http://onrc.net/

[3] Bumpei Yatsu, Takahiro Yamazaki, Tetsuya Nakamura, Motoshi Tamura, Shigeru Iwashina and Takashi Shimizu: "An Approach for Using Network Virtualization Technology in Mobile Communication Networks," Proc. of the IEICE General Conference 2012, Communications (1), pp. 581, Mar. 2012 (in Japanese).