Year 2014

Dissertation

Signaling-based Dependable Services on the Internet

Takeshi Usui

Advisor: Professor Jun Murai

平成26年度

博士論文

インターネット上でシグナリングプロトコルを活用した高信頼サービスの実現に関する研究

臼井 健

指導教員： 村井 純 教授

Dissertation Overview Year 2014

Signaling-based Dependable Services on the Internet

Signaling protocol begins to be used to perform billing and authentication to devices, and to exchange the parameters for QoS setting, before the devices start the communication on the global Internet. This dissertation proposed an advanced architecture to improve the availability of the services using the signaling protocol. By utilizing the proposed architecture, the communication restriction on the earthquake disaster can be diminished. In addition, the proposed architecture makes it possible to utilize the signaling protocol in the services using the large number of devices, e.g., Internet of Things (IoT) services, which is expected to be breakthrough in the future.

The issues regarding the architecture and operational methods for providing the services by using the signaling protocol have became apparent and never been solved in the even single network domain. There are architectural issues where the signaling protocol cannot prevent the communication of users from being disrupted, when the volume of the traffic increases beyond the bandwidth of the routes and the capacity of the servers. Because the current architecture statically uses the only shortest path and the servers registered beforehand, the routes having the available bandwidth and the servers having the available capacity are not used for the communication.

On the other hand, there is an operational issue where it becomes difficult for the operators to locate where the signaling messages are lost, because the signaling messages are exchanged among the multiple servers at the multiple times. The network operators need to prepare the servers for the redundancy so as to store the active servers so as to maintain the quality. Many messages are exchanged in the signaling protocol. Moreover every time the messages are exchanged, they need to store the active servers. Then, there is another issue where the operational load for the servers in the redundancy becomes high, because many servers are required. Therefore, the operational load for the signaling protocol is currently high.

This dissertation proposed the advanced architecture to utilize the route having available bandwidth so as to improve the availability of the services, and to make the devices connected into the servers having the available capacity so as to improve the availability of the signaling protocol. This makes it possible to provide the dependable services which can response to the unexpected communication request by flexibly utilizing the resources of the routes and servers in the whole network.

This dissertation also proposed the operational methods that locate the lossy links of signaling messages from the limited information and reduce the number of the servers used for the redundancy without degrading the service quality by selective storing. The proposed methods reduce the time for the trouble shooting of signaling protocol and streamline the operational method by reducing the number of servers for the redundancy.

The proposals described above were evaluated based on the performance requirement of the single network domain. The further study regarding the function to adjust the policy among

the multiple network domains is required to provide the services using the signaling on the global Internet. However, the proposed architecture and operational methods evaluated in this dissertation can be easily extended on the global Internet, because the global Internet consists of the multiple network domains and each network domain manages the services by itself. As the result, this dissertation becomes a large foundation to improve the availability of the services using the signaling protocol on the global Internet.

Keywords: Network operation, Network architecture, Signaling protocol, Internet

博士論文概要 2014 年度（平成 26 年度）

インターネット上でシグナリングプロトコルを活用した高信頼サービスの実現に関する研究

インターネット上のサービスで、実際のデータ転送の前に、デバイスの認証や課金、QoS のパラメータの交換などをするためにシグナリングプロトコルの利用が増えている。本論文では、これらシグナリングプロトコルを用いたサービスの可用性を向上するためのアーキテクチャを提案した。本アーキテクチャを用いることにより、震災時に発生する通信規制の緩和や、今後発展が予測されている Internet of Things (IoT) サービスなどデバイスの数が飛躍的に増加するサービスの通信制御にもシグナリングプロトコルを用いることが可能になる。

シグナリングプロトコルを用いサービス提供するためのアーキテクチャや運用手法について、一つのネットワークドメインにおいてさえ現在までに多数の課題が明らかになり解決されていない。まずアーキテクチャ面の課題として、サービスに対して予測できない通信が発生しても、必ず最短経路のみが利用され、また事前に登録処理したシグナリング処理サーバしか利用されず、網内の他の経路やサーバでリソースに余力があってもユーザの通信に活用されないことがある。そのため、単一の経路の帯域、または単一のサーバの処理容量を超えるトラフィックが到達した場合、現在のアーキテクチャではユーザの通信は停滞または停止する。

次に運用面の課題として、複数のサーバで複数回メッセージ交換されるため、シグナリングプロトコルを監視しメッセージロス発生箇所を短時間での特定が困難であることがある。さらに、サービスの品質維持のため冗長方式は必須であるが、シグナリングプロトコルでは多数のメッセージ交換が行われその度に冗長のため同期が必要である。この際、冗長方式を取るために多数のサーバが必要になりサーバ運用の負荷が上がる課題もある。これらのため、シグナリングプロトコルの運用負荷は現状では大きい。

本論文では、アーキテクチャの課題を解決するために、まずサービスの可用性向上を目的にシグナリングプロトコルにより空き帯域がある経路を活用するアーキテクチャ、さらにシグナリングプロトコル自体の可用性向上のためリソースに空きがあるサーバへユーザを接続切り替えさせるアーキテクチャを提案した。経路やサーバを柔軟にネットワーク全体で利用可能にするアーキテクチャにより、突発的な通信要求に応える高信頼サービスを提供可能にした。

本論文では運用面の課題を解決するため、シグナリングメッセージの再送数という限定的な情報からロス発生個所を短時間で特定する方式と、ユーザの通信品質を落とすことなく同期個所を選択し冗長のために用意するサーバ数を削減する方式を提案した。提案方式により、トラブルシューティングに要する時間を短縮し、冗長のために用意するサーバ数を削減することで運用を効率化した。

本論文の提案の評価は、一つのネットワークドメインでの要求性能に対して行った。グローバルなインターネットでのシグナリングプロトコルの利用は、その規模性やそれぞれのネットワークの運用ポリシの調整機能などの検討が必要である。しかし、グローバルなインターネットは複

数のネットワークドメインで形成されそれぞれ独自で管理されることから、本論文で提案、評価したアーキテクチャを応用可能である。結果として本研究はグローバルなインターネットにおけるシグナリングプロトコルを用いたサービスの可用性の向上の大きな礎になる。

キーワード： ネットワーク運用、ネットワークアーキテクチャ、シグナリングプロトコル、インターネット

Acknowledgment

I would like to express my appreciation to many individuals toward the submission of this dissertation. First and foremost I would like to extend my appreciation and gratitude to my supervisor Professor Jun Murai for the opportunity to pursue this dissertation under his supervision. I would like to thank my co-advisers Professor Hideyuki Tokuda for this opportunity as well as his precise advice and guidance, Professor Osamu Nakamura for his time and continuous guidance since my undergraduate years in Keio University and Executive Director of KDDI, Dr. Masahi Usami for his time, precise guidance.

I would like to deeply thank many individuals of KDDI Corportation and KDDI R&D Laboratories Inc. In particular, I would like to express my appreciation to Dr. Yoshinori Kitatsuji. The result of this dissertattion could not have been achieved without his supervison and guidance for the past several years. I am deeply thankful to Chairman of the Board of Directors, Fumio Watanabe, President & CEO, Yasuyuki Nakajima, Executive Vice President,Masatoshi Suzuki, Executive Directors, Shinichi Nomoto and Itsuro Morita for their kind support and guidance. I would like to thank many present and former members of the Mobile Network Laboratory, for their generous discussions,fruitful comments and feedback. I also would like to thank to Dr. Kenji Kumaki at KDDI Corportation. I was motivated to tackle these research with his powerful action, when I worked in KDDI headquater. I also would like to thank many present and former members of the New Generation Network Laboratory at National Institute of Information and Communications Technology (NICT,in particular Professor Masayoshi Murata, Dr. Nozomu Nishinaga, Dr. Yozo Shoji and Dr. Kiyohide Nakauchi)

I would also like to thank many present and former members of the Internet Research Laboratory at Keio University. In particular, I would like to thank Shunsuke Fujieda, Kotaro Kataoka and Achmad Husni Thamrin for their guidance during my studies at Keio University. This dissertation would not have been possible without the experience gained through many years of studies with them. I would like to thank Professor Keiko Okawa, Assistant Professor Keisuke Uehara, the administrative staff of Professor Murai, including Yasue Watanabe, Rika Aoki,HiromiHiga, Yukie Shibuya and the secretary of Professor Tokuda, YukaMatsuo for their generous support and arrangements.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

More than 40 years have seen the development of various commercial and free-of-charge communication services over the Internet. The Internet is now became a fundamental position for the communication services. Today, the Internet serves is the global infrastructure for education, commercial and social activities. The Internet still improves its ability to admit Internet of Things (IoT) devices. Cisco Internet Business Solutions Group predicts there will be 25 billion devices connected to the Internet by 2015 and 50 billion by 2020 [1].

The increase of IoT devices will change the traffic pattern in the Internet. Now, the numbers of simultaneous communication among the users are limited, and the traffic volume on the peak periods becomes four times more than on the off periods. However, the IoT devices will try to communicate simultaneously. The architecture needs to be considered not only the number of connected devices but also the simultaneous communication, in order to provide the dependable services.

In Great East Japan Earthquake, the traffic of the signaling protocol was 50 times more than the off periods concentrated [2]. However, the existing architecture could not treat such traffic and rejected them. As the result, many people could not communicate for long periods. Because the communication pattern of the IoT devices generates such concentration, the architecture is required in order to treat the traffic without rejecting.

Most communication services over the internet utilize signaling protocols, in order to control and manage the services such as Session Initiation Protocol (SIP) [3]. IP Multimedia Subsystem (IMS) [4] employing SIP control and manage the Telecom applications, such as telephone and video services. Hereafter, Hereafter, the 'signaling' is referred to as the 'procedure of a signaling protocol'. And, the services using the signaling protocol is termed as "signaling-based services".

The typical achievements of controlling and managing a service are a registration of the location of devices, an authentication, an accounting, a searching for the corresponding nodes, an exchanges of the parameters, the negotiation of the network resources and so on.

The availability and reliability of a service strongly depend on how a procedure of its signaling protocol executes smoothly. In this case, the messages of signaling messages are not seamlessly exchanged. Its service is disturbed or, in the worst case collapsed. This kind of failure easily occurs in the services executed over the Internet.typically due to the message loss because of resource shortages in networks and/or servers dedicated for the signaling. A service provider usually provider usually retains network and server resources in a redundant manner.

The current study focuses only on creating the signaling protocol made of IP technology, which have the same function with the ones on the circuit switched network. The issues regarding the architecture and operational methods in the even single network domain to provide the services by using the signaling protocol have became apparent and never been solved. Now, for providing the signaling-based dependable services on the internet, the service operators need to

solve the issues regarding the architecture and the network operation.

The issues regarding the architecture and operational methods in the even single network domain to provide the services by using the signaling protocol have became apparent and never been solved. There are architectural issues where the signaling protocol cannot prevent the communication of users from being disrupted, when the volume of the traffic increases beyond the bandwidth of the routes and the capacity of the servers. Because the current architecture uses the only shortest path and the servers registered beforehand, the routes having the available bandwidth and the servers having the available capacity are not used for the communication.

On the other hand, there is an operational issue where it becomes difficult for the operators to locate where the signaling messages are lost, because the signaling messages are exchanged among the multiple servers at the multiple times. The network operators need to prepare the redundancy to backup the active servers so as to maintain the quality. Then, there is another operational issue where the load of the network operation becomes high, when the servers for the redundancy increase in number.

1.1 Motivation

The work in this dissertation is motivated to improve the availability of the services using the signaling protocol on the global Internet. This dissertation solves these issues from two aspects: operational methods, and implementation architecture. The operational method is ways to retain session states which is composed of service control and management updated when signaling messages are processed. Usually, the way to retain the session states are achieved by backup and restoration of the session states with redundant servers. When a role back of the session states occurs in restoring the session states from the failed server to the backup server, the signaling is collapsed. The implementation architecture is a way to prevent overload in signaling servers, and to smooth signaling message delivery in congested networks. Hereafter, the service execute with the operational methods and implementation architecture is referred to as 'signal-based dependable service'.

Actually, the signaling messages and protocol will be upgraded along with the evolution of the applications and services on the Internet,because some new parameters are required for the application and services. It is very meaningless to consider the operational methods and network architecture, every time the new signaling messages and protocol are generated. Therefore, I consider the operational methods and network architecture based on the basic feature feature feature feature of all the signaling messages and protocol.

1.2 Goal

The goal of this dissertation is to provide the dependable signaling-based services on the global. For this goal, this dissertation solves the issues regarding the architecture and operational methods. This dissertation proposes the architecture to utilize the route having available bandwidth so as to improve the availability of the services, and to make the devices connected into the servers having the available capacity so as to improve the availability of the signaling protocol.

This dissertation also proposes the operational methods that locate the lossy links of signaling messages from the limited information and reduce the number of the servers used for the redundancy without degrading the service quality. The proposed methods reduce the time for the trouble shooting of signaling protocol and streamline the method for the redundancy by reducing the number of servers required for the redundancy.

Actually, the signaling messages will be upgraded along with the evolution of the applications and services on the Internet,because some new parameters are required for the application and services. However, the basic functions of the signaling messages will not change. Therefore, the dissertation also aims to provide the operational methods and network architecture which can be easily extended for the various applications and services in the various ways in the future.

Design of Operational Methods

To solve the operational issues, this dissertation shows the design of the method to locate the lossy links of the signaling messages from the limited information. If all the log of the signaling messages are checked to detect the loss, a large amount of resources are necessary.

In addition, this dissertation shows the design of the method to restore the servers processing the signaling messages by the limited number of the times to store the session states. Every time the session states are stored for restoring, a large number of the backup servers are required to treat the process to store the session states.

The analysis shows that the designed methods satisfies the dependability by detecting the problems related to the signaling process and restoring the servers within the short duration that do not influence the quality of communication.

Design of Network Architectures

To solve the architectural issues, this dissertation shows the design of the solutions that consists of transferring the traffic into the route having surplus bandwidth and connecting the users to the server having surplus capacity. This solutions ensure the dependable services enabling users to communicate continuously, even if the scale of the services becomes larger and the unpredictable volume of the traffic is concentrated. This dissertation shows that the proposed methods satisfy the quality required by the dependable services with flexible selection of the routes and servers in the whole Internet.

1.3 Outline of this Dissertation

This dissertation is divided into seven chapters. The first and the last are the overview and conclusions, respectively. Chapter 2 introduces the overview of the signaling-based services

by showing the actual examples. As the operational method detecting the failure, Chapter 3 proposes locating the lossy links of signaling messages from the limited information. As the operational method storing the session states, Chapters 4 also proposes the selective storing of session states in order to restore the servers from the limited number of storing times. As the network architecture realizing the traffic engineering, Chapter 5 proposes the traffic management by utilizing the signaling messages in MPLS networks. As the network architecture realizing the server access control, Chapter 6 also proposes the session state migration enabling the session states to move into the arbitrary servers. Lastly, Chapter 7 concludes this dissertation and presents the future works.

Chapter 2

Signaling-based Services

This chapter describes the approach of this dissertation, and how the signaling protocol are currently used in the services by showing the actual examples. In addition, this chapter describes the structure of this study proposing the operational methods and network architecture for the dependable signaling-based services.

2.1 Signaling Protocol on the Internet

On the circuit switched network, the signaling protocol are used only for the initiation of the application, the authentication and the accounting. Moreover, on the Internet, the signaling messages are used for the registration of the location of devices, the search for the corresponding nodes, the exchanges of the parameter for initiating the application, and the negotiation of the network resources. Not only the people, but also the things i.e., IoT devices utilize the signaling protocol for their communication. To provide the dependable services on the Internet, how to utilize and manage the signaling protocol are technical challenges.

In this dissertation, the advanced architecture is proposed to improve the availability of the services using the signaling protocol. However, this dissertation does not make the signaling protocol from the scratch, and make the advanced architecture which runs on the base of the existing signaling protocol. This is because we consider that the interoperability is required for the services having been using the signaling protocol.

This dissertation assumes that there is the relationship among the signaling protocol, applications,and transport technology as drawn in Fig. 2.1. Here, the network is divided into three layers (service, service management, and transport technology layers) so as to provide the signaling-based services. The signaling protocol exists between the applications and transport technology. The signaling protocol manages the session states in the telecom and Internet applications and interacts with the IP transport.

2.2 Overview of Existing Signaling Protocol

This section describes how the existing signaling protocol are used in the existing signaling-based services. In the future, the different kind of the signaling-based services could be generated on the Internet and then the signaling messages and protocol are updated for the services. The proposed operational methods and network architecture as described after Chapter 3 could be easily utilized, because they are designed based on the common feature of the signaling protocol.

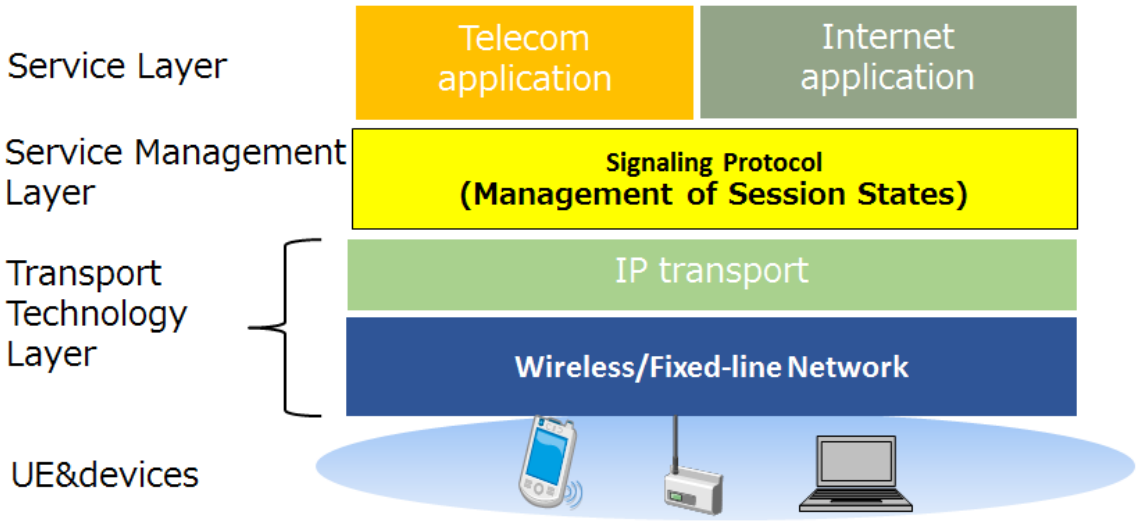


Figure 2.1: Three layers for signaling-based services

2.2.1 SIP

Session initiation Protocol (SIP) [3] is a popular signaling protocol, widely used for controlling multimedia communication sessions such as voice and video calls over IP networks. The protocol defines the messages that are sent between user equipment which establishes, and terminates a call. SIP can be used for creating, modifying and terminating sessions consisting of one or several media streams. SIP can be used for two-party (unicast) or multiparty (multicast) sessions. Other SIP applications include video conferencing, streaming multimedia distribution, instant messaging, presence information, file transfer, fax over IP and online games.

It is an application layer protocol designed to be independent of the underlying transport layer; it can run on Transmission Control Protocol (TCP), User Datagram Protocol (UDP) or Stream Control Transmission Protocol (SCTP). It is a text-based protocol, incorporating many elements of the Hypertext Transfer Protocol (HTTP) and the Simple Mail Transfer Protocol (SMTP). SIP works in conjunction with several other application layer protocols that identify and carry the session media. Media identification and negotiation is achieved with the Session Description Protocol (SDP)[5].

2.2.2 IMS

The IP Multimedia Subsystem (IMS)[4] is standardized in 3GPP[6] for providing various service services over IP-based networks, e.g., VoIP, instant messaging, and video conferencing. In the IMS, the services received at the user equipment (UE) are managed through multiple call/session control function servers (CSCFs) , that is, proxy servers for SIP.

IMS is utilized to realize application- level session control through the following main functional entities [4]. The IMS client is the session control endpoint, and participates in session setup and management via SIP extensions specified by the Internet Engineering Task Force and 3GPP IMS-related standards.

Proxy-/interrogating-/serving-call session control functions (P-/I-/S-CSCF) are the core entities of IMS. They realize several main functions, including localization, routing out/ingoing SIP messages, associating an IMS client with its S-CSCF (as indicated within the client profile).

2.2.3 IMS for IoT

Utilizing the IMS for the application-level session control of the IoT is discussed in [6, 7] The application server (AS) allows the introduction of new IMS-based services. For instance, IMS enables IoT servers to be realized as specific ASs, and IoT devices hosting an IMS client can be controlled by and participate in IMS dialogs. S-CSCF modifies the routing of specific types of SIP messages to ASs depending on filters/triggers specified by client profiles (IMS filter criteria) maintained by the HSS.

The service providers or the 3rd parties execute the authentication and billing through the IMS. In addition, they can manage the traffic of IoT devices through the IMS. Because the API for IMS client has been developed, the operators of IoT devices easily utilize the API and access the services provided by the service providers.

3GPP has also standardized some common IMS services such as the Presence Service (PS) that, following a publish/ subscribe model, allows UEs and hardware/software components to publish data to interested entities previously subscribed to the IMS PS server, defined as presentities and watchers, respectively. Further details about the IoT and IMS were discussed at [8, 9, 10].

2.3 Structure of this Study

I briefly summarize the structure of this study in this dissertation before describing each of the contribution in more details. The overview structure of this study is shown in Fig.2.3. Fig. 2.3 means that the function under the arrow supports the apex of the arrow.

To realize the operational methods and network architecture, two functions, respectively are studied. Finally, the four functions support the dependable signaling-based services.Each function is designed to treat much more number of users and IoT devices than on the current Internet.As below, the summary is explained.

Location of Lossy Links of Signaling Messages

As the method to locate the lossy links of the signaling messages, this dissertation proposes locating the lossy links of the signaling messages from the number of the retransmission. This method focuses on the SIP as the signaling protocol. Because the algorithm for locating the lossy links is based on the basic retransmission pattern of the signaling messages, the proposed method can be easily applied for the new signaling protocol. The proposed method contributes to the reduction of the time for troubleshooting.

Efficient Backup of Session States

As the efficient method to backup the session states, this dissertation proposes selective storing of the session states. This method also focuses on the backup of the CSCFs in the IMS-based

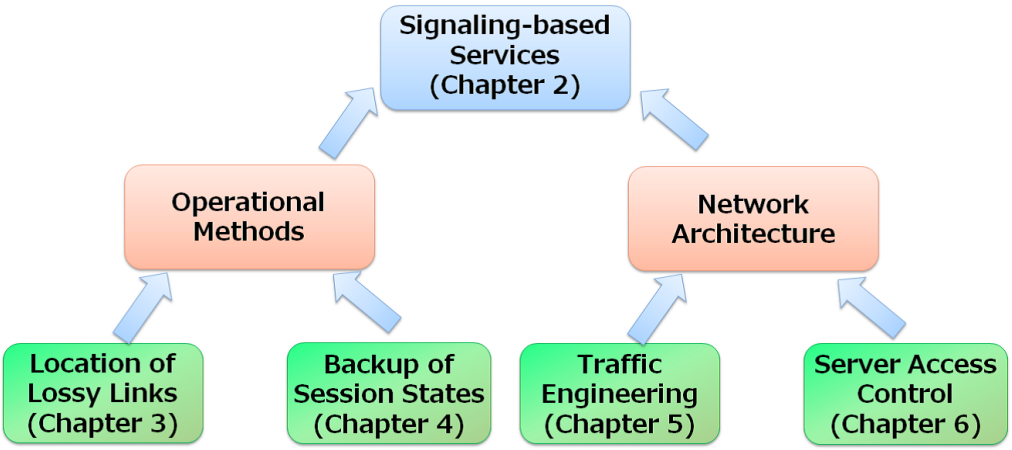


Figure 2.2: Structure of this Study

services. Because the efficient algorithm to decide the points where the session states are stored is based on the basic retransmission pattern of the signaling messages, the proposed method can be easily applied for the new signaling protocol. The proposed method contributes to the streamlines of the network operation.

Traffic Engineering by Utilizing Signaling Protocol

As the network architecture to realize the traffic engineering, this dissertation proposes the traffic management cooperating with the IMS in the MPLS networks. The procedures in the architecture focuses on the IMS as the signaling protocol. Howerver, the procedures and algorithm for selecting the paths in MPLS networks do not utilize the particular function of the SIP. Therefore,even if the protocol is updated or changed, they can be utilized as it was. The proposed method contributes to the improvement of the congestion tolerance regarding the traffic in the network.

Server Access Control utilizing Session State Migration Architecture

As the network architecture to realize the server access control regarding the server treating the signaling messages, this dissertation proposes session state migration architecture. This architecture does not focus on the particular protocol. Therefore, the proposed architecture easily can be applied for the server treating the new signaling protocol. The proposed method contributes to the improvement of the congestion tolerance regarding the signaling messages.

Chapter 3

Proposal of Locating Lossy Links of Signaling Messages

3.1 Background

This dissertation proposes locating the lossy links of the signaling messages from the number of the retransmission. This method focuses on the SIP as the signaling protocol. Because the algorithm for locating the lossy links is based on the basic retransmission pattern of the signaling messages, the proposed method can be easily applied for the new signaling protocol.

Circuit switched public telephone services are gradually being replaced by VoIP services. Many service operators adopt the SIP [3] as SIP becomes a de facto standard for the session controls of VoIP services. SIP allows NPSs to control services and to collect information relating to charging for the usage of their customer communications.

When user terminals having the SIP-based services, such as IP telephony start to communicate with other user terminals, a series of SIP messages are exchanged through the SIP proxy servers between the user terminals. Now, a user terminal is termed "users", hereinafter. The exchange of SIP messages is followed by direct exchange of media traffic between the users. In this dissertation, we refer to a series of SIP messages exchanged in a SIP session as a SIP signaling call flow.

The loss of SIP messages defers the beginning of the media traffic, although SIP has a retransmission mechanism that eventually completes the session establishment. The loss indicates that there are network faults (e.g., on links, in network equipment or at SIP proxy servers). In the worst case, the SIP-based services may result in going down since many SIP signaling call flows are not completed. In particular, we consider that a SIP signaling call flow is of great importance since they are essential for starting/finishing communication (i.e., media traffic) and managing information relating to charging.

The goal of the proposed method is to quickly locate where SIP messages are lost in a light weight manner. we regard a path from one SIP node (a user or a SIP proxy server) to another SIP node as a logical link. The logical link where SIP messages are lost is referred to as a lossy link in the rest of this dissertation. In the case that the SIP proxy server uses UDP for the transport protocol, which many service operators adopt, the retransmission mechanism defined in SIP will be used. In the SIP's retransmission mechanism, there exists the relationship between the lossy link and the number of retransmitted SIP messages. Based on this relationship, we propose a method to locate the lossy links. The proposed method uses only the number of retransmitted SIP messages, which can be collected from Management Information Base (MIB) [11][12] often implemented in SIP proxy servers. We need neither collect all the raw SIP messages from the

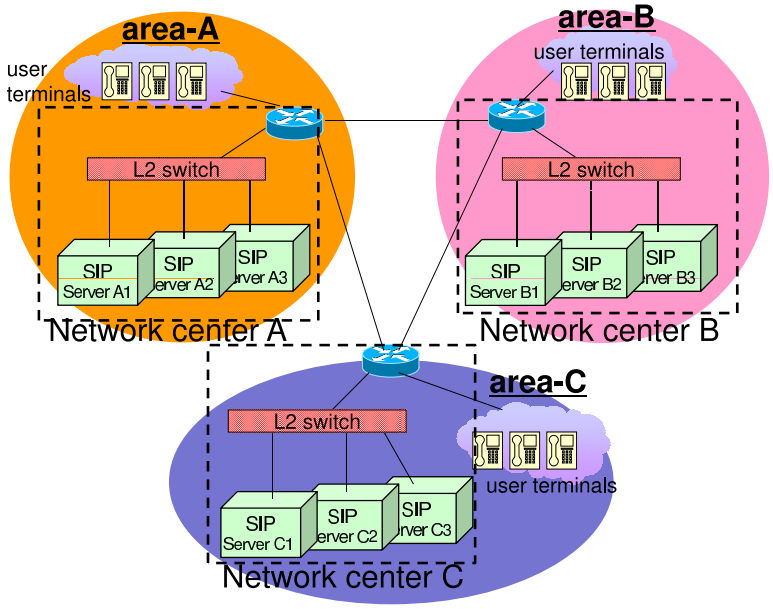


Figure 3.1: Network architecture of SIP-based services

SIP proxy servers nor place extra measurement equipment on the network to locate where SIP messages are lost.

Locating the lossy links provides an important hint for conducting a solution to recover network faults. In general, a network operator performs the following typical network operation: the network operator checks the warning logs or events (they may result from network faults), and then, if there are some problems, the location where the network faults occur is identified to conduct the solution for the problems. SIP signaling call flows suffers from the loss of SIP messages, when the network fault occurs somewhere on a lossy link. Even though the lossy link includes several network segments (i.e., local area networks among neighboring routers) locating the lossy link narrows down the candidates of the network equipment to those on the lossy link.

3.2 Monitoring SIP-based Services

3.2.1 Network Architecture of SIP-based Services

We assume that a service operator accommodating a huge number of customers (e.g., ten to a hundred millions) provides SIP-based services with a large number of SIP proxy servers. In the service operator, there are many network centers which accommodate the SIP proxy servers. We also assume that network centers where SIP servers and network equipment are collocated are geographically decentralized, e.g., in Tokyo, Osaka, Fukuoka, and Sapporo.

In such architecture, users register with a SIP server that is at the network center nearby the users. We refer to a geographical area where users register with SIP servers in the same network center as an area. This can localize the exchanges of the SIP messages and complete the SIP signaling call flows quickly [13] when users in the same area communicate with each other using the SIP based service.

Fig. 3.1 depicts the network of the SIP-based services. There are three network centers (in area A, B and C). In each network center, there are some SIP proxy servers and users which

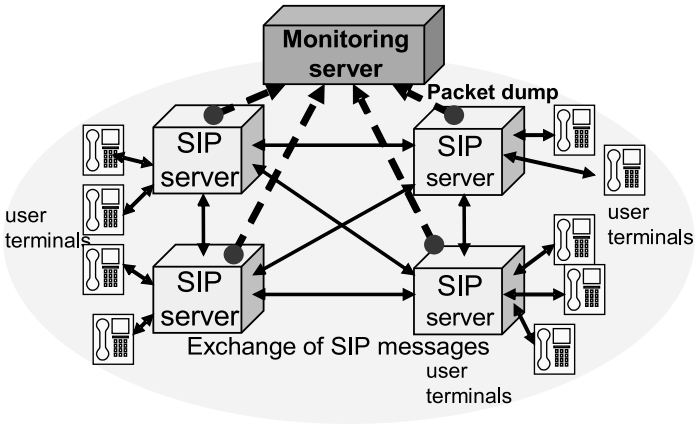


Figure 3.2: Overview of the System Collecting All the SIP messages

register with the SIP proxy servers in the same area.

3.2.2 Issue of Monitoring for SIP-based Services

Many practical tools and methods (e.g., [14][15][16]) to locate where network faults occurred have been studied. Their objective focuses on grasping where some network faults (e.g., packet loss, and long transmission delay) occurred in their networks. These methods ascertain the communication quality by measuring the one-way delays of the probe packets sent from multiple end-hosts in a full mesh manner. Fr

If service operator adopts these methods for investigating the SIP message loss, a lot of measurement equipment needs to be introduced in the network. This is considered to require much cost in a large scale carrier network. Additionally, a lot of probe packets may degrade the quality of the customer communication.

With regard to monitoring SIP signaling call flows, there are the systems [17][18] collecting all the SIP messages as depicted in Fig. 3.2.2. An agent running at each SIP proxy server obtains the packet dump of all the SIP messages. A monitoring server collects the dumped packets from all the agents and reconstructs the SIP signaling call flow. These systems require more computational resources (e.g., computation power and large amount of memory) as the number of the users increases. These systems are convenient for analyzing the contents of the SIP messages (e.g., SIP header and SDP [5]) ,but not suitable for locating the lossy link.

When SIP messages are lost, the SIP messages are retransmitted. But, detecting the retransmitted SIP messages does not always lead to locating the lossy links. This is because multiple SIP proxy servers and users retransmit the related SIP messages, as described in Sections 3.3.3 and 3.3.4. To locate the lossy links, it is necessary to find out the logical links which caused the retransmission of SIP messages from SIP proxy server or user. The operators need to grasp the retransmission mechanism to locate the lossy links.

The proposed method locates the lossy links with the retransmitted SIP messages by understanding the details of the retransmission mechanism. The proposed method complements the function which the systems [17][18] lack. It is possible to use the tool which has the function of the proposed method in an independent monitoring tool or to combine use of the tool which has the function of proposed method and the systems [17][18].

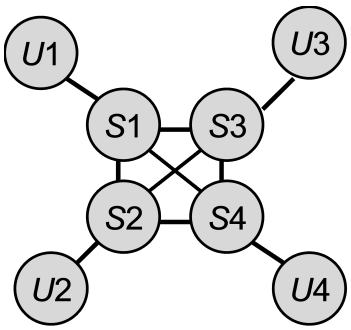


Figure 3.3: Graph for logical connections of SIP nodes

3.3 Proposed Method for Locating Lossy Links

This section presents the method for locating the lossy links through the relationships based on the SIP's retransmission mechanism.

3.3.1 Overview of the Proposed Method

The system which adopts the proposed method to locate the lossy links takes the following two steps:

1. The monitoring server collects the number of retransmitted SIP messages from all the SIP proxy servers periodically (e.g., using MIB as described in section 3.4), and

2. each time after collecting, the monitoring server picks up a SIP proxy server one by one, and search for the lossy links related to the SIP proxy server based on the relationships described in Section 3.3.3 and 3.3.4

In fact, network faults cause the losses of SIP messages not only on single logical link but also on multiple links concurrently. The proposed method can locate multiple lossy links concurrently. In the following section, we introduce a basic method to locate single lossy link, and then, introduce a method to locate the multiple lossy links.

3.3.2 Definitions for the Proposed Method

Fig. 3.3 shows a graph that depicts the logical links among the SIP nodes defined in [3], (i.e., SIP proxy servers and users). In Fig. 3.3, "S" represents a SIP proxy server, and "U" represents a group of users that register themselves with the same SIP proxy server to utilize the SIP-based services. In Fig. 3.3, there are four groups of users, $U1$ through $U4$, and each user in Ux registers itself with Sx (x = 1, 2, 3, 4).

Each edge between nodes represents the route through which SIP messages are exchanged between the nodes. The SIP message from user Ux is first sent to SIP proxy server Sx, and then, Sx relays the message to another SIP proxy server Sy (where $x \neq y$) that accommodates corresponding user Uy. In the rest of this dissertation, an edge or a sequence of edges is denoted by concatenated strings that represent nodes (i.e., S or U) with "-" (hyphen). For example, $U1$-$S1$ and $S1$-$S2$-$U2$ represent the logical link between $U1$ and $S1$ and between $S1$ and $U2$ via $S2$, respectively. There are some network segments in a logical link.

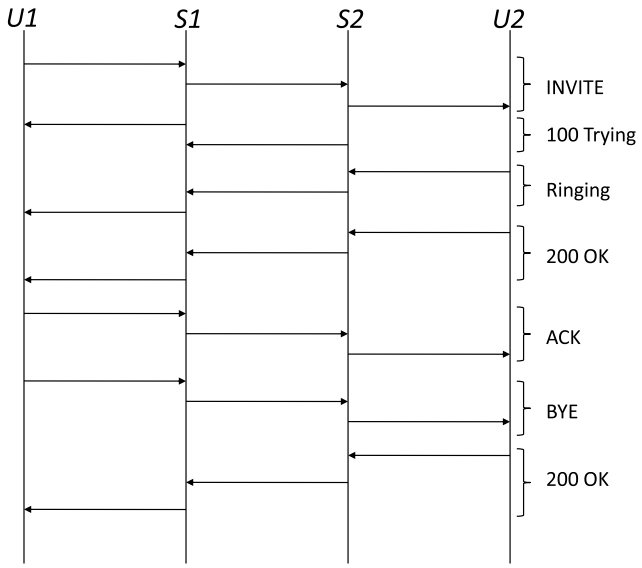


Figure 3.4: SIP signaling call flow

Table 3.1: Relationship between a SIP message group and a retransmitted SIP message

Message pair (First/Second)	Retransmitted message
INVITE/100Trying	INVITE
200OK/ACK	200OK
BYE/200OK	BYE

Each edge is potentially a lossy link. this dissertation assumes that the SIP messages take the symmetric route and network faults occurs on a network segment. There are some network segments in a logical link, and multiple logical links exist on a common network segment.

In this dissertation, this dissertation assumes the SIP signaling call flow in Fig. 3.4 defined in [3]. The 100Trying message from the callee UE in $S2$-$U2$ may be omitted in many implementations. In this call flow, there is a Request-Response relationship among two SIP messages, such as INVITE and 100Trying. In this dissertation, we call the pair of such two SIP messages a "message pair". Table 3.1 shows the message pairs in the SIP signaling call flow in Fig. 3.4. In the case that either of a SIP message in a certain message pair is lost, the SIP signaling call flow always restarts from the SIP message which corresponds to the Request message. This is called a retransmission mechanism. Table 3.1 also shows the SIP messages from which the SIP signaling call flow will restart.

There are three types of the SIP retransmission mechanism, hop-by-hop retransmission, end-to-end retransmission, and no retransmission. Each of SIP message pairs is defined to conduct either retransmission of the three types. Therefore, the SIP message pairs can be categorized based on the conducted retransmission type. In this dissertation, we refer to the category as a message group, or MG. For the message group of hop-by-hop retransmission (MG1), every SIP node (SIP proxy server or users) retransmits the SIP messages. For the message group of end-to-end retransmission (MG2), only users retransmit the SIP messages. For the message group of no retransmission (MG3), no SIP node retransmits a SIP message even if the SIP message is lost. Table 3.2 summarizes the message groups.

Table 3.2: Types of message group

Type of message group	Retransmission mechanism	way of retransmission
MG1	Provided	Hop-by-hop*
MG2	Provided	End-to-end
MG3	No Provided	-

*Except between SIP proxy server and callee

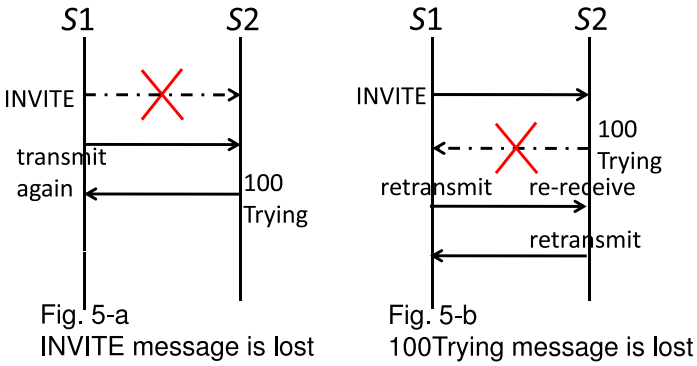


Figure 3.5: Example of retransmission mechanism of MG1

3.3.3 Locating Single Lossy Link

Focusing on the SIP's retransmission mechanism, there are relationships among the logical links where the lossy occurred, the SIP message to be retransmitted, and SIP node(s) to re-receive and retransmit it. The lossy link can be identified based on the relationships.

Fig. 3.5 illustrates the relationships in MG1, such as INVITE/100tyring message pair. In MG1, there are two cases of message loss, i.e., that the first message of MG1 (INVITE in Fig. 3-5-a) is lost and that the second message of MG1 (100trying in Fig. 3-5-b) is lost. Fig. 3.5 shows the two cases in $S1$-$S2$.

Relationship $R1$: In the case that the first message (from $S1$) of MG1 is lost, $S1$ transmits the first message again. In this case, the number of the retransmission at $S1$ and $S2$ does not change. S1 does not retransmit but repeats trasmitting the first message

Relationship $R2$: In the case that the second message (from $S2$) of MG1 is lost, $S1$ retransmits the first message. In this case, the number of the retransmission at $S2$ is incremented as depicted in Fig. 3-5-b.

Relationship $R3$: In the case that the second message (from $S2$) of MG1 is lost, $S2$ re-receives the first message and retransmits the second message. In this case, the number of the retransmission at $S2$ is incremented as depicted in Fig. 3-5-b.

These relationships of MG1 are also applicable in $U1$-$S1$.

Fig. 3-6 illustrates the relationships of MG2, such as 200OK/ACK message pair. In MG2, there are six cases of message losses, i.e., the first message of MG2 (200OK in Fig. 3-6-a) is lost in $U1$-$S1$, $S1$-$S2$, or $S2$-$U2$, and the second message of MG2 (ACK in Fig. 3-6-b, c and d)

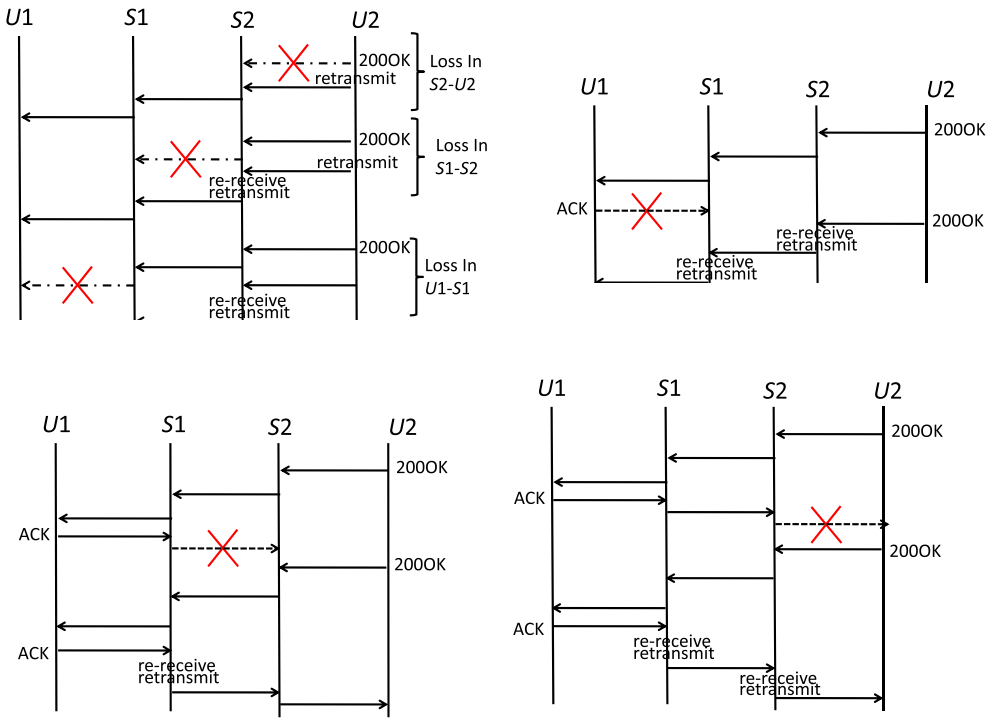


Figure 3.6: Example of retransmission mechanism of MG2

is lost in $U1$-$S1$, $S1$-$S2$, or $S2$-$U2$, respectively.

Relationship $R4$: In the case that the first message (from $U2$) or the second message (from $U1$) of MG2 is lost, $U2$ retransmits the first message. Whenever the SIP message of MG2 is lost, $S1$ and $S2$ re-receive and retransmit the first message. By the first message of MG2, the lossy link cannot be identified because the first message always is retransmitted every time.

Relationship $R5$: In the case that the second message (from $U1$) of MG2 is lost in $S1$-$S2$ or in $S2$-$U2$, the number of the retransmission at $S1$ is always incremented, as shown in Figs 3-6-c and d. In the case that the second message (from $U1$) is lost in $S2$-$U2$, the number of the retransmission at both $S1$ and $S2$ are incremented. In these cases, we focus on the number of the retransmission for the second message of MG2 in the adjacent two SIP proxy servers. The number of the retransmission at S1 is incremented in both cases when the second message of MG2 is lost in S2-U2 or S1-S2, whereas the number of the retransmission at S2 is incremented in only the case when the second message is lost in S2-U2.

Table 3.3 summarizes the relationships ($R1$-$R6$) among the logical links where the loss occurred, the SIP message to be retransmitted, and SIP node(s) to re-receive and retransmit it. Table 3.3 indicates that relationships $R1$, $R2$, $R3$ and $R5$ can be applied to locate the lossy link. It is because the lossy link always can be identified in a logical link. we will describe about $R6$ in Table 3.3 in the following subsection.

Table 3.3: Summary of retransmission characteristics

	Message group	SIP server action	Locations of logical link(s)
$R1$	MG1	Retransmission of first message	A logical link to which the message is retransmitted by SIP proxy server.
$R2$		Retransmission of first message	A logical link to which the message is retransmitted by SIP proxy server.
$R3$		Retransmission of second message	A logical link to which the message is retransmitted by SIP proxy server.
$R4$	MG2	Retransmission of first message	N/A (cannot be used for determination)
$R5$		Retransmission of second message at $S1$ and $S2$ or at $S1$	A logical link or subsequence logical links to which the retransmitted message is relayed
$R6$		Retransmission of second message at $S1$ and $S2$ or at $S1$	If the number of the retransmission at $S2$ is incremented and the number of the retransmission at $S1$ is more than that at $S2$, a logical link($S1$-$S2$) and adjacent logical link ($S2$-$U2$) to which the retransmitted message are lossy links

3.3.4 Locating Multiple Lossy Links

A message group categorized into MG1 as shown in Table 3.2 can be used for locating multiple lossy links. This is because MG1 can discover lossy links in a hop-by-hop manner.

If there are lossy links in $S1$-$S2$ and in $S2$-$U2$, the number of the retransmission are incremented in both $S1$ and $S2$. By only relationship $R5$, we cannot locate these multiple lossy links accurately because it is impossible to identify the retransmission at $S1$ and at $S2$. For locating multiple lossy links, the proposed method needs to compare the number in the adjacent two SIP proxy servers.

When there is the retransmission of the second message at $S2$, $S2$-$U2$ is located to be a lossy link. Additionally when there are more retransmissions of the second message of MG2 at $S1$ than ones at $S2$, $S1$-$S2$ is also located as a lossy link. It is because the count at $S1$ includes the number of the retransmitted second messages which are lost at $S1$-$S2$ and $S2$-$U2$.

Relationship $R6$: In the case that the second message of MG2 is lost in $S1$-$S2$ and $S2$-$U2$, the number of the retransmission of the second message at both $S1$ and $S2$ are incremented. In this case, the number of the retransmission at $S1$ is more than that at $S2$.

Based on relationship $R1$, $R2$, $R3$ and $R6$, we can locate multiple lossy links.

Table 3.4: Message groups and applicable relationships ($R1$-$R5$) for locating single lossy link

Message	Message group	Applied relationship	Lossy links discovered by $S1$	Lossy link discovered by $S2$
INVITE	MG1	$R2$	$U1$-$S1$	$S1$-$S2$
100trying	MG1	none	-	-
180ringing	MG3	none	-	-
200OK	MG2	$R4$ and $R5$	-	-
ACK	MG2	$R5$	$S1$-$S2$ or $S2$-$U2$	$S2$-$U2$
BYE	MG2	$R4$	-	-

3.4 Implementation

This section shows how we apply and implement the proposed method for the actual SIP signaling call flow.

3.4.1 Implementation of MIB in SIP Proxy Server

IETF have standardized MIB[11] for SIP[12] that have various MIB which count the number of the SIP messages handled by the SIP proxy servers. SNMP[19] enables us to obtain the number about the retransmitted SIP messages, which the adjacent SIP nodes retransmitted.

The SIP proxy server supporting the MIB counts the number of the retransmitted SIP messages (e.g., INVITE and ACK) and for the individual proximate nodes of the retransmitted message (e.g., sipCommonStatsRetries object)[12]. For example, when $S2$ in Fig. 3.4 re-receives the ACK message from $S1$, $S2$ increments the number. The MIB is prepared for each pair of $S1$ (sender node) and $U2$ (next relay node).

3.4.2 Applying the Relationships to Actual SIP Singaling Call Flow

Considering the defined MIB in SIP proxy server, we show which relationships are applicable to locate the lossy link. Table 3.4 summarizes the message and the applicable relationship ($R1$-$R6$) based on the Table 3.3 for locating single lossy link in the SIP signaling call flow depicted in Fig. 3.4.

Once $S1$ receives a retransmitted INVITE message from $U1$, we can determine $U1$-$S1$ as the lossy link due to the 100Trying message lost at $U1$-$S1$. Thus, by seeing the retransmission of the INVITE message, we can also locate $U1$-$S1$ as the lossy link. This relationship is also applicable in $S1$-$S2$. There is no MIB for the retransmission of 100Trying in the defined MIB so that we cannot use the 100Trying message.

If only $S1$ retransmits the ACK message to $S2$ and $S2$ does not re-receive the ACK, we can determine $S1$-$S2$ as the lossy link. If both $S1$ and $S2$ re-receive the ACK message, we can determine $S2$-$U2$ as the lossy link. Thus, we can determine $S1$-$S2$ or $S2$-$U2$ as the lossy link by counting the number of the retransmitted SIP messages at $S1$ and at $S2$.

The 200OK message belongs to both relationships $R4$ and $R5$ in the SIP signaling call flow in Fig. 3.4 . Because it is impossible to identify which relationship ($R4$ or $R5$) the retransmission of the 200OK message belongs to, the 200OK message cannot be used for locating the lossy link.

Table 3.5: Message groups and applicable relationships ($R1$-$R6$) for locating multiple lossy links

Message	Message group	Applied relationship	Lossy links discovered by $S1$	Lossy link discovered by $S2$
INVITE	MG1	$R2$	$U1$-$S1$	$S1$-$S2$
100trying	MG1	none	-	-
180ringing	MG3	none	-	-
200OK	MG2	$R4$ and $R6$	-	-
ACK	MG2	$R6$	$S1$-$S2$ and $S2$-$U2$	$S2$-$U2$
BYE	MG2	$R4$	-	-

Table 3.5 summarizes the message and the applicable relationship ($R1$-$R6$) based on the Table 3.5 for locating multiple lossy link. Relationship $R6$ in Table 3.5 is different from that in Table 3.4. The INVITE message is also applicable to locate multiple lossy links in $U1$-$S1$ and $S1$-$S2$. By comparing the number of the retransmitted ACK in $S1$-$S2$ and $S2$-$U2$ based on the relationship $R6$, we can locate multiple lossy links in $S1$-$S2$ and $S2$-$U2$, or in only $S2$-$U2$.

3.4.3 Algorithm of the Proposed Method

This dissertation shows the overview about the algorithm of the proposed method. The monitoring server collects the number of the retransmission from all the SIP proxy servers. The algorithm obtains the number of one SIP proxy server $Si(i = 1, 2, ..., n)$.

1. The monitoring server examines whether there is the lossy link in Si - Ui matching relationship $R2$, and memorizes the link as a lossy link if it matches.

2. The algorithm obtains the number of the counterpart SIP proxy server $Sj\,(j = 1, 2, \cdots, n, (i \neq j))$.

3. The monitoring server examines whether there is the lossy link in Si-Sj matching relationship $R2$, and memorizes the link as a lossy link if it matches.

4. The monitoring server examines whether there is the lossy links inSi-Sj and inSj-Uj matching relationship $R5$ and $R6$ by comparing the number from Si to Sj to that from Sj to Uj ,and memorizes the links as lossy links if it matches.

5. The algorithm obtains the number of the other SIP proxy server $(j = 1, 2, \cdots, n)$ and returns the procedures 2., until it obtains those of n SIP proxy servers. After obtaining that of n SIP proxy servers, the algorithm go to the procedure 6.

6. The algorithm obtains the number of the other SIP proxy server $(i = 1, 2, \cdots, n)$ and returns the procedure 1. ,until it obtains those of n SIP proxy servers.

3.5 Numerical Model for System Evaluation

In this section, we derive a numerical model to evaluate the proposed method in terms of how often the proposed method fails to locate the lossy link.

3.5.1 Evaluation Metric

Although network faults actually exist in the logical links including the SIP proxy server, the proposed method may fail to locate the lossy links when the SIP proxy server does not receive the required retransmitted SIP messages. we evaluate how often the proposed method encounters such failures.

This dissertation introduces the fault-detection failure rate as an evaluation metric. The rate is the probability of failing to locate the lossy link within an interval. The fault-detection failure rate represents the probability of not receiving retransmitted SIP messages that could be utilized for locating the lossy link, i.e., the INVITE and ACK messages in the case of the SIP signaling call flow specified in Fig. 3.4, within the interval.

We derive the relationship among the rate fault-detection failure rate M, the average rate of call arrival λ, the interval I for obtaining the number from SIP proxy servers and the message loss rate p. The call arrival indicates that a user starts the SIP signaling call flow to make a call to another user. We define that the message loss rate is related to the packet loss rate of the network segment that has a fault. The interval and the message loss rate influence on the number of the retransmitted SIP messages within the interval. We consider that network faults exist on a network segment and the message loss rate on a network segment is equal to the packet loss rate on the network segment with the fault, and that call arrival from each group of users obeys a Poisson distribution.

3.5.2 Model for Logical Link between SIP Proxy Server and Users

In order to locate a lossy link between users (e.g., $U1$) and the SIP proxy server (e.g., $S1$), at least one of the subsequent two events needs to occur:

a) $S1$ re-receives INVITE messages from $U1$. (relationship $R2$)

b) $S1$ increments the number of retransmitted ACK messages from the other SIP proxy server to $S1$, and that SIP proxy server increments the number of retransmitted ACK messages from users of that SIP proxy server to $S1$ (relationship $R5$)

First, we introduce the fault-detection failure rate in the event a). Let $P_{S1-k}(I, \lambda)$ denote the probability that $S1$ deals with k call requests (i.e., relays the SIP messages) from $U1$ that register with $S1$ within an interval. $P_{S1-k}(I, \lambda)$ is derived as follows:

$$P_{S1-k}(I, \lambda) = \frac{(\lambda I)^k}{k!} e^{-\lambda I} \tag{3.1}$$

Suppose that the average call arrival rate from all groups of users (e.g., $U1$, $U2$.) is equal. Let $X_k(p)$ denote the probability that no INVITE message is lost in k calls. The probability $X_k(p)$ is derived as follows:

$$X_k(p) = (1-p)^k \tag{3.2}$$

Let $Y_{S1-k}(I, \lambda, p)$ denote the probability that any retransmissions of INVITE message are not included in the k times of calls in $S1$. Since each call arrival arises independently, the probability $Y_{S1-k}(I, \lambda, p)$ is derived as follows:

$$\begin{aligned} Y_{S1-k}(I, \lambda, p) &= X_k(p) P_{S1-k}(I, \lambda) \\ &= (1-p)^k \frac{(\lambda I)^k}{k} e^{-\lambda I} \end{aligned} \tag{3.3}$$

Let $A_{S1}(I, \lambda, p)$ denote the fault-detection failure rate in the event a) in $S1$ for a logical link in $S1$-$U1$ focusing on the retransmission of the INVITE message. We can obtain $A_{S1}(I, \lambda, p)$ by the summation of $Y_{S1-k}(I, \lambda, p)$ for all k as follows:

$$\begin{aligned} A_{S1}(I, \lambda, p) &= \sum_{k=0}^{\infty} Y_{S1-k}(I, \lambda, p) \\ &= \sum_{k=0}^{\infty} (1-p)^k \frac{(\lambda I)^k}{k!} e^{-\lambda I} \\ &= e^{-\lambda I} \sum_{k=0}^{\infty} \frac{((1-p)\lambda I)^k}{k!} \\ &= e^{-\lambda I + ((1-p)\lambda I)} = e^{-p\lambda I} \end{aligned} \tag{3.4}$$

This implies that the fault-detection failure rate exponentially decreases as the average rate of call arrival, interval, or message loss rate increases.

Secondly, we introduce the fault-detection failure rate in the event b). We define one of the SIP proxy servers that relay the SIP messages to $S1$ as Sn. Let N denote the number of SIP proxy servers. In this dissertation, calls from a group of users (e.g., $U1$) arrive at each group of users equally. Hence, the average rate of call arrival relayed from a SIP proxy server to another SIP proxy server is equal to λ/N. Let $Q_{Sn-k}(I, \lambda, N)$ denote the probability that Si deals with k calls (i.e., relays the SIP messages) from Ui. $Q_{Si-k}(I, \lambda, N)$ is derived as follows:

$$\begin{aligned} Q_{Si-k}(I, \lambda, N) &= P_{Si-k}(I, \frac{\lambda}{N}) \\ &= \frac{(\frac{\lambda}{N} I)^k}{k!} e^{-\frac{\lambda}{N} I} \end{aligned} \tag{3.5}$$

Let $Z_{Si-k}(I, \lambda, p, N)$ denote the probability that any retransmissions of the ACK message are not included in the k times of calls from Si to $S1$. Let $B_{Si}(I, \lambda, p, N)$ denote the fault-detection failure rate in the event b) for a logical link in $S1$-$U1$ when $S1$ and Si exchange the SIP messages. We can obtain $B_{Si}(I, \lambda, p, N)$ from Si to $S1$ by summation of $Z_{Si-k}(I, \lambda, p, N)$ for all k.

$$\begin{aligned} B_{Si}(I, \lambda, p, N) &= \sum_{k=0}^{\infty} Z_{Si-k}(I, \lambda, p, N) \\ &= \sum_{k=0}^{\infty} (1-p)^k Q_{Si-k}(I, \lambda, N) \\ &= \sum_{k=0}^{\infty} (1-p)^k \frac{\left(\frac{\lambda}{N} I\right)^k}{k!} e^{-\frac{\lambda}{N} I} = e^{-p\frac{\lambda}{N} I} \end{aligned} \tag{3.6}$$

Let $C_{S1}(I, \lambda, p)$ denote the fault-detection failure rate in the event b) for a logical link in $S1$-$U1$ when $S1$ exchanges the SIP messages with N SIP proxy servers. $C_{S1}(I, \lambda, p)$ is obtain by multiplication of $B_{Si}(I, \lambda, p, N)$ as follows:

$$C_{S1}(I, \lambda, p) = \left(e^{-p\frac{\lambda}{N} I}\right)^N = e^{-p\lambda I} \tag{3.7}$$

Note that there is the call originated from $U1$ returns to $U1$. Hence, not *N-1* but N SIP proxy servers treat calls reaching $U1$. Let $M_{US}(I, \lambda, P)$ denote the fault-detection failure rate when

the lossy link is a logical link between the SIP proxy server and users. $M_{US}(I, \lambda, P)$ is derived, as follows:

$$\begin{aligned} M_{US}(I, \lambda, P) &= A_{S1}(I, \lambda, p)\, C_{S1}(I, \lambda, P) \\ &= e^{-2p\lambda I} \end{aligned} \tag{3.8}$$

3.5.3 Model for Logical Link between SIP Proxy Servers

In order to locate the lossy link between the SIP proxy server (e.g., $S1$) and another SIP proxy server (e.g., $S2$), at least, one of the following four events needs to occur:

<1> $S1$ retransmits INVITE messages from $S2$.

<2> $S2$ retransmits INVITE messages from $S1$.

<3> $S1$ increments the number of retransmitted ACK messages from $U1$ to $S2$, and $S2$ does not increment the number of retransmitted ACK messages from $S1$ to $U2$

<4> $S2$ increments the number of retransmitted ACK messages from $U2$ to $S1$, and $S1$ does not increment the number of retransmitted ACK messages from $S2$ to $U1$.

Suppose that the average rate of arrival call from $S2$ to $S1$,and that from $S1$ to $S2$ are equal to λ/N. Let $D_{S1}(I, \lambda, p, N)$ denote the fault-detection failure rate in the event <1> for a logical link in $S1$-$S2$. $D_{S1}(I, \lambda, p, N)$ is derived from the average rates of call arrival λ/N and Equation (4), as follows:

$$D_{S1}(I, \lambda, p, N) = e^{-p\frac{\lambda}{N}I} \tag{3.9}$$

Let $E_{S1}(I, \lambda, p, N)$ denote the fault-detection failure rate in the event <3> for a logical link in $S1$-$S2$. $E_{S1}(I, \lambda, p, N)$ is derived from Equation (6), is derived from the average rates of call arrival λ/N and Equation(6) as follows;

$$E_{S1}(I, \lambda, p, N) = e^{-p\frac{\lambda}{N}I} \tag{3.10}$$

Let $M_{SS}(I, \lambda, p)$ denote the fault-detection failure rate on a logical link between SIP proxy servers.

$M_{SS}(I, \lambda, p, N)$ is derived, as follows:

$$M_{SS}(I, \lambda, p) = D_{S1}D_{S2}E_{S1}E_{S2} = e^{-4p\frac{\lambda}{N}I} \tag{3.11}$$

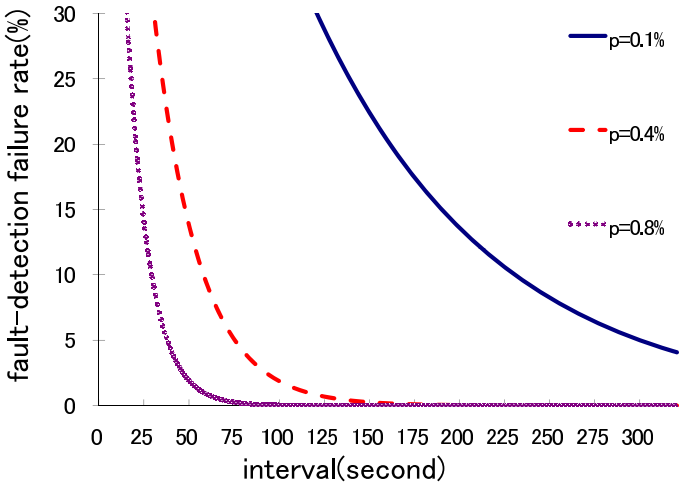
This implies that the fault-detection failure rate increases with increasing the number of SIP proxy servers, whereas the fault-detection failure rate decreases with increasing the average rate of call arrival, interval, or message loss rate.

3.6 Evaluation

3.6.1 Parameter Derivation

We evaluate how accurately the proposed method can locate the lossy links. For the evaluation of the fault-detection failure rate, We adopt 0.1%, 0.4% and 0.8% as the message loss rate.

We introduce that the average call arrival rate in one second is 5 in a commercial SIP proxy server, which is based on the telecom data book[21] in Japan and the following assumptions.



The data book says the annual average rate of call arrival from an individual user is about 1.5 times per day. Further, there is a commercial product of SIP proxy server that can accommodate 800,000 users at maximum. If the service operator makes each SIP proxy server accommodate users up to 40% of its capacity, the average rate of call arrival per SIP proxy server is about 5 calls per second. It is possible for SIP proxy server to accommodate more than 40% of its capacity. Considering the severe condition, we adopt 40% as the capacity. If the rate of call arrival increases, the fault-detection failure rate decreases. In the case that service operator accommodates 30 million users, service operator needs 100 SIP proxy servers for users.

3.6.2 Discussion on Fault-Detection Failure

First, this dissertation analyzes fault-detection failures with regard to each of the logical links between users and a SIP proxy server (*U-S*), and between two SIP proxy servers (*S-S*). Second, we discuss how much the fault-detection failure rate in *S-S* can be improved, considering the typical network architecture of SIP-based services in the carrier network.

Fig. 3.7 shows the fault-detection failure rate in *U-S* that is obtained from Equation (8). The X-axis represents the interval in which the monitoring server collects the number of all SIP servers and locates the lossy links, while the Y-axis represents the fault-detection failure rate. There is a trade-off between the interval and fault-detection failure rate. If the interval becomes shorter, the fault-detection failure rate becomes higher. The network operators adopting the proposed method in SIP-based services needs to be careful of the trade-off. Considering a five-minute interval, which is the same interval as that of MRTG[20] for collecting traffic statistics, the fault-detection failure rates are almost 5%, 0% and 0% where the message loss rates are 0.1% or 0.4%, and 0.8%, respectively. As a result, the proposed method achieves the low fault-detection failure rate in a five-minute interval.

Fig. 3.8 shows the fault-detection failure rate in *S-S* that is obtained from Equation (11). In this analysis, the number of SIP servers N is 100 (30 million users as described in the previous section). The X-axis represents the interval, while the Y-axis represents the fault-detection failure rate. Achieving the low fault-detection failure rate in *S-S* takes longer than it does in *U-S*.

Considering the typical network architecture of SIP-based services such as in Fig. 3.1, the fault-detection failure rate can be decreased. The network segment between the network

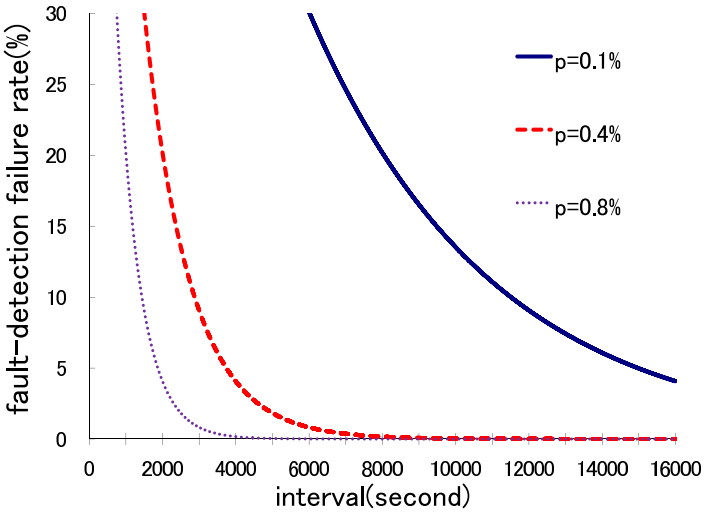


Figure 3.8: Fault-detection failure rate in *S-S*

centers includes multiple logical links because logical links lie in full mesh among all SIP servers. Locating at least one logical link of them in the common network segment is sufficient to determine the network faults that cause the SIP message loss.

The following equation shows how much the aggregated logical links can decrease the fault-detection failure rate. Let m denote the number of SIP proxy servers in one network center. For simplicity, we assume that each network center has the same number of SIP proxy servers. Hence, m^2 logical links are in the network segment between two areas. Let $F(I, \lambda, p, N, m)$ denote the probability of failure to determine network faults on network segments aggregating m^2 logical links share. Hence, $F(I, \lambda, p, N, m)$ is derived from Equation (11), as follows:

$$F(I, \lambda, p, N, m) = \left(e^{-4p\frac{\lambda}{N}I}\right)^{m^2} = e^{-4p\frac{m^2\lambda}{N}I} \tag{3.12}$$

When there are 10 areas in order to distribute the SIP proxy servers, SIP proxy servers in single area are ten ($m = 10$) in number (total numbers (N) of the SIP proxy server do not change), and $F(I, \lambda, p, N, m)$ is derived as $e^{-4p\lambda I}$. Fig. 3.6.2 shows the fault-detection failure rate on the network segment in aggregating *S-S* in the case of m=10 and N=100. The X-axis and Y-axis is same with Figs. 3.7 and 3.8. In this case, the time to locate the lossy links becomes less than 1/100 times, compared to single *S-S* case shown in Fig. 3.8 .

By comparing Equations (8) and (12), when $m > \sqrt{N/2}$, the network segment between two areas can have the low level of the fault-detection failure. As a result, the proposed method achieves the low fault-detection failure rate in a five-minute interval. This aggregation can decrease the fault-detection failure rate.

3.7 Summary

This dissertation proposed an operational method to locate the lossy links where SIP messages are lost, by using the number of the retransmitted SIP messages based on the relationship in Tables 4 and 5. This dissertation clarified what kind of request/response message pair can be utilized to determine the location of the SIP message loss. By utilizing the number of the retransmitted SIP messages, the storage and computational resources required by the proposed method is much less than that of the conventional method which collects all the SIP messages.

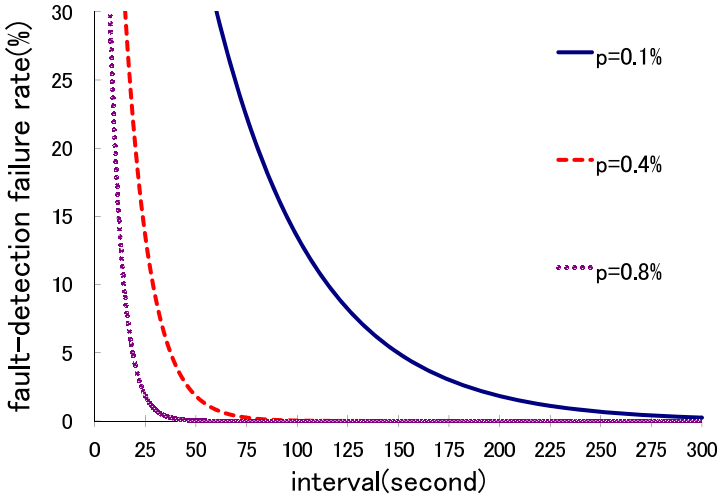


Figure 3.9: Fault-detection failure rate on network segment in aggregating *S*-*S* in case of m=10 and N=100

With a numerical model, we evaluated the fault-detection failure rate of the proposed method during the monitoring interval. In this evaluation, we revealed two features. First, there is a trade-off between the monitoring interval and the fault-detection failure rate. Second, the fault-detection failure rate on a network segment can be decreased as the number of the logical links aggregated on the network segment increases.

We can conclude that the proposed method is applicable for locating the lossy link of the SIP signaling call flows. If the network operators locate the lossy links quickly, they can determine the solution for the problems which caused the SIP message loss.The relationship of Table 3 is applicable to the SIP signaling call flow that includes the other types of SIP messages (e.g., "PRACK"[22], "Cancel" and "ACK for a non-2xx final response"). , and the case that there are more than three SIP proxy servers.

Future research work includes applying the proposed method to different signaling-based services, such as IMS (IP Multimedia Subsystem)[4] in which SIP messages are relayed by more than two SIP proxy servers. Because the algorithm for the detection is based on the basic retransmission pattern of the signaling messages, the proposed method can be easily applied for the other signaling protocol.

Chapter 4

Proposal of Efficient Backup of Session States

4.1 Background

As the efficient method to backup the session states, this dissertation proposes selective storing of the session states. This method focuses on the CSCFs in the IMS-based services. Because the algorithm to decide the points where the session states are stored is based on the basic retransmission pattern of the signaling protocol, the proposed method can be easily applied for the new signaling protocol.

In the case where the IMS encounters any fault, such as CSCFs fault, UEs are not able to receive the services smoothly. This situation is caused by the disruption of the SIP signaling call flow. The SIP signaling call flow is defined as a series of exchanged SIP messages that control and manage the services. The literature [23] describes that many users feel dissatisfied with telephone services when the SIP signaling call flow is not completed within 6.00 seconds. To complete the SIP signaling call flow without delay is vital to provide satisfactory services.

For service continuity in the case of server failures, service operators generally prepare some level of restoration mechanism. This is basically accomplished by arranging replacement servers. As the restoration system, service operators have used an $n + k$ redundancy model [24, 25] to reduce the number of servers, where n and k ($n > k$) denote the numbers of active servers handling the SIP messages from UEs and backup servers storing the session states from the active servers.

This redundancy model is applicable to SIP proxy servers as described in previous studies [26, 27]. A replacement server takes over the configuration and session states maintained by the halted server. The session states correspond to registration information, transaction and dialog. These are detailed in Section 4.3.4.

In order to hand over the session states kept by active servers, the session states are synchronized between the active and backup servers. Whenever the CSCFs treat the SIP messages, the previous studies store the session states from the active to the backup servers. Because so many processes are needed to store them for all the CSCFs, service operators need to maintain a large number of backup servers. If there is an insufficient number of backup servers, it introduces a delay in the SIP signaling call flow because the process to store the session states becomes congested. It leads to degradation of service quality. Reducing the number of backup servers without degrading service quality is a technical challenge.

This dissertation proposes an IMS restoration system having two features: selectively storing the session states by utilizing the retransmission mechanism of SIP [28], and prioritizing the

specific session states kept by the halted one. This method allows the service operator to reduce the number of backup servers. Although the number of times to store the session states is reduced by the proposed method, the service continuity can be achieved because the SIP signaling call flow is not disrupted. The proposed method cannot be applied for the critical telephone service which requires restoring the CSCFs within the retransmission timer (500 milliseconds), but is effective for the VoIP service which permits spending more than 500 milliseconds on restoring the CSCFs and requires the rapid completion of call setup delay.

4.2 Requirements for IMS Restoration System

Our restoration system aims to restore the CSCFs without disrupting the SIP signaling call flow and to minimize the number of backup servers. There are the following four requirements (R1-4) for the restoration system when CSCFs encounter a fault, that is, a case where CSCFs do not respond in any interaction:

R1 to detect a fault of a CSCF so as to start the process for restoring the CSCF;

R2 to store the session states so as to prepare the replacement CSCF when the faults occurred;

R3 to minimize the number of backup servers; and

R4 too avoid disruption of the SIP signaling call flow handled by the halted CSCF.

Regarding R1, the restoration system needs to detect a fault of a CSCF so as to start the process for the CSCF. Regarding R2, a simple method is to synchronize the session states between the active and backup servers every time the session states are updated in the SIP signaling call flow. However, this causes the delays of the SIP signaling call flow in the case where this synchronization process is performed every session state update. We consider the method which does not delay the SIP signaling call flow.

Regarding R3, reducing the number of times that the session states are stored is essential to reducing the number of the backup servers. Synchronizing the session states between active and backup servers every time the session states are updated leads to many repetitions of the process of storing the session states in the backup servers. This also increases the load of the backup servers. To decrease the load of each backup server, more backup servers are required.

Regarding R4, preparing the replaced CSCF and to copy the essential session states quickly is required in order to take over processing session states when the fault is detected. The important aim of this replacement is to hide the CSCF fault from UEs. A key is that restoration of the critical session state is rapidly performed.

There is a reference expressing the service level requirement for the IMS-based services. One referenced value regarding service level requirement is 6.00 seconds. This value is (as explained in the literature [5]) the 95th percentile value of the call setup delay distribution that will satisfy customers receiving the telephone services; in brief, it is an acceptable call setup delay. It is desirable that the call setup delay completes within the acceptable call setup delay, even when a fault has occurred and the CSCFs are restored.

In terms of system construction, the restoration system should be built in the network near the active server so as to minimize the time for storing the session states without delaying the completion of SIP signaling call flow and to achieve their rapid restoration, i.e., so as to satisfy R2 and R4,respectively. In the case where the IMS is developed over multiple data centers, a function set of the restoration system should be built in each data center for the rapid restoration.

4.3 IMS Restoration System

4.3.1 System Overview

In the IMS, there are three types of CSCFs: Proxy-CSCF (P-CSCF), Serving-CSCF (S-CSCF), and Interrogating-CSCF (I-CSCF). P- and S-CSCFs maintain the session states of the UEs while they are registered with them, while I-CSCF provisionally manages (memorizes) the session states so as to verify portions of SIP signaling call flow.

We consider that the following three functions are necessary for our restoration system:

(1) a monitoring function to detect the faults of the CSCFs, and then to order backup servers to execute function (3);

(2) a synchronizing function to store the session states from active CSCFs to backup servers when CSCFs are still alive (before CSCFs encounter any fault); and

(3) a restoring function to copy the session states into an replacement CSCF and to minimize the time taken to restore the CSCFs in order to avoid disruption of the SIP signaling call flow.

We list which physical server executes each function as it follows:

Function(1): monitoring server

Function(2): backup server

Function(3): backup server

For function (1) satisfying R1, sending probe packets periodically is a promising way to check whether the CSCF is alive or not. For function (2) satisfying R2 and R3, we select the points in which the session states are stored in the SIP signaling call flow. The backup servers execute function (2)

Function (3) satisfying R4 copies the session states into the replacement CSCf so as to avoid the disruption of SIP signaling call flow. Because the CSCFs maintain some hundreds thousands of session states, it could be difficult for out restoration system to restore such a large number of session states instantaneously. Therefore, we define the critical and non-critical session states by distinguishing the types of session states stored in the backup servers and prioritizing the specific types of session states to enable rapid restoration.

Figure 4-1 shows the construction of our restoration system. Our restoration system consists of multiple active CSCFs in the roles of S-CSCF, and P-CSCF, monitoring server, backup servers, and a few replacement CSCFs inheriting the processes of SIP signaling from the active CSCFs. These entities accomplish the restoration of the CSCFs, when the active servers halt, with functions (1)-(3). The backup servers enable function (2) from multiple CSCFs indicated by the bidirectional arrow lines.

The management network transferring the stored session states should be separate, as a dedicated network, from the service network where the SIP messages are transferred. This can prevent the transferred session states from being dropped due to congestion of the service network.

The following steps are processed so as to restore the CSCF, when the fault of active CSCF occurred in a condition that the backup servers stores the session states in order that the restored CSCF treats the SIP messages retransmitted by UEs.The details are explained in Section 4.3.5.

step 1 Function (1) detects a fault in a CSCF, and orders restoring the CSCF

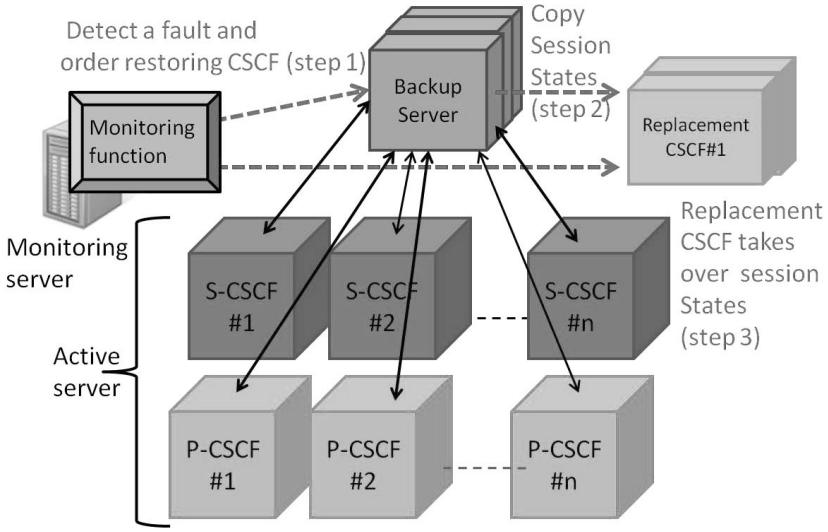


Figure 4.1: Construction of our restoration system.

step 2 The backup server restores the session states into the replacement CSCF (function (3)).

step 3 The replacement CSCF takes over the session states and deals with the subsequent SIP messages sent by CSCFs or UEs.

These steps correspond to the dashed arrow lines in Figure 4-1. The literature [23] says that 6.00 seconds is the value at the 95th percentile of the distribution of the acceptable call setup delay for the telephone services. Among the IMS-based services, we focus on the telephone service requesting the high service quality. We employ this value as a maximum delay limitation for the SIP signaling call flow, even when a fault occurs and the CSCFs are restored.

In the next subsection, we estimate the interval for sending probes in function (1), and in Section 4.3.3, we calculate how much time remains to carry out function (3) considering the delay limitation.

4.3.2 Estimation of Probe Sending Interval

A fault in the CSCF can be quickly detected if a short interval is used when function (1) sends probe packets. However, this interval should be carefully determined in order to avoid subjecting the network to a large load. The interval for sending probes depends on how many response messages of probes are not successively received and the timeout to wait for each corresponding response message at step 1 described in Section 4.3.1.

There usually are two points where the probe responses are lost. One is at the dedicated management network (network nodes), and the other is the monitoring function itself, specifically, the monitoring function falsely detects that a CSCF has halted.

The primary losses in the network node are due to router or switch device failures or network congestion. For a network device failure, Hot Standby Routing Protocol (HSRP) [33] has the ability to switch over two network devices in a short time, such as one second or less, when one of them halts.

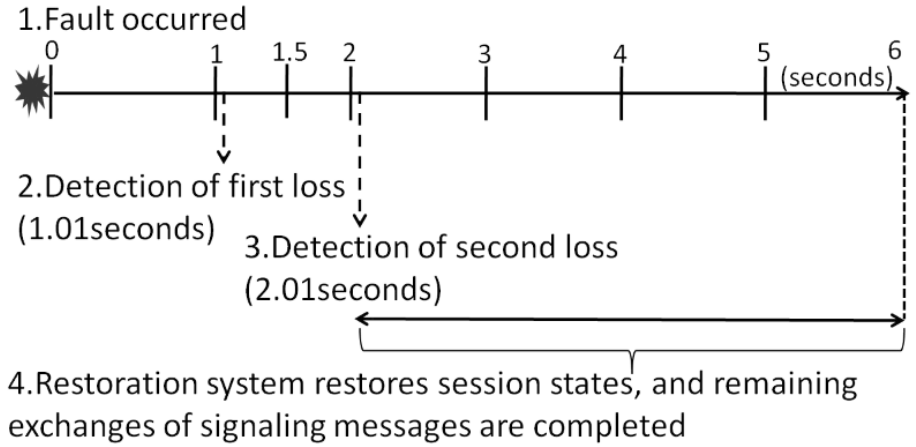


Figure 4.2: Time chart when our restoration system restores CSCFs.

A key to determine the interval of sending probes is fault detection time of network node taken by HSRP. In terms of detecting the CSCF faults, we take this one second taken by recovery of the network node fault for the interval of sending probes. Because, even though we take an interval much less than this one second, the monitoring function is forced to wait for the network node recovery triggered by HSRP (one second interval) for the one second.

From the derivation of the interval of sending probe (one second), two times or more of probes not to be successively responded are taken as the number of probes to determine the CSCF fault. Because, as mentioned above, there is an opportunity where network node failure may happen to lose a probe or its response while the network node recovery. Generally, more times to wait for the successive probe response losses can preventing more the CSCF fault detection failure, that is, wrongly judging the CSCF normally working as having any failure.

The congestion can be minimized by carrying out the facility design of the dedicated network corresponding to management traffic requirement. Additionally, queuing technology, e.g., priority queuing, can prevent network congestion influencing the probe and its response messages, even if momentary network congestion happens.

The dedicated management network can also minimize the chance to cause an extremely large transmission delay of the probe and its response messages. As addressed in Section 4.2, the restoration system is required to be constructed in the same network center. Therefore, even 10 milliseconds of timeout are sufficient to detect the CSCF faults.

When the probe packet is not lost because of the network congestion and the network device failure in succession, the interval for sending probes should be 1 second or less, and two losses are sufficient for making the decision that the CSCF has halted. To detect the fault of CSCFs, the monitoring function sends the probe packet every one second at twice. Comprehensively, we can take about 2 seconds for step 1, when a fault occurred in the CSCF shortly after the monitoring function receives a probe response message.

4.3.3 Estimation of Time for Restoring Session States

Figure 4-2 shows the time chart regarding the fault detection (function (1)) and restoration of session states (function (3)). Four events are numbered as they occur in a sequential manner. All the events should be completed within the delay limitation: 6.00 seconds.

The chart begins from the situation where a fault occurred in the CSCF (event 1) shortly

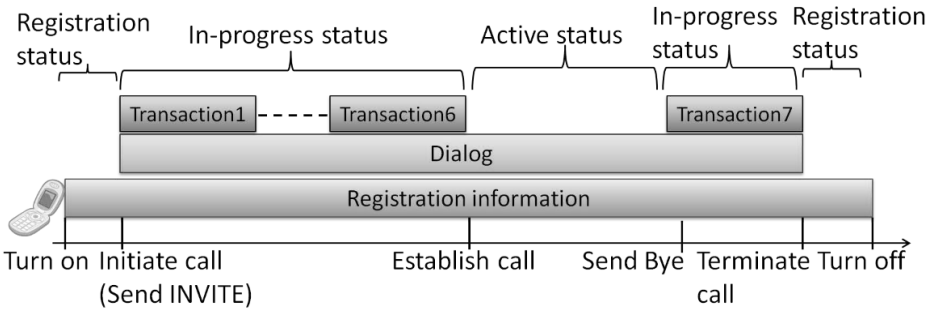


Figure 4.3: Relationship of each status, registration information, dialog and transaction.

after the monitoring function receives a probe response message. In this case, the monitoring function encounters the first loss (event 2) 1.01 (probe interval+ timeout) seconds after the fault occurs. Eventually, it makes a decision about the CSCF's fault after the timeout of the second probe response (event 3), that is, 2.01 seconds after the fault. Next, our restoration system executes steps 2 and 3, and finally, the SIP signaling call flow receives the CSCF's fault returns (event 4) to be processed once the replacement CSCF is available.

According to this event sequence, our restoration of session states must be assigned less than 3.99 (6.00 - 2.01) seconds at the maximum. In 3.99 seconds, the remaining SIP signaling call flow need to be completed after the CSCF recovery takes some time (hundreds of milliseconds). These two actions totally need to be taken less than 3.99 seconds.

4.3.4 Definition of Status of Session States

This subsection first defines three statuses representing the session states based on the feature of SIP. Based on the definition, this subsection secondly introduces the method to store the session states for each status in our restoration system.

Figure 4-3 shows the relationships of three statuses of the session states, and the named data types: registration information, dialog and transaction. The statuses change according to the operation of UEs: turning on, starting to communicate, and turning off.

Registration status represents the state when UEs register registration information with CSCFs but before the UEs initiate the call. *Registration information* is memorized by P-CSCF and S-CSCF and includes the authentication information, SIP URI, and IP address of UE. When the UE is turned on, it begins the registration procedure to be executed, and the UE periodically notifies its availability to the CSCFs. When the UE is turned off, the registration information is erased in the CSCFs.

In-progress status represents the state when the SIP signaling call flow is initiated among the UEs and CSCFs for the call initiation and termination. This status changes to the active status when the call is established, and it changes back to registration status when the call is terminated.

In addition to the registration information, the in-progress status includes dialog and transaction as the session state. *Dialog* is memorized by P-CSCF and S-CSCF. The dialog represents a specific session state and describes the progress of transactions in service initiation and termination call flows and the service-specific information used by the UE (transport-layer port number, encoding/decoding types of media of a call, etc.). *Transaction* is also memorized by P-CSCF and S-CSCF. The transaction represents a specific section in the SIP signaling call flow and allows the UE and CSCFs to maintain the retransmission of SIP messages. When the

SIP signaling call flow is resumed, the transaction is replaced with a new one.

Active status represents the state when the media (voice) are being transmitted in a call. In addition to the registration information, the active status includes dialog, but not transaction.

Based on the definition of each status, we propose that our restoration system takes different timings (opportunities) to store the session states are used for individual statuses. In the case of the registration status, when the registration procedures are completed the registered status is copied to the backup servers. In the case of in-progress status during the SIP signaling call flow, its status are copied to the backup servers multiple appropriate timings described in next subsection. This allows the restoring CSCF to take over session states without disruption of SIP signaling call flow even if the fault occurred in the previously active CSCFs.

The backup servers do not store the active status, because the in-progress status becomes the active status when the SIP signaling call flow is completed. Because the in-progress status is stored, the backup servers do not need newly to store the active status from the CSCFs.

4.3.5 Selective Storing of Session States

This subsection first explains the retransmission feature of SIP. Based on the feature, this subsection secondly introduces selectively storing of session states.

Figure 4-4 shows the SIP signaling call flow for the initiation and termination procedures in the IMS. The individual SIP messages are numbered. Before call initiation, UE-A and UE-B had registered themselves with the different P-and S-CSCFs. As shown in this figure, 7 transactions progress serially (except where transactions 4 and 5 overlap). The dialog is updated through the SIP signaling call flow.

When either the request or response message is lost in any segment between the UE and CSCFs, UE-A begins to retransmit the request message after its retransmission timer expires. This pair of request and response messages is maintained by the transaction. we focus attention on this feature for selective storing of session states.

This dissertation proposes selective storing of session states in order to reduce the number of times to store the session states without disrupting the SIP signaling call flow when faults occurred in the CSCFs. Whenever the in-progress status is updated, storing the session states from the CSCF delays the progress of SIP signaling call flow. Not to extremely delay the SIP signaling call flow, the number of times to store the session states needs to be at minimum. The viewpoint of 'selective' storing is to select the points in which the session states are stored in the SIP signaling call flow, in order that the restored CSCF can treat the SIP messages retransmitted by UEs.

In order to allow the replacement CSCF to continue processing the SIP signaling call flow, the session states stored in the backup server should fundamentally be consistent. A key to allow this is leveraging the retransmission feature of SIP messages; that is, the request message is retransmitted when either the request or response message in the transaction is lost.

Suppose that a CSCF has just finished sending the request message in transaction i, and the CSCF halts. Because no response message is received, the caller UE retransmits the request message after the replacement CSCF is ready. Therefore, it is sufficient for the recovered session states in the replacement CSCF to indicate the fact that the previous transaction (i-1) is complete. There is no need for the session states to indicate that the halted CSCF had received the request message of the target transaction i. For the restoration, the backup servers need to store the session states which are created after the CSCF received the response messages and tranfererred the SIP messages in the transaction.

The detail in the service initiation and termination procedures to store the session state is as follows: 1) when the CSCF receives the request message in each transaction, it just processes

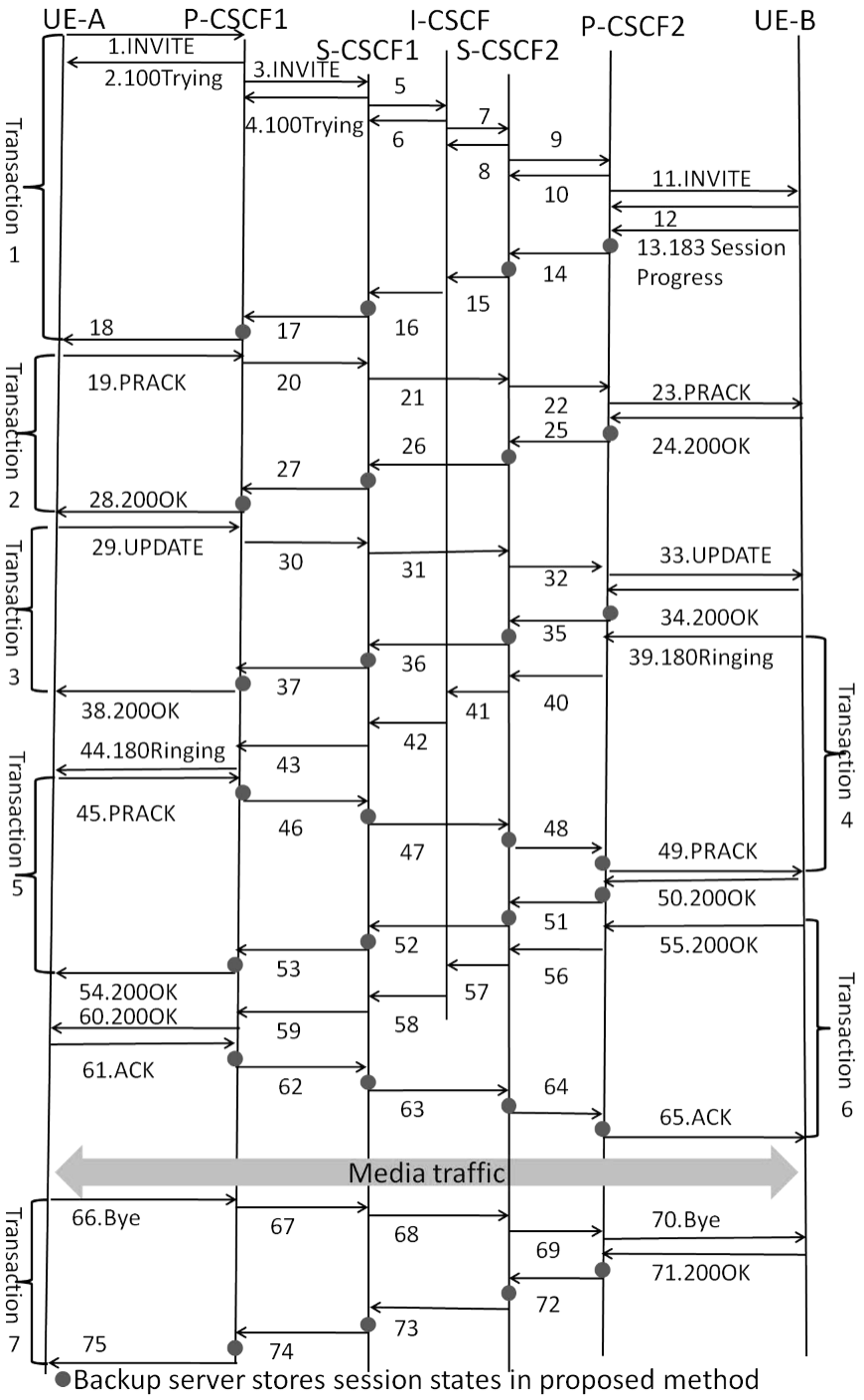


Figure 4.4: Service initiation and termination procedures in IMS.

and forwards the request message to the following UE or CSCFs; 2) when the CSCF receives the response message, it first runs processes to complete the transaction for the message and update the session state; 3) the CSCF then stores the updated session state; and 4) the CSCF forwards the response message to the following UE or CSCF. The circles in Figure 4-4 indicate the points at which the updated session states are stored in the backup server.

Table 4.1: Identifier of SIP message and points at which session states are copied from P-CSCF and S-CSCF.

Transaction ID	SIP message ID before which updated session states are stored	Corresponding ranges of SIP message ID
1	14,15,17,18	1-18
2	25,26,27,28	19-28
3	35,36,37,38	29-38
4	46,47,48,49	39-49
5	51,52,53,54	45-54
6	62,63,64,65	55-65
7	72,73,74,75	66-75

The procedure is performed by P- and S-CSCF, but not I-CSCF. Regarding I-CSCF, the backup servers do not need to store any session states from the I-CSCF. This is because I-CSCF is only concerned with the completion of each transaction, but not with dialog (the order of transactions) or registration information.

Table 4.1 summarizes the situation where the P-CSCF and S-CSCF store the session states in the backup servers. The center and right-side columns indicate the message identifiers before which the updated session states are stored and corresponding range of SIP messages, respectively. The proposed method can halve the number of times to store the session states, compared with the case where storing the session states is executed whenever the session states are updated.

When the CSCFs receive the request messages: INVITE, PRACK (messages 19-22), UPDATE, 180 Ringing, 200 OK (messages 55-59) and BYE in transactions 1 through 7, the backup servers do not store the session states from the CSCFs. Conversely, when the CSCF receives the response messages: 183 Session Progress, 200 OK (messages 24-27, 34-37, 50-53 and 71-74), and RRACK (message 45-49), the backup servers store the session states. 100 Trying messages indicate to extend the retransmission timer for the INVITE. 100 Trying messages can be treated as request messages. Regarding PRACK (messages 45-48); the CSCFs store the updated session states before processing the response messages of transaction 4.

After the backup servers store the session states at the 200 OK (messages 71-74), they erase the session states some time later. This extra time to erase is for the case when the 200 OK messages are lost. Therefore, the backup servers need to set the timer in order that the session states are erased. Regarding the call termination, at messages 71-74, the CSCF updates the UE's status in the backup server as the registration status.

There may be a very few cases where the replacement CSCF immediately receives the response message without receiving the request message. In this case, the replacement CSCF should silently discard the message and wait for the corresponding request message.

Our method to reduce the number of times to store the session states is based on the UDP retransmission mechanism in SIP. Reliable transport (TCP and SCTP) is also prescribed as the transport protocol in SIP [34]. In this case, the proposed method cannot be adopted because the communication connections cannot be recovered by the replacement CSCFs. However, we presume that the UDP is widely used in production SIP proxy servers, and they can all be candidates to which our restoration system will be adapted.

4.3.6 Prioritized Restoring of Session States

This subsection describes a way to avoid service degradation or disruption when stored session states are recovered to the replacement CSCF. Because some production CSCFs can maintain hundreds of thousands of UEs, the recovery may not be complete within the delay limitation.

We put a priority on the session states when our restoration system restores the CSCFs. We propose that our restoration system first copies the in-progress statuses and restores the in-progress statuses in the replacement CSCF, so that the replacement CSCF handles the retransmitted SIP messages.

Our solution is prioritizing the session states depending on the status of sessions when storing them in the backup servers. The session states are restored in the order of the following priorities:

- *high priority* is given to session states in the call initiation call flow (referred to in-progress status);
- *medium priority* is given to session states in the active status; and
- *low priority* is given to session states in the registration status.

For the in-progress status, the replacement CSCF receives the retransmitted SIP message, and it needs to process the SIP signaling call flow. Because there are only short duration in which to prevent degradation of the service initiation, the in-progress status should be recovered with the highest priority.

Regarding assigning medium priority to the active status, it is most important to complete the service termination call flow when users try to finish an on-going call. In the case where our restoration system restores a CSCF just after BYE messages are sent by UEs, the termination of the application is delayed and incorrect charging may occur. As one of the methods, this dissertation proposes that service operators must recognize how long the CSCF was operating and charge these UEs the communication fee until the CSCF stopped operating.

When the INVITE message first arrives at the replacement CSCF before the migration of the registration information is completed, there is a probability that the replacement CSCF cannot handle that INVITE message. In this case, the replacement CSCF requests the backup server to send the corresponding registration information. The replacement CSCF has a function to request the registration information from the backup servers before the migration of the registration information is completed.

When the active statuses are restored, there is a probability that BYE messages are sent by UEs before the migration of the active statuses is completed. In this case, CSCFs normally return error code messages and applications stop retransmitting the BYE messages and terminate. As a result, service operators cannot accurately obtain accounting information from UEs.

We append a function to ensure that the restored CSCFs discard the BYE message and do not return any error code messages before completing the restoration of all the active statuses so as to solve the issues described above. When UEs retransmit BYE messages after CSCFs are restored, the session is normally terminated by the CSCF taking over the session state.

4.4 Implementation of IMS Restoration System

We show an implementation of our restoration system using OpenIMSCore [35] in general-purpose servers. Figure 4-5 shows the overview of CSCF implementation for cooperating the backup servers. OpenIMSCore is an open source for P-CSCF, and S-CSCF implementation. The implementation of the P-CSCF/S-CSCF/I-CSCF is based on the SIP Express Router (SER)

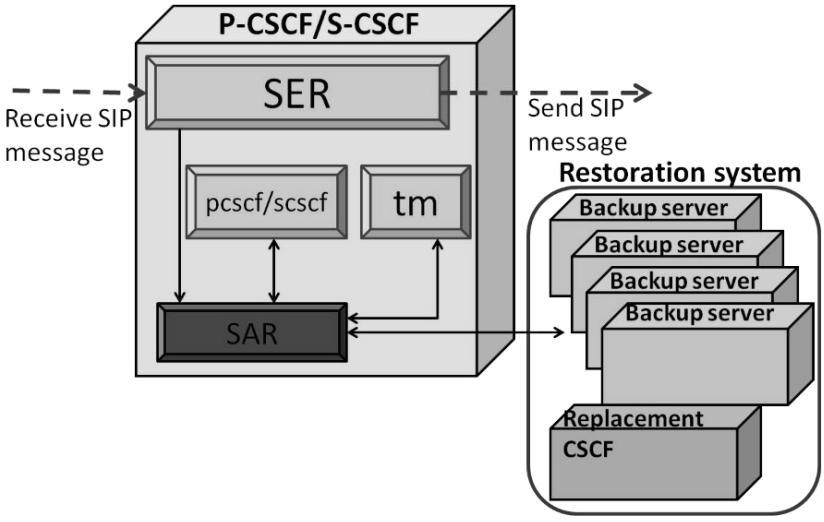


Figure 4.5: Overview of CSCF Implementation for cooperating backup servers.

[36], an open-source SIP proxy server. The SER receives and sends the SIP messages, and transfers the information written in the SIP messages into pcscf, scscf, and icscf program.

The implementation of P-CSCF, S-CSCF, and I-CSCF in OpenIMSCore initially contains five programs (i.e., SER, tm, and pcscf, scscf or icscf). The pcscf, scscf, and icsf program manage the registration information and the dialogs through the SER. The tm program manages the transactions in the SIP signaling call flow through the SER.

We developed a program (termed "storing and restoring [SAR] function") that interacts between the SER, the tm, pcscf, scscf, and icscf programs, and the backup servers. SAR function connects with the SER, pcscf, scscf, icscf program and tm. The SAR function is extended from the original OpenIMSCore for our restoration system. The SAR function stores the session states based on the rule described in Section 4.3.5 and restores the in-process status based on the priority described in Section 4.3.6.

In the proposed method described in Section 4.3.2, the SAR function stores the session states by sending the binary data of the session states to the backup server through the TCP connections. Each backup server stores the session states of P-CSCF and S-CSCF in the database as a unit of binary data.

The SAR function searches for transaction, dialog, and registration information of the in-progress statuses kept by the halted CSCF in the database, when the monitoring function detects a fault in the CSCF. To restore the session states, following eight steps are executed in the implementation:

(1) The replacement CSCF is prepared first of all and the configuration of the halted CSCF is reflected, when a fault is detected.

(2) Our restoration system searches for the transaction, dialog, and registration information of the in-progress status related to the halted CSCF from the database.

(3) The backup server starts transferring the transaction, dialog, and registration information to the replacement CSCF as the in-progress status.

(4) The SAR in the replacement CSCF transfers the dialog and registration information to the pcscf or scscf program, and the transaction to the tm.

(5) The backup server starts transferring the dialog and registration information to the replacement CSCF as the active status.

(6) The SAR function in the replacement CSCF transfers the dialog and registration to the pcscf and scscf program.

(7) The backup server starts transferring the registration information as the registration status.

(8) The SAR function in the replacement CSCF transfers the registration information to pcscf and scscf program.

4.5 Performance Evaluation

4.5.1 Time to Restore In-progress Status

(1) Experimental Environment

We evaluate the time to restore the in-progress status. Additionally, we examine how many the in-progress status can be restored in assumption of the call arrival in the commercial network and the performance of current CSCF product. Considering the time chart in Figure 2, the time to restore the in-progress status need to be less than 3.99 (6.00-2.01) seconds, when the time spent by the storing process in the backup servers is supposed to be minimum.

To evaluate the time to restore the in-progress status, we examine how long the implementation takes to execute steps (2), (3) and (4) described in Section 4 by varying the number of in-progress statuses. We prepare the environment where the backup server keeps the same number of the in-progress statuses as the current CSCF product in the commercial network. The number of in-process status kept by a product of CSCF is examined by referring to the data in the commercial network.

We refer to Telecom Data Book so as to examine how many in-progress statuses are generated. The Telecom Data Book [20] says that the annual average rate of call arrival from an individual user is about 1.7 times per day. Suppose that the offered calls in busy periods are 20 times as frequent as the average call arrival rate (1.7) per day.

The number of in-progress statuses increases in direct proportion to the maximum number of UEs in one CSCF. We refer to the maximum number of UEs in the product of CSCF. There is currently a product of CSCF [21] that can accommodate up to 800,000 UEs. The SPs do not normally accommodate UEs at the maximum. In this experiment, we suppose that each CSCF accommodates UEs up to 60% of its capacity. Then, one P-CSCF accommodates 480,000 UEs at the maximum in number of UEs.

We calculate the number of in-process statuses from the calls received by one CSCF during one second in the busy periods in which the offered calls are 20 times as frequent as the average call arrival rate (1.7) per day. Thus, the number of calls per P-CSCF per second is derived about 188 (480,000 $\times$ 1.7 [call per day] $\times$ 20 / (24 [hours] $\times$ 60 [minutes] $\times$ 60 [seconds])) in the busy periods in average.

To keep the service quality required for the telephone service, the call setup delay needs to become less than 6.00 seconds. We suppose that our restoration system restores the in-progress statuses kept by the CSCF during this 6.00 seconds. Hence, the number of in-progress statuses kept by the backup server for the caller and callee during the busy period becomes about 2256 (188 $\times$ 2 $\times$ 6 [seconds]) at the maximum.

To evaluate the time to restore more in-progress status kept by one CSCF than one in the current commercial network, we vary the number of in-progress statuses in the range from 1128

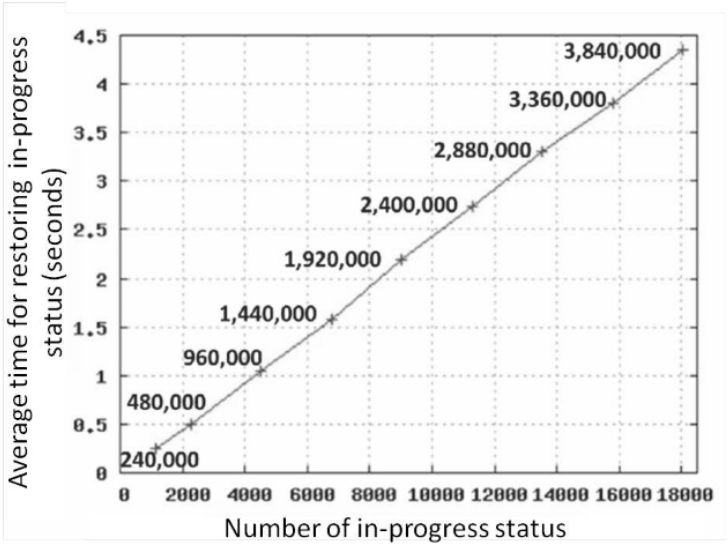


Figure 4.6: Average time for restoring in-progress status.

to 18048. In this experiment, we evaluate whether the CSCF can accommodate 8 times number of UEs more than one in the current commercial network or not.

When the CSCF accommodates 3,840,000 (480,000 × 8) UEs at the maximum in number of UEs, it keeps 18048 (2256 × 8) in-progress statuses. In this experiment, we evaluate the time to restore the in-progress statuses for the cases that the CSCF accommodate 8 times number of UEs more than the current product of CSCF.

We used a PC with an Intel Core 2 Duo E8600 3.33 GHz CPU, 4 GB RAM, and Ubuntu Linux 11.04 as the backup server, database, P-CSCF, and S-CSCF. We used 1 Gigabit Ethernet as the physical network interface in all the servers. We prepared one backup server, one replacement CSCF, and two P-CSCFs and S-CSCFs.

In the experiment, the P-CSCF has the fault, and the in-progress statuses are copied from the backup server to the replacement CSCF. The number of in-progress statuses kept by one P-CSCF is changed based on the assumption explained above. In this situation, the replacement CSCF is turned on beforehand and that step (1) described in Section 4 is completed for a moment, because the volume of the configuration is small. For each value of in-progress statuses, we executed the experiments 50 times.

(2) Experimental result

Figure 7 shows the average time to restore the in-progress statuses in the range from 1128 to 18048. The Y-axis represents the time taken to restore the in-progress statues, measured in seconds. The time almost linearly increases as the in-progress statuses increase. In Figure 7, annotated numbers with dots indicates the maximum number of UEs which can be accommodated by one CSCF derived from the number of in-progress statuses (X-axis). The annotated numbers with dots are calculated based on the data where CSCFs accommodating 480,000 UEs keep 2256 in-progress statues.

From Figure 7, our implementation restores 2256 in-progress statuses which one CSCF is assumed to keep in the current commercial network within 0.50 seconds. Within 3.99 seconds, our implementation restores 15792 in-progress statuses. Our implementation can restore the in-progress status of the CSCF keeping 7 times more than in the current commercial network.

Our implementation keeps the service quality required for the telephone services, even if the CSCF in the future accommodates 7 times more number of UEs than the current product of CSCF. The SIP signaling call flow can be completed within 3.99 seconds, even if the fault occurred in the CSCF. Namely, our implementation has sufficient performance regarding the time to restore the in-progress status, when we do not care of the storing process in the backup server. Next, we investigate how many backup servers are required by evaluating the time taken by the storing process in the backup servers.

4.5.2 System Time for Storing Session States

(1) Simulation Model

In this subsection, we perform two evaluations. First, we investigate the required number of backup servers by varying the write speed to the disk in the backup servers. In the first evaluation, the SP accommodates 50 million UEs in the network. Second, we investigate the number of backup servers by varying the number of UEs accommodated by the SP.

The term "system time" corresponds to the time during which CSCFs and UEs process the SIP messages and the backup servers receive the session states from CSCFs and write the data of the session states to the disk. When the system time is prolonged, it also significantly lengthens the call setup delay. Then, the service quality is degraded. The maximum system time from all the call arrivals is investigated by using a time-based simulator. The maximum system time is exploited as a maximum value among all values of the system time during one hour in the simulation.

In this simulation, each backup server treats the storing process in a FIFO manner and has infinite buffer space to receive the request messages for storing. The SIP signaling call flow is proceeded and the request messages for storing the session states are transferred into the backup servers, when the time passed in the simulation. The backup server sequentially stores the session states by writing the data in the database. The CSCFs are equally connected to each backup server for storing the session states. Then, the request messages for storing the session states are equally distributed into each backup server. UEs and CSCFs take 1 millisecond to process the SIP messages, and do not experience any congestion when processing the SIP messages.

To investigate the effect of the proposed method (termed proposal), we compare it to the case where the backup servers store the session states every time CSCFs receives SIP messages in the SIP signaling call flow. Hereafter, we term this naive method "allcopy". This is the same as in the previous studies [8, 9].

In the simulations, we adopt the same average of the call arrival rate and CSCFs in Section 5.1. We assume that the call arrival follows Poisson distribution. We adopt 120 seconds as the call duration. Based on the result of Section 5.1.2, our restoration system takes 0.50 seconds to restore the in-progress statuses kept by the current CSCF product in the commercial network. Considering the time chart in Figure 2 and the result in Section 5.1.2, we adopt 3.49 (6.00 - 2.01 - 0.50) seconds as the criterion for the permissible call setup delay which includes the process for storing the session states.

Based on the criterion (3.49 seconds), and the propagation delay of messages, we calculate the criterion of the maximum system time in the proposal and allcopy below. The propagation delays in the simulation are shown in Table 2. The propagation delay in the wireless link between the P-CSCF and UEs is 0.05 seconds according to the literature [22], where Long Term Evolution (LTE) [23] is adopted as the wireless link. The signaling messages are normally categorized as the traffic having the highest priority. Hence, the value of propagation delay in the network

Table 4.2: Propagation Delay of Messages.

Network	Value(milliseconds)
Between UE and P-CSCF	50
Among the CSCFs (P-CSCF1 and S-CSCF1,S-CSCF1 and S-CSCF2, S-CSCF1 and I-CSCF,I-CSCF and S-CSCF2, S-CSCF2 and P-CSCF2)	2
Between backup servers and CSCFs	2

does not vary largely, and we adopt the static value (0.05 seconds) as the propagation delay of the wireless link.

The propagation delay of messages are calculated below. In the proposal, the session states are stored for 12 repetitions as drawn in Figure 4 until 180 Ringing messages are received by UE-A. From Table 2, the duration of the propagation delay for sending the session states and acknowledging the messages in the proposal excluding the system time becomes 48 (12 [number of times to store session states] $\times$ 2 [two messages] $\times$ 2 [milliseconds]) milliseconds. We term this duration of the propagation delay "storing message delay". In the same way, the storing message delay in the proposal is computed to be 148 (37 [number of times to store session states] $\times$ 2 [two messages] $\times$ 2 [milliseconds]) milliseconds.

We calculate the call setup delay excluding the system time in the proposal and allcopy. The SIP messages, i.e., message 1, 2, 11, 13, 18, 19, 23, 24, 28, 29, 33, 39, and 44 in Figure 4 are exchanged through the wireless links 13 times (total 650 milliseconds). When UE receives the message 33 (UPDATE message), UE is supposed to send the message 39 (180 Ringing message) in 20 milliseconds. 25 SIP messages are transferred among the CSCFs (total 50 milliseconds) until 180 Ringing message arrives at UE. Because the storing message delay becomes 48 milliseconds in the case of the proposal, the call setup delay excluding the system time becomes 768 (650 + 50 + 48 + 20). Similarly, in the case of allcopy, the call setup delay excluding the system time in the backup server also becomes 868 (650 + 50 + 148 + 20) milliseconds.

For the criterion (3.49 seconds), we subtract those value calculated above. Then, the maximum system time in the backup server must be less than 2,722 (3490 - 768) milliseconds in the proposal, and 2,622 (3490 - 868) milliseconds in the allcopy.

We use 1,000, 750, 500 and 250 (Mbps) as the write speed for writing the data in the backup server from solid-state drive [24] currently available as the product. We use the data size of transaction and dialog from the OpenIMSCore. The data size of the transaction and dialog are 3,791 and 1,0133 bytes, respectively. The backup server stores the session states (transaction and dialog) in each write speed.

(2) Simulation Result

We denote the write speed as s (Mbps) for writing the session states in the backup server, and the number of backup servers as b. By changing the number of UEs, we investigat the required number of backup servers b.

Table 3 shows the maximum system times that are closest to 2,722 and 2,622 milliseconds in the simulation when we adopt write speed s = 1,000, 750, 500, 250 (Mbps). As shown in Table 3, in the case of write speed s = 1,000, 750, 500 and 250, the number of backup servers b in the proposal is less than 38% of the allcopy.

Figure 8 summarizes the required number of backup servers based on the write speed in

Table 4.3: Maximum system time during an hour in the simulation when we adopt write speed $s = 1000, 750, 500, 250$ Mbps

Write Speed (Mbps)	Type	Number of backup servers (b)	Maximum system time(milliseconds)
250	Proposal	114	2,684
250	Allcopy	309	2,619
500	Proposal	57	2,684
500	Allcopy	154	2,622
750	Proposal	39	2,670
750	Allcopy	105	2,611
1,000	Proposal	29	2,583
1,000	Allcopy	177	2,602

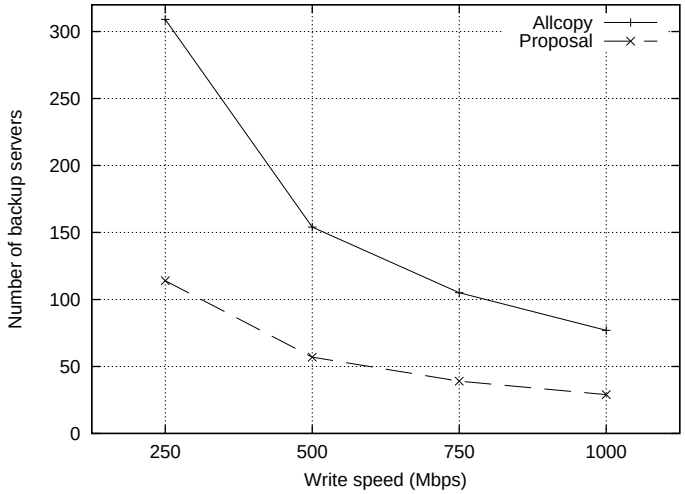


Figure 4.7: Required number of backup servers based on write speed in case that number of UEs is 50 million.

the case that the number of UEs is 50 million. As drawn in Figure 8, the required number of backup servers decreases in according with the increase of the write speed, and the write speed does not generate the severe bottleneck. Even if the write speed is slow, the number of backup servers is required to be more than the assumption.

Figure 9 shows the number of backup servers b based on the number of UEs accommodated by the SPs, in the case of write speed $s = 500$ (Mbps). The number of UEs is varied from 25 million to 100 million. The results indicate that when the number of backup servers increases, it is desired that the SPs prepare the backup servers in direct proportion to the number of UEs so as to satisfy the criterion of the maximum system time. The SPs need to prepare the backup servers in direct proportion to the number of UEs.

Next, we note the increasing rate of the maximum system time based on the number of backup servers. Table 4 shows the relationship between the descreasing rate of the number of backup servers and the increasing rate of the maximum system time. We investigate the maximum system time by changing the number of backup servers in the range from 58 to 52.

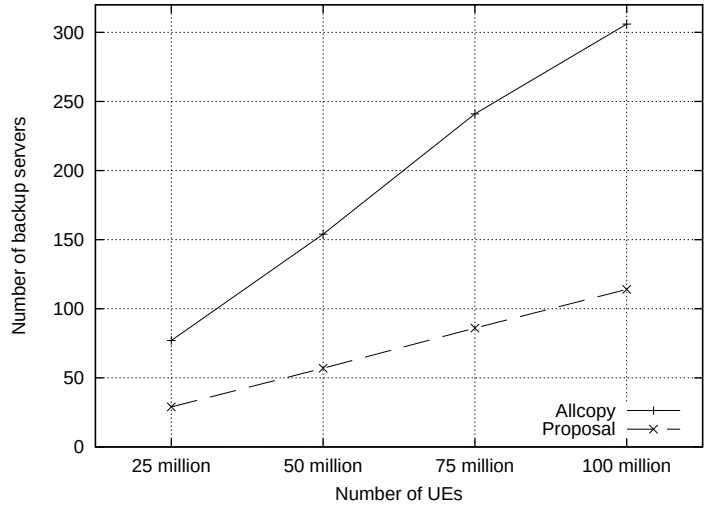


Figure 4.8: Required number of backup servers based on number of UEs in case of write speed $s = 500$(Mbps).

Table 4.4: Relationship between descreasing rate of number of backup servers and increasing rate of maximum system time

Number of backup servers(1)	Maximum system time(2)	Decreasing rate of (1)	Increasing rate of (2)
58	2,586	-	-
57	2,684	1.75%	3.65%
56	2,817	1.79%	4.72%
55	2,922	1.82%	3.59%
54	3,047	1.85%	4.10%
53	3,170	1.89%	3.88%
52	3,319	1.92%	4.49%

When the number of backup servers b is 58 in the case of accommodating 50 million UEs and adopting 500 (Mbps) as write speed s, the maximum system time in the proposal becomes 2,586 milliseconds. The maximum system time in the proposal becomes 2,684 milliseconds, when the number of backup servers b is 57 in the same environment. Then, the maximum system time increases by 98 milliseconds, when the number of backup servers b changes from 58 to 57. This means that the decreasing rate of the number of backup servers is 1.75% (1/57) and the increasing rate of the maximum system time is 3.65% (98/2684).

In the same way, we calculate the decreasing rate of the number of backup servers and the increasing rate of the maximum system time as shown in Table 4. Single reduction of the number of backup servers generates almost same increasing rate (about 4%) of the maximum system time. This indicates that the load of the backup servers is distributed and there is no serious bottleneck in the process for storing the session states. We can lower the maximum system time in proportion to the number of backup servers.

Regarding the service initiation and termination call flow, the proposed method reduces the number of backup servers to less than 38% compared to the previous studies [8, 9]. Including

the registration call flow, we can obtain the larger effect. For the registration call flow, it is sufficient that the proposed method execute storing the session states only once i.e., after the registration call flow is completed. Therefore, compared to the case of allcopy, the proposed method requires a much lower number of backup servers than in the previous studies.

4.6 Summary

This dissertation proposed the IMS restoration system with selective storing of session states. The proposed system reduced the number of times to store the session states by using the retransmission mechanism of SIP. Additionally, it prioritized restoring specific session states to avoid disrupting the SIP signaling call flow of UEs.

This dissertation evaluated the performance of the proposed system applied for the telephone service. The time to restore the in-progress status was evaluated by implementing it on a general-purpose server. The implementation restored the in-progress statuses kept by the current product of CSCF in the commercial network within 0.50 seconds.

The required number of backup servers was evaluated in the time-based simulation by adopting the call setup delay as the criterion. The simulation result showed that the proposed system reduced the number of backup servers to less than 38%, compared to the previous studies using the $n+k$ redundancy model. Although the proposed method can archive such a large reduction, it does not degrade the service quality. Even if faults occurred, the call setup delay did not exceed 6.00 seconds. Consequently, the service operator can reduce the facility cost in the $n+k$ redundancy model. The proposed method to backup the session states focuses on the CSCF in the IMS-based services. Because the algorithm to decide the points where the session states are stored is based on the basic retransmission pattern of the signaling messages, the proposed method can be easily applied for the other signaling protocol.

Chapter 5

Proposal of Traffic Engineering by Utilizing Signaling Protocol

5.1 Background

As the network architecture to realize the traffic engineering, this dissertation proposes the traffic management cooperating with IMS in MPLS networks. The procedures in the architecture focuses on the IMS as the signaling protocol. However, the procedures and algorithm for selecting the paths in MPLS networks do not utilize the particular function of the SIP. Therefore,even if the protocol is updated or changed, they can be utilized as it was.

Many service operators are now promoting convergence towards NGN (next generation network) [41] architecture, in anticipation of cost-effective synergy between legacy and Internet services. NSPs design and construct the NGN to provide various services on single network infrastructure. These services also have various QoS requirements, for example, VoIP traffic should be guaranteed, some transaction or signaling/control traffic may be delay-sensitive, and the Internet traffic can be best-effort.

Nowadays, NSPs are considering more traffic accommodation in the transport stratum to provide the network resources for various services. However, particular applications consume more bandwidth than before, and the traffic requirements of particular customers occupy most of the bandwidth. NSPs need to manage the traffic in the consideration that the traffic of particular customers becomes dominated and holds down the traffic of the other customers

In NGN architecture, IMS [42] is a key technology, where CSCF (call/session control function) [43] is responsible for establishing the session using SIP (session initiation protocol) [44] before the communication of users. NSPs can gain the QoS demand (e.g., bandwidth and delay) of each application before data transmission through the SIP signaling messages exchanged between users and CSCF. Such demand is transferred to the policy control server (RACF: Resource and admission control functions) [45] in order to determine whether the session can be accepted or not. However, IMS itself does not specify the transport stratum issues, (e.g., how to realize QoS in the transport stratum)

On the other hand, many NSPs have introduced MPLS [46] in their transport stratum to realize flexible traffic engineering, by setting up logical circuits (LSP: Label switched path [47]) between the pairs of edge routers reflecting various constraints and the operator's policy. In addition, NSPs could collect the traffic statistics per LSP directly related to the pair of edge routers by utilizing the MIB (Management Information Base). This statistics is convenient to enable RACF's call admission control to realize more precise traffic management.

This dissertation studies the traffic management for the transport stratum by utilizing the

function of IMS. Our research goal is to provide a stable communication environment to customers and to achieve the fairness and maximum of the traffic accommodation between the pairs of edge routers by utilizing IMS as the service stratum. We propose feasible LSP selection and extension of IMS function to achieve this goal.

5.2 Issue of QoS control in MPLS and IMS Architecture

5.2.1 MPLS Traffic Engineering

MPLS networks are composed of edge and core routers. Packets are transferred along one of the LSPs which are established between the ingress and egress edge routers. Once a packet enters the MPLS networks and is transferred into the LSP, the core routers transfer the packet along the LSP. The mapped label value for LSP is delivered by RSVP (Resource Reservation Protocol) [48] to the adjacent routers. The adjacent routers also deliver the mapped label for the LSP to their adjacent routers. In this way, the label delivery is conducted in a hop-by-hop manner.

MPLS traffic engineering provides benefits over IP network, that is to say, achieving the flexible control of the traffic in the transport stratum. The LSPs can either be routed explicitly (manually), or dynamically routed by the CSPF (constrained shortest path first) algorithm.

DS-TE (DiffServ-Aware MPLS traffic engineering) [49] has been standardized and implemented [50] [51] as one of MPLS traffic engineering methods. Based on the CSPF, this function not only automatically adjusts the LSP bandwidth but also dynamically reroutes the LSPs, when a certain physical link on the current LSP routes becomes short of capacity.

To execute the function of DS-TE, the routers need to have high functionality as each LSP needs to monitor the traffic statistics and compute the suitable bandwidth. To guarantee Qos, the routers need to have traffic shaping function in each LSP. Because the routers generally have many LSPs, such functions in each LSP generates large load on the routers.

Moreover, the function of DS-TE changes the end-to-end delay of certain media traffic because of the sudden rerouting. The operators normally want to ascertain the route. The sudden rerouting of LSP gives much load to the operators in NSPs when they manage the LSP.

The CSPF is a "greedy" algorithm that adopts the route which has sufficient capacity one at a time with no reference to any other LSPs which treat with the same traffic class being placed. Therefore, the LSP from the particular edge router sometimes occupies much capacity ahead. It generates the unfair capacity assignment for the LSPs between all the pairs of edge routers. Based on these issues, we assume that it is difficult for NSPs to adopt the function of DS-TE in their MPLS networks.

From this discussion, it is clearly desirable that the edge routers have LSPs statically configured for the traffic class, and that the edge routers determine the route and distribute the traffic over their networks. And the traffic accommodated in the network should be taken into account for admission control (i.e, acceptance, rejection or which route the traffic should be transferred).

5.2.2 Session Control Procedure in IMS

The procedure for call/session establishment of IMS was standardized in ITU-T. SIP signaling originated by a user is sent to the CSCF, and the CSCF responds the acceptance or reject of the session of the user after obtaining the decision from the RACF

Fig. 5-1 abstracts the functional view of the service operator network using IMS in MPLS networks. The transport stratum is composed of the access networks, the AGWs (access gate-

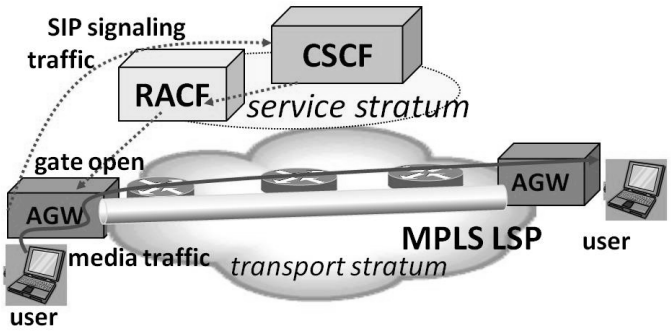


Figure 5.1: Functional structure using IMS in MPLS networks

ways), and transport (core) network. The AGW is located between the access and core network and enforces QoS control for the user-data traffic (termed 'media traffic'). Gate opening/closing and marking the media traffic with the determined priority level. Packet marking is done by setting the bit value in the DSCP (diffserv code point) [52] or TOS (type of service) field in the IP header. IMS provides session demands from uses for the service operator before the beginning of their communications. Such demands are useful to determine the target LSP for the communications; however, the following items should be considered for the traffic management cooperating with IMS in MPLS networks:

1. deploying cooperative session control procedures between CSCF, RACF, and AGWs,
2. recognizing the resource utilization of the transport stratum, and
3. an admission control method to deal with multiple traffic classes

As for item 1, the architecture of the RACF for MPLS which utilizes DS-TE has already been proposed in ITU-T [53] [54]. It is difficult for the operators of NSPs to manage this architecture because of utilizing the function of DS-TE. In this architecture, there is no policy to transfer the traffic for improvement because it maps the same traffic class to the one LSP In MPLS network, we can collect LSP-based traffic statistic by using the standardized MIB [55]. For item 2, the LSP traffic statistics concerning resource utilization in the core network are useful. For item 3, the method should take into account fairness and maximum of the traffic accommodation describe in Section 4.

5.3 Design for Proposed Traffic Management

5.3.1 Definition of Traffic Class for Traffic Management

In this dissertation, we adopt at least three traffic classes to realize the traffic management, as described in Table 5.1. Additional traffic classes for more fine-grained treatment may be defined in certain NSPs. For example, the traffic of streaming applications requires lower delay variation. Our proposed traffic method can be extended and easily applicable in such NSPs

As the primary-class traffic, we assume the signaling traffic which requires the minimum delay, while the standard class is for media traffic requiring sufficient bandwidth. Additionally, we consider the best-effort traffic (e.g., Internet access) without QoS requirement.

There are multiple SIP signaling messages exchanged in establishing the session. The individual SIP signaling message goes sequentially back and forth between users and CSCF. This

Table 5.1: Traffic class for proposed traffic management

Priority level	Traffic class	Traffic treatment
High priority	Primary class	Delay restriction required - shortest or sufficiently small delay routes
	Standard class	Loss sensitive - traffic distribution over multiple routes
Low priority	Best effort	No guarantee - transferred into the shortest route if there is capacity

implies that even if the one-way delay for signaling message takes a few milliseconds, the completion of signaling message takes several times longer than sending the single message. Although the maximum domestic transmission delay (e.g., peaking at 10 milliseconds in Japan) may have little impact on the media traffic (application), but the round-trip time for exchanging the signaling messages is not small. Therefore, NSPs have to take these effects into account to minimize the signaling duration when designing and operating their networks. The signaling message should be treated as the primary class

We consider that the total of primary-class and standard-class traffic which are defined as high priority have the threshold in each physical link, while the remaining capacity can be allocated for the best-effort class which is defined as low priority. This threshold (termed "acceptable capacity") is defined for each physical link. For the accommodation of the traffic, we consider that the operators should design the acceptable capacity which is less than the demands of primary-class traffic for all the physical links. In our design, the routers have PQ (Priority Queue) and prioritize the traffic of the high priority. Therefore, when the amount of the traffic in one physical link is beyond the acceptable capacity, the best-effort traffic may be dropped.

The total demand for the primary-class and standard-class traffic may exceed the acceptable capacity. In such situations, by using multiple (more than two) LSPs, the standard class traffic is distributed over multiple LSPs in the proposed method. The traffic of signaling message has strong requirement for the minimum delay, so the primary-class traffic should be transferred the shortest route between the pairs of edge routers. For the standard class, we assume the capacity requirement to be stronger than that of the delay. When the packet loss does not occur, there is no negative impact even if the standard-class traffic is distributed over multiple LSPs and gets little additional delay.

5.3.2 MPLS LSP Configuration

Fig. 5-2 shows the example of the proposed network architecture. The AGW has the same function with MPLS edge routers and executes packet marking of the traffic from the users. In typical MPLS network operation, single LSP with the shortest routes is established for all the pairs of (ingress and egress) edge routers. In our proposed method, we propose setting up multiple LSPs between all the pairs of edge router. Our LSP configuration does not impose an additional load on the routers because the routers need not have functions like DS-TE.

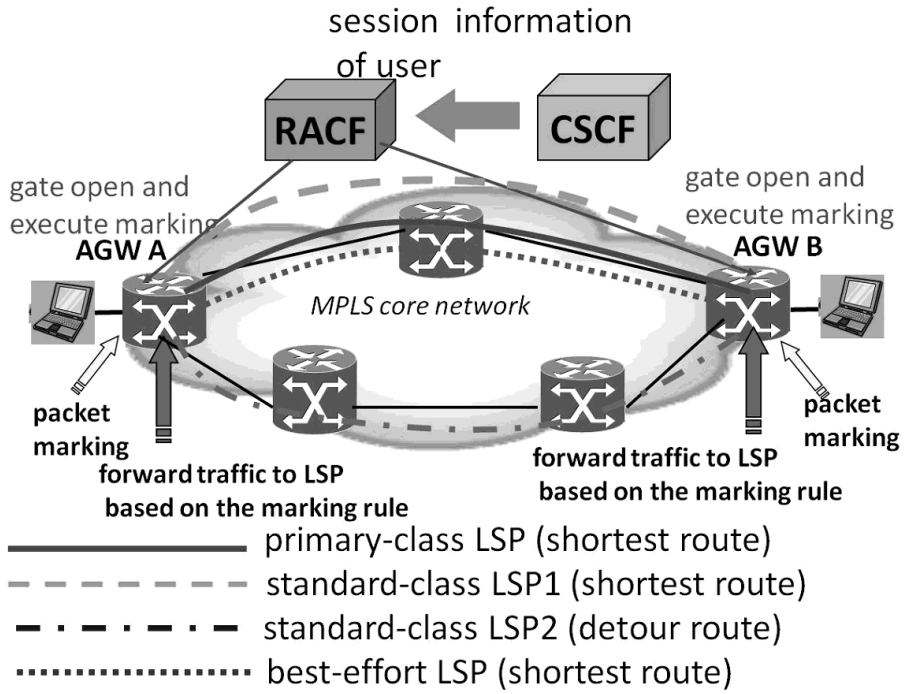


Figure 5.2: Example of proposed network architecture

In Fig. 5-2, three LSPs along the shortest route (primary-class LSP, standard-class LSP1 and best -effort LSP) and one LSP along the detour route (standard-class LSP2) are established as the example of the proposed method between AGW A and B. One of the shortest LSPs and the detour LSP are assigned to the standard-class traffic. In this example, the standard-class traffic is distributed over the two LSPs.

The traffic utilization for each LSP can be collected with SNMP. Gathering statistics on the traffic of multiple LSPs between all the pairs of edge routers, we can get total amount of the traffic in the core network. By setting up LSP, we can acquire such traffic utilization in the detail. This allows the RACF to acquire the traffic utilization for each LSP, and the RACF can utilize it for admission control.

5.4 Proposed Method

5.4.1 Capacity Assignment

To achieve much traffic accommodation in the core network, it is necessary to consider how to accept the traffic from users and to distribute the traffic over multiple LSPs. When it is congested in some physical links, the RACF needs to have the capacity assignment policy for the standard-class traffic. In my proposed method, the primary-class traffic is transferred into the shortest route without any admission control because it degrades the quality of service with the additional delay. We consider the capacity assignment policy which not only maximizes the traffic accommodation but also accommodates the standard-class traffic fairly between all the pairs of edge routers. The service operator also considers the traffic of particular customers should not occupy in the core network

In this dissertation, the fair accommodation means that certain level of capacity assignment

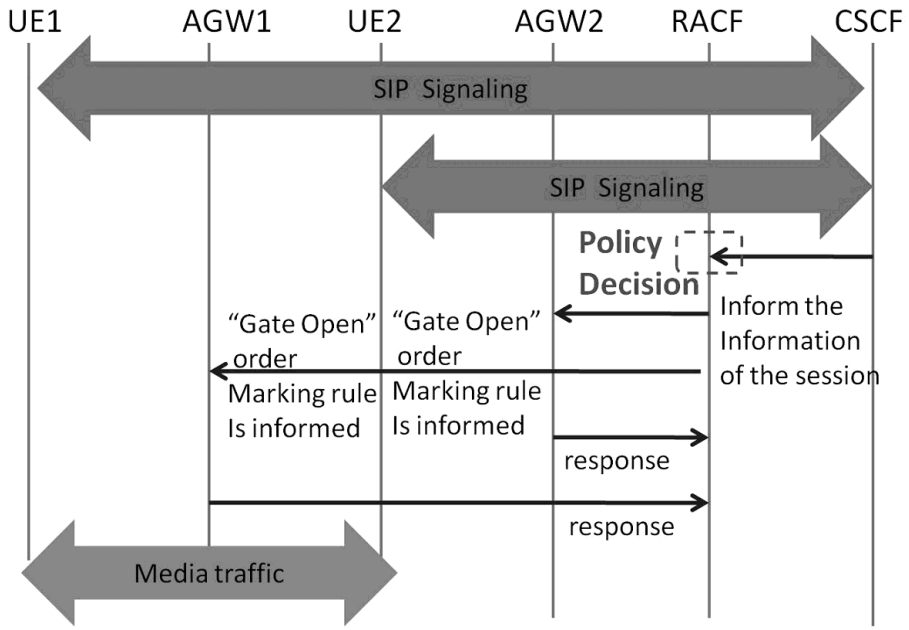


Figure 5.3: Session initiation procedures in IMS

is guaranteed between any pair of edge routers in NSPs. This allows customers under any edge routers to establish the session at some level. Considering the actual network in NSPs, the amount of the traffic is not equal between all the pairs of edge routers. Applying the idea of the fair accommodation into the actual network, it is necessary to put weight value onto the amount of the traffic between all the pairs of edge routers. To simplify the explanation in the rest of the dissertation, we assume that the amount of the traffic is equal between all the pairs of edge routers.

To accommodate more standard-class traffic, while considering the fair accommodation between the pairs of edge routers, This dissertation proposes dual-phase capacity assignments. In the first phase, an identical capacity is assigned to the LSPs between all the pairs of edge routers. In the second phase, this dissertation adopts the strategy of maximizing the traffic accommodation between all the pairs of edge routers by greedily assigning the remaining capacity in the physical link, based on the requests of the users.

5.4.2 Session Initiation Procedures

Fig. 5-3 illustrates the session initiation procedure in IMS standards. The AGWs execute packet marking for QoS control. The CSCF reports the demands of users to the RACF. Then the RACF requests the AGWs to open the gate ("Gate Open") for the traffic of the UEs, before they begin the communication through the AGWs. In the proposed method, session initiation procedures in Fig. 5-3 are expanded to for the session initiation procedures of the proposed method. The traffic management cooperating with IMS in MPLS networks means that the RACF determines the selection of LSP or the acceptance of the call for the standard-class traffic. Based on the policy decision in the RACF, the AGW executes packet marking for the media traffic.

By utilizing the information which IMS function provides, the AGWs acquire the traffic class of the packets from the UEs before the media traffic arrives at the AGWs. The AGWs are

informed of the marking rule from the RACF. We use the basic MPLS function, and need not expand the function of the MPLS edge router in the AGWs. For the media traffic, the behavior of AGW for each traffic class is as follows:

- If the media traffic is identified as primary class, it is executed packet marking by the AGW and transferred into the primary-class LSP which traverse the shortest route.
- If the media traffic is identified as standard class, the RACF decides which LSP among the multiple LSPs between the pair of edge routers it should be transferred into. Then the AGW executes the packet marking for it and transfers it into the suitable LSP.
- If the media traffic which AGW is not reported by the RACF arrives at the AGW, it is transferred into best-effort class LSP. In the physical link, the primary-class and standard-class traffic is prioritized by setting PQ in the physical link of the routers. If the physical link is congested, the best-effort traffic will be dropped at first.

To prevent the packet loss for the traffic of the primary and standard classes, the session initiation procedures for the next call from users is performed. In the RACF, the utilization of the physical links is regularly monitored and referred to decide whether the call of the standard-class traffic from the UEs is accepted or not, depending on the LSP into which the standard-class traffic is transferred. The monitoring is conducted and the utilization of all the physical links is computed. We extend the standard session initiation procedure of IMS for the proposed traffic management, which is as follows:

1. The UE initiates the procedure with the CSCF to establish SIP session.
2. The CSCF queries the RACF to determine the LSP in the core network through which the media traffic of the caller and the callee is transferred. Here, various parameters are informed from the RACF, (e.g., application type, IP addresses and port numbers).
3. The RACF distinguishes the media traffic from the parameter into one of the traffic classes and determines whether the media traffic can be accepted and which LSP it can be transferred. As for the standard-class traffic, the RACF refers to the utilization of the physical links along the LSPs assigned to the media traffic.
4. Incoming media traffic is rejected if the total traffic of the primary-class and standard-class in one physical link where the standard-class LSPs traverse has reached the acceptable capacity.
5. The RACF responds to the CSCF if the LSP is determined.
6. The CSCF report the establishment of the bearer to the users.
7. The RACF sets up the AGWs of the caller and the callee to open the gate for the media traffic. Then the AGW executes the packet marking.

This dissertation assumes that the RACF and the AGW have common marking rule between the bit value (DSCP or TOS bit values) and the corresponding LSP.

5.4.3 Admission Control Procedures

Based on the computation of the capacity assignment for the network, the RACF determines which standard-class LSP the AGW should transfer the media traffic into and whether it is accepted or not as the admission control procedures of the proposed method. The RACF

calculates the capacity assignment for all the physical links beforehand every interval (e.g., 1 or 3 minutes). The utilization of the physical links and LSPs which the RACF collects by SNMP is updated at specific time intervals. The capacity assignment for the standard-class LSPs between all the pairs of edge routers is also updated at this interval.

The RACF checks whether the total traffic of the primary-class and standard-class traffic is more than the acceptable capacity or not in each physical link. If the total traffic of the primary-class and standard-class in even one physical link is more than the acceptable capacity in the routes of one LSP, the standard-class traffic of the call which arrives newly is distributed to the other LSP until the total traffic of the primary-class and standard-class traffic becomes less than the acceptable capacity. If there is no standard-class LSP which has the physical links in which the total traffic of the primary-class and standard-class traffic is less than the acceptable capacity, the calls which arrive newly are rejected by the RACF. The RACF responds to the CSCF and the CSCF sends "Cancel" message by SIP signaling to the users when the media traffic (e.g., a VoIP service using bidirectional traffic) is rejected.

5.5 Evaluation of Proposed Capacity Assignment

5.5.1 Modeling of Dual-phase Capacity Assignment

This dissertation adopts LP (linear programming) approach to make the model for the dual-phase capacity assignment in our proposed method. This dissertation assumes that the session demand from users becomes real number to simplify the LP computation. This dissertation considers the model for dual-phase capacity assignment for the standard-class traffic: the first assignment maximizes the minimum capacity between all the pairs of edge routers, with achieving the fair accommodation for all the standard-class traffic at minimum level; and the second assignment maximizes the total capacity for the remaining capacity in the core network.

Maximizing the minimum capacity in the first phase assignment is computed by solving LP, which maximizes the identical capacity assigned to all the pairs of edge routers. We define the following objective function for the first phase assignment:

$$C = d_k = \sum_i d_{k,i}$$

where d_k and $d_{k,i}$ denote the assigned total capacity of the edge router pair k, and the capacity of LSP i for edge router pair k. Here, we define the following constraint conditions for the above objective function :

$$\sum_k \sum_i x_{e,k,i} - u_e \leq 0$$

for $e \in E$, where where $x_{e,k,i}$ and u_e denote the assigned capacity of LSP i for the edge router pair k in the physical link e, and the available capacity for the standard class traffic in the physical link e. Although identical capacities are assigned to each pair of edge routers, this capacity can be distributed via multiple LSPs between some pairs of edge routers.

The second phase assignment is also computed by solving the LP, which maximizes the total capacity as it follows:

$$C = \sum_k \sum_i f_{k,i}$$

under the constraint conditions

$$\sum_k \sum_i x_{e,k,i} + \sum_k \sum_i x_{e,k,i} - u_e \leq 0$$

for $e \in E$, where $f_{k,i}$ and $y_{e,k,i}$ denote the additional assigned capacity of LSP i for the edge router pair k, and that in one physical link e.

5.5.2 Evaluation Method

The characteristics of the proposed method is that multiple LSPs between all the pairs of edge routers are set and that the RACF has the specific admission control for the capacity assignment. In this section, we evaluate how much capacity the RACF admits the capacity for the LSPs between the pairs of edge routers. Based on the model of capacity assignment which is defined in the previous subsection, we perform the simulation and compare the four capacity assignments by varying the number of the routers and the LSPs between all the pairs of edge routers. The first is the dual-phase capacity assignment with single LSP for the standard class traffic between all the pairs of edge routers. This is termed "1-path max-min". The second is the dual-phase capacity assignment with two LSPs (termed "2-path max-min") between all the pairs of edge routers. The third is the dual-phase capacity assignment with three LSPs between all the pairs of edge routers (termed "3-path max-min"). For the fourth, only the second phase of the dual-phase capacity assignment is applied without any fair accommodation consideration. It has two LSPs between all the pairs of edge routers, and greedily assigns the capacity (termed "2-path shortest-first"). We assume that 2-path shortest-first assignment is same as setting up two LSP between the edge routers by adopting the standardized DS-TE.

To emulate the topology of the carrier network, we use BRITE (Boston University Representative Internet Topology) [56]. BRITE is a tool for emulating network topology in an AS (autonomous system). BRITE provides the BA (Barabasi-Albert) model [57], which is often used to emulate the network topology. As the parameter of BRITE, we set the number of routers and the degree, (the number of physical links per individual router) and generate the network topologies for the simulation.

In generating the network topology, we defined 80% of the routers as the edge routers, and 20% as the core (transit) routers. This reflects the situation in the actual MPLS networks of service operator, where there are many edge routers at the head and tail ends of the LSPs, and a smaller number of core routers, which transit the traffic for the edge routers by switching the LSP. In the simulation, the edge routers have a degree of at least "2" since the edge router normally has two physical interfaces to connect the core (upper) network in service operator for redundancy. The core routers have a degree of over three for connecting to the edge routers and redundancy. Using "1" as the lowest degree, the effect of utilizing multiple LSPs will not be large.

This dissertation evaluates how much capacity is admitted between the pairs of edge routers, based on each capacity assignment. We assume that all the physical links in the generated network topology have identical link capacities. In addition, the shortest LSP and detour LSPs (when multiple LSPs are used) are computed for all the pairs of edge routers. The four capacity assignments are compared in the same network topology provided by BRITE, while BRITE also varies the topology at random as it generates. 20 simulations were executed, and the average was treated as the result. Considering the actual network operation, this dissertation evaluates each capacity assignment

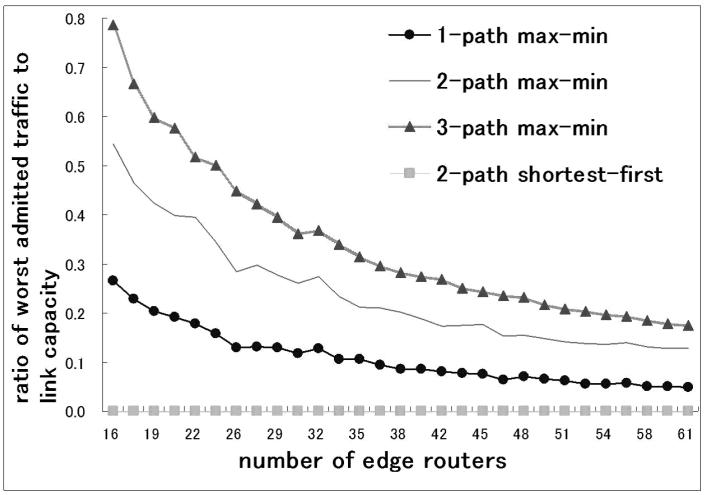


Figure 5.4: Average admitted capacity between the pairs of edge routers

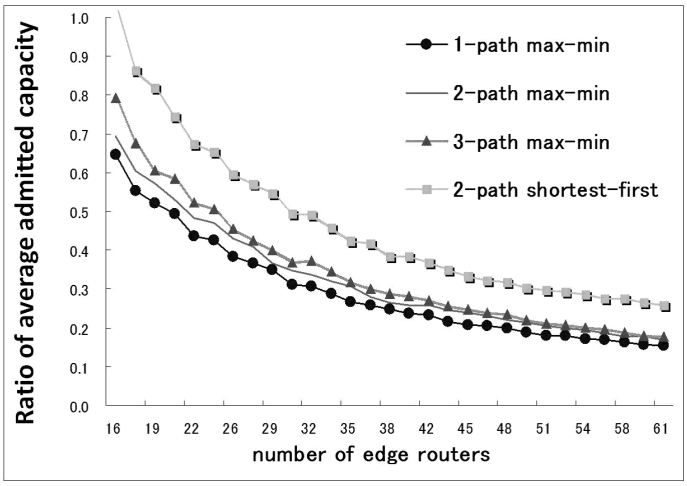


Figure 5.5: Lowest admitted capacity between the pairs of edge routers

5.5.3 Evaluation Result

The results of the simulations are shown in Fig.s 5-4, 5-5, 5-6 and 5-7. Fig.s 5-4 and 5-5 show the average and lowest value of admitted capacity between all the pairs of edge routers based on the proposed capacity assignment policy. The X-axis is the number of edge routers, while the Y-axis is the average or lowest value of admitted capacity which is computed by the RACF for standard-class traffic between one pair of edge routers. To evaluate the fairness and total accommodation of the traffic, we treat with the average and lowest value of admitted capacity. The admitted capacity means the bandwidth value which is assigned to transfer the traffic into all the standard-class LSPs between one pair of edge routers. The value at Y-axis is normalized by the value of the physical link bandwidth.

The average ratio of the admitted capacity in Fig. 5-4 shows that 2-path shortest-first assignment achieves the largest traffic accommodation. Compared with 2-path max-min assignment and 2-path shortest-first assignment, the average ratio of admitted capacity for 2-path max-min assignment is more than 80% of that for 2-path shortest-first assignment.

In the case of the ratio of lowest admitted capacity (Fig. 5-5), 3-path max-min assignment

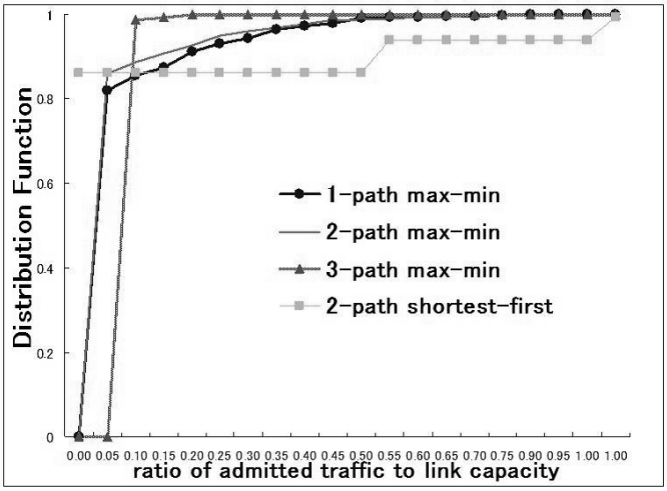


Figure 5.6: Cumulative distribution function in case of 20 edge routers

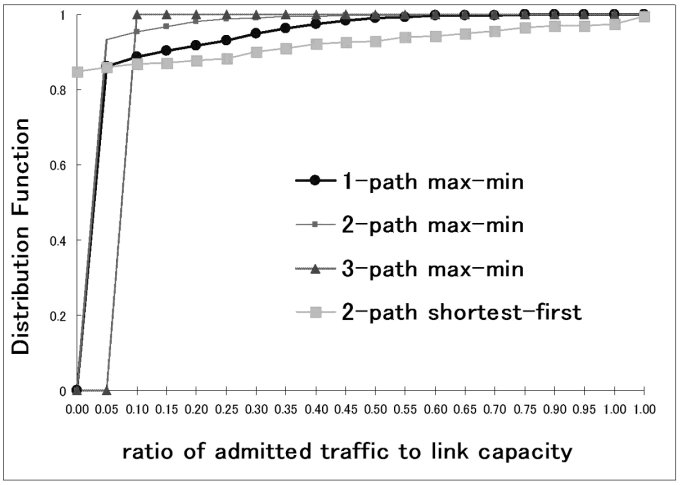


Figure 5.7: Cumulative distribution function in case of 40 edge routers

achieves the largest value. 2-path max-min assignment achieves about twice times as the admitted capacity as 1-path max-min does. For 3-path max-min assignment, the admitted capacity is 1.2-1.3 times larger than with 2-path max-min assignment. And as the number of edge routers increases, the difference between 3-path and 2-path max-min assignments becomes little. From Fig. 5-5, 2-path shortest-first assignment generates a lot of "0" capacity assignment. 2-path-shorest-first assignment can achieve maximizing the capacity assignment between the pairs of edge routers, but cannot assign the capacity for some LSPs between the pairs of edge routers.

Fig.s 5-6 and 5-7 show CDF (cumulative distribution functions) for the ratio of the admitted capacity in the cases of 20 and 40 edge routers. The X-axis and Y-axis show the ratio of the admitted capacity and the CDF. These results of the 2-path shortest-first assignment indicate that there are some standard-class LSPs which do not have any admitted capacity. In the other assignments, all the standard-class LSPs between the pairs of edge routers always have some values of the capacity.

In terms of fair accommodation, it is too constraining to take 2-path shortest-first assignment in the operation of NSPs. These results indicate that 2-path shortest-first assignment is impractical for NSPs, while 2- and 3-path max-min assignments are effective in terms of fair

accommodation. Focus on the differences about the ratio of lowest admitted capacity between the 1-, 2- and 3-path max-min assignments in Fig. 5-5, we can say that the lowest value of admitted capacity is improved by setting multiple LSPs between the pairs of edge routers. It also achieves fair accommodation between the pairs of edge routers more effectively. However, when the number of edge routers increases, the effect of setting up the multiple LSPs becomes less.

Focusing on the number of LSPs between the pairs of edge routers, the benefit of 3-path max-min assignment from 2-path max-min assignment is less than that of the benefit from 1-path max-min assignment to 2-path max-min assignment. And, comparing the average admitted capacity of 3-path max-min assignment with that of 2-path max-min assignment, the benefit becomes less worthwhile as the number of edge routers increases. Generally, it is more difficult to set up multiple disjoint paths as the number of LSPs increases between all the pairs of edge routers. It is also generally difficult for service operator to operate the network as the number of LSPs increases. The average admitted capacity of 2-path assignment and 3-path max-min assignment becomes almost same, as the number of edge routers becomes more than 45. Therefore, we consider that the progress of 3-path max-min assignment is less than that of 2-path max-min assignment although the network operation becomes difficult. we conclude that 2-path max-min assignment is applicable in the large network of the carrier network which operates many routers.

The number of variables in LP to solve the capacity assignment increases as the number of routers increases in the simulated topology. However, even with over 60 edge routers in our simulation, the computation time required to solve the modeled LP was generally less than one second. We used an off-the-shelf PC with a 2.00 GHz Intel Core 2 CPU and 1.99 GB memory for the simulation. Therefore, the computational load of our proposed capacity assignment is sufficiently low.

5.6 Related Word

3GPP and 3GPP2 standardizes PCRF (Policy and Charging Rules Function) [58] as the QoS and admission control function in mobile networks. The PCRF has the same role as RACF standardized in ITU-T by utilizing the SIP. However, the issue of how to adapt the PCRF function to the control for the transport stratum also remains unresolved.

Tamura et al. numerically examined that the optimal threshold for commencing traffic distribution over two LSPs and the optimal distribution over the two LSPs that will maximize the admitted traffic among n pairs of edge routers in reference [59]. This study assumed that the pairs of edge routers would initially use single LSP between them, and then begin to use secondary LSP when the traffic exceeds the threshold. This study does not consider multiple traffic class. In this study, the traffic which has high priority and is sensitive for the delay may be transferred into the detour LSPs. This dissertation assumes that primary-class traffic (high priority) is always transferred into the shortest route even if the threshold is exceeded.

5.7 Summary

The RACF which is adopted in NGN knows what a kind of application users will use before their communications. This dissertation utilizes this function to select LSPs. The traffic management cooperating with IMS in MPLS networks was proposed. The existing function of the RACF for MPLS network which has been standardized in ITU-T only considers mapping each traffic class into one LSP by utilizing the function of DS-TE. This dissertation showed the extended

RACF function as the way to utilize multiple LSPs based on the traffic class.

This dissertation showed the definition of traffic class and the policy for the selection of MPLS LSP to utilize our proposed method effectively. In our proposed method, signaling traffic (e.g., SIP messages) and media traffic is prioritized as high priority. Then, the signaling traffic is transferred into the shortest LSP and that the media traffic is transferred into multiple LSPs (shortest, and multiple detour LSPs). To implement our proposed method, it is sufficient to be able to set up LSPs statically between the edge routers in MPLS networks. MPLS routers do not need to have the high functionality like DS-TE in which LSPs automatically adjust the bandwidth and suddenly change the routes.

This dissertation also proposed the dual-phase capacity assignment which assigns the bandwidth for the LSPs between the pairs of edge routers and distributes the standard-class traffic over the multiple LSPs. It achieves maximizing the lowest admitted capacity, and maximizing the remaining bandwidth for the standard-class traffic.

The evaluation compared the effect of the dual-phase capacity assignments with 1-path and 2-path, and 3-path cases between all the pairs of edge routers, and 2-path case in the shortest-first assignment which greedily assigns the capacity just as DS-TE. This dissertation showed that the 2-path case in the shortest-first assignment generated the uneven capacity assignment and is not suitable for the actual network operation. The difference of the average admitted capacity between 2-path and 3-path cases was little, as the number of edge routers increases. And then, in order to apply the proposed method into the large network of the service operator which has many edge routers, This dissertation concluded that dual-phase capacity assignment utilizing two standard-class LSPs between the pair of edge routers has most effective.

The procedures and algorithm for selecting the paths in MPLS networks do not utilize the particular function of the SIP. Therefore,even if the protocol is updated or changed, they can be utilized as it was.

Chapter 6

Proposal of Server Access Control Utilizing Session State Migration Architecture

6.1 Background

As the network architecture to realize the server access control, this dissertation proposes session state migration architecture. This architecture does not focus on the particular protocol. Therefore, the proposed architecture easily can be applied for the server treating the new signaling protocol.

Various services, such as streaming, network games, and thin clients [62] have recently been provided by servers in a datacenter. The increase in user access to these services has caused a rapid increase in the number of servers. The servers in these services retain a session state for each user access. In this dissertation, the session state refers to a representation of application-specific processing status. E.g., the application-specific processing status corresponds to content names and frame numbers for streaming services, and dialog and transaction for SIP [3] proxy servers.

The service operators are faced with the situation that most servers are residual during off-peak periods, e.g., early mornings when user access decreases. However, SPs cannot turn off servers while the servers keep connections with user terminals (UTs) even if the number of connection is small, and cannot obviously reduce energy consumption. Another aspect should be considered for the case when SPs update server software and reboot them. Because SPs need to avoid disrupting services as much as possible, they wait for UTs to complete their service access in each server. There are methods for SPs to estimate the demand from UTs to minimize the number of active servers [63, 64, 65]. However, comprehensive estimates are generally difficult to make, and accidental maintenance by servers cannot be undertaken. For flexible server consolidation, a method of relocating a session state of each UT and switching over to another connection without disrupting the service is needed.

This dissertation proposes a session state migration architecture for flexible server consolidation. The session state migration architecture splits a session state from a connection and bind the session state to another connection in any servers without disrupting services. For this purpose a unique identifier is assigned to each session state. The session state migration architecture enables service providers to conduct server maintenance at their own convenience, and to conserve energy consumption at servers by consolidating them.

One of technical challenges is how to relocate a session state from a connection to another

connection. A conventional server and client application assumes that a session state is statically bound to a connection between the server and UT. If the connection is disrupted, the session states normally disappear and the services are disrupted.

There are some existing studies on server consolidation and session state migration. First, virtual machine (VM) and process migrations [66, 67, 68, 69, 70] are already practically used for server consolidation. However, they cannot contribute to maintaining the software of VMs and processes, because SPs also need to wait for UTs to complete their access to VMs and processes themselves. Moreover, VM migration only relocates a whole service in a VM, and needs sufficient computation resources in physical machines (PMs). We compare session state migration architecture to VM migration in terms of how efficiently servers are consolidated, and reveal that VM migration has fewer chances of executing the server consolidation than session state migration architecture.

Second, "M-TCP" [71] and its extensions [72, 73, 74] relocate a session state from a TCP connection to another one. However, these techniques focus on TCP connections and cannot be applied to UDP connections. There are important services (e.g., SIP and video services) that adopt UDP and manage session states. These studies use TCP option header. However, some deployed firewall products block the traffic with the uncommon TCP option, due to considering it to be a security risk. Third, STEM [75] and ROCK [76] support the switching of connections, they do not take into consideration the relocation of session states. Finally, dedicated methods for HTTP [77] and SIP [78] are also proposed. The session migration architecture is independent of application layer protocols and can be applied to any application.

This dissertation also classifies common procedures of session state migration. The procedure for session state migration can be divided into three steps: 1 switching over the connection from one server to the other, 2 relocating the session states, and installing the session states to the server. Step 3 depends on the applications. Because the former two can have common procedures for various applications, this dissertation designs the required functions.

We reveal that the migration latency caused by the session state migration architecture is smaller than the existing study by analyzing the procedures. By using the implementation [79], this dissertation presents that the session state migration architecture does not give a large impact on a real-time application.

6.2 Issues in Achieving Session State Migration

Figure 6-1 illustrates basic procedures of communication initiating and session state migration. The server and client applications have some fundamental issues in these procedures. A UT and two servers (Server 1 and Server 2) activate the same application for a specific service, and Server 2 succeeds in processing with the session state for the UT. There are two prime procedures: first, the UT establishes a communication path to Server 1, and second, the session state is relocated to Server 2. The procedures of communication initiation and session state migration involve eleven step:

(1) A server application starts in Server 1 and Server 2.
(2) The server applications prepare to receive the connections.
(3) A client application starts in the UT.
(4) The server and client applications establish a connection for communication.
(5) The server and UT applications create session states with each other.
(6) The server and client application begin to exchange user data.
(7) The session state is migrated (the details are explained in Section 3.2).
(8) The server application in Server 2 succeeds in processing the service for the UT.

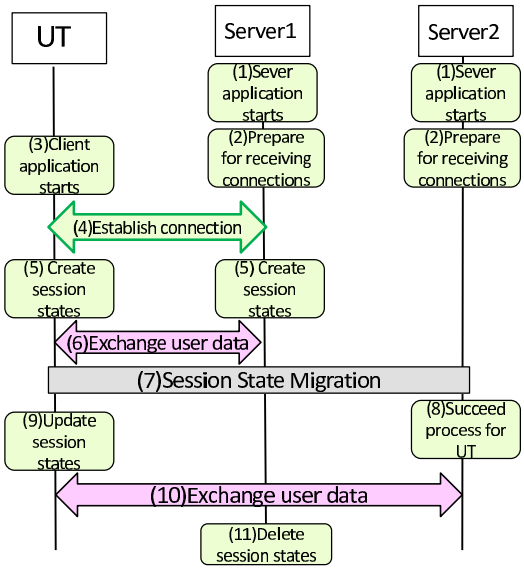


Figure 6.1: Basic procedures of communication initiating and session state migration.

(9) The client application updates the session states because the connection has been switched over.
(10) The server and client applications begin to exchange user data.
(11) The server application in Server 1 deletes the session states of the UT.

The UT, Server 1, and Server 2 need to bind the session states to the connection at steps (5) and (9) whenever connections are established and switched over. However, the conventional server and client application assumes that the session states are statically bound to the connections once the connection has been established. After the session state migration, the connection between Server 1 and UT are switched over to that between Server 2 and UT. The application needs to continue to utilize the same session state as those that were utilized before the session state migration for services to continue. The session state migration architecture needs a function to bind the session state to the TCP/UDP connection. We term this function *"binding of session state to connection"*. The details are described in Section 3.2.

The multiple session states in Server 1 at step (7) need to be relocated, e.g., some video-on-demand (VoD) services utilize multiple connections for the Real Time Streaming Protocol (RTSP) [80], Real Time Protocol (RTP) [81], and Real Time Control Protocol (RTCP)[82] between the server and client. Here, the session state migration architecture needs to simultaneously relocate the multiple session states. In other cases, some services require the server, to which UTs are directly connected, to simultaneously establish other connections with the other server, e.g., a Web server creates a new connection with a database server storing Web content. Therefore, the server needs a function to bind the session state to one or more connections with one or more servers. We term this *"multiple migration support"*. The details are described in Section 3.3.

Further, when the UT switches over the connections from Server 1 to Server 2 at step (7), the session state migration architecture needs to prevent server and client applications from terminating services. If the connection is disrupted, the following output operations are not normally continued. To prevent this, the previous connection should be replaced by the new one without terminating any applications. We term this function *"connection switching"*. The

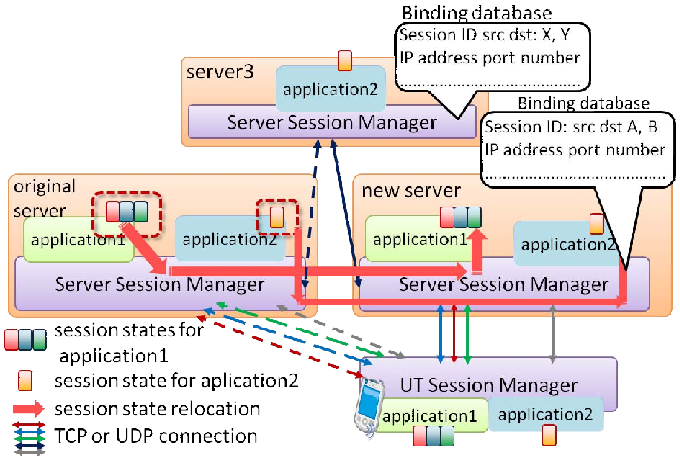


Figure 6.2: Overview of session state migration architecture.

details are described in Section 3.4.

Furthermore, although disruptions to communication cannot be avoided in switching connections at steps (6) through (10), the client and server applications send their user data. Then some user data may be lost. When an application employs TCP connections, it does not assume that packet loss will occur in many cases. Finally, TCP applications cannot recover the lost user data, unlike UDP applications which normally prepare the retransmission mechanism inside. TCP applications in the worst case scenario disrupt their services. Therefore, the session state migration architecture needs a function to block user data while session states are being relocated. We term this *"user data blocking"*. The details are described in Section 3.5.

6.3 Session State Migration Architecture

6.3.1 Overview

Figure 6-2 illustrates the overview of session state migration architecture, where there are two distinct applications: application 1 and application 2. Application 1 has three session states for RTP, RTCP, and RTSP, which have individual connections with the server. The three session states are relocated from one server (termed "original server) to another server (termed "new server"). Application 2 has a session state bound to the two connections between the original server and UT and between server 3 and the original server. The original server has a different IP address from the new server.

We define a session manager (SM), which has a set of required common functions, described in Section 2. The SM is located underneath the server and client applications in each node (the server and client), and it intermediates their communications. The SMs in the server and UT are termed a server session manager (SSM) and a user session manager (USM).

The SMs manages and controls the connections, and the applications exchange their user data through their SMs. When the session states are relocated from the order of the operators as outlined in Figure 6-2, the connections are newly established between the new server and UT.

The SM assigns a unique ID for a session state. We term this ID a *'session ID'*. The session ID is used to distinguish the session state of each UT, which is created by applications. When the session state migration is executed, the operator indicates the session ID for the session

state which the operators want to relocate. When the session states are relocated, the SSM in the original server appends the session ID in the data structures of the session states. Then the SSM in the new server keeps the session ID as the ID of relocated session state.

The SM manages a pair of the session ID and the connection with a binding database. In our proposal, the SSM and USM exchanges session IDs each other when the server and client applications initiate a communication for a specific session state. The procedures for binding of the session state to the connection and exchanging the session IDs are described in the next subsection.

The SSM indicates the session ID of the session state which the USM keeps to the USM, when the SSM requests the switching of connection to the USM. Then, the USM switches over the connection which the session state which is indicated by the SSM is using.

The session ID must be unique among the UTs and servers. The candidate is global unique identifier (GUID) [83] which has 128 bits in length. We propose that MAC address of UTs and servers is used for the prefix of the session ID. This guarantees that the session IDs of the server and UT do not overlap, when they start to communicate. Whether the session ID is unique or not need to be checked when the session state migration is executed. If it overlaps, the SSM in the original server re-assign the session ID for the relocated session state. After this, the session ID remains unchanged even if the session state is relocated, whereas the bounded connections switches.

6.3.2 Binding of Session State to Connection

The function of *"binding of session state to connection"* outlines procedure 1 to assign a session ID for a session state which the SMs in server and UT create, procedure 2 to share these session IDs between the SSM and USM, and procedure 3 to bind these session IDs to the TCP and/or UDP connections at the SSM and USM.Procedure 1,2, and 3 are executed at steps (4) and (7) in Figure 6-1, respectively.

We position that the assignment of a session ID is a task of the SM instead of applications. This is in order for the SM to distinguish the session state with this session ID but not with the application-created identifier which is defined by the application.

Regarding to two types of connections: TCP and UDP, the SM must have individual procedures, because the TCP and UDP uses different procedures in the connection establishment. TCP has a dedicated procedure for the connection establishment before the user data is exchanged whereas UDP has no specific procedure between the server and UT, without which the user data is immediately sent. The SM is required to detect the request of connection establishment in the TCP, and the initial user data sent in the UDP. In the following, this dissertation explains the procedures for binding session states to the TCP and UDP connections with Figure 6-3 and 6-4, respectively.

In the TCP case (Figure 6-3), the session ID between the server and UT should be shared just after the connection is established. This is conducted with a three-way-handshake. Then, the SSM and USM bind their session IDs to the established connection. The binding procedures for the TCP connection involve seven steps:

1. A server application initiates receiving the connection from UTs.

2. The SSM sets the application information (details are explained in the next subsection) for receiving the connection from the application initiation .

3. The UT application initiates to establish a connection.

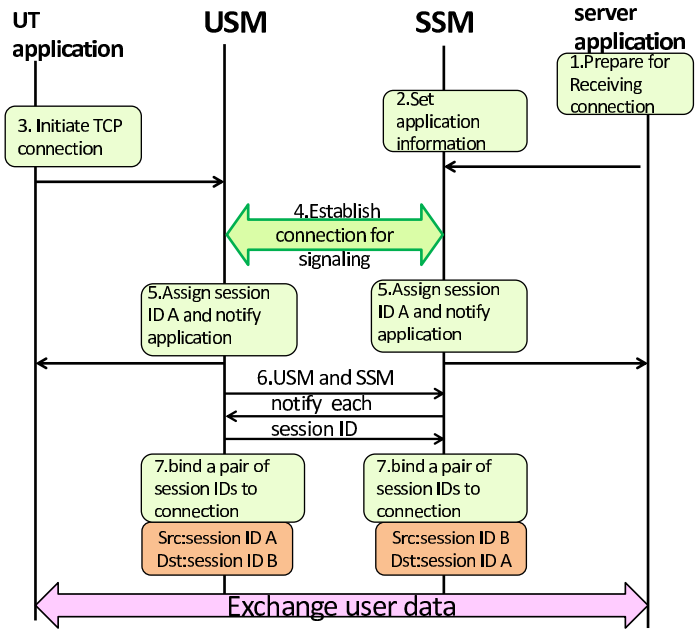


Figure 6.3: Binding procedures for TCP application.

4. The USM begins to establish the connection to the SSM through the TCP standard procedure instead of the server and UT applications.
5. The SSM and USM assign session ID A and B, respectively, and notify them to the individual applications.
6. The SSM and USM conduct three-way handshake to share the session IDs each other.
7. The SSM and USM bind both session IDs, A and B to the connection.

The point is that the server and client application indicate the SSM and USM to prepare for receiving the connection. In the common procedure interacted between a user process and kernel in an operating system (OS), the communication initiation in the applications is corresponded to the binding procedures. We extend the existing system call so as to communicate with the SSM and USM. The details are explained in Section 6.5.2.1.

Figure 6-4 depicts the binding procedure for the UDP connection. In the figure, This dissertation assumes that the UT application sends the user data to the server application. If the role of the UT and server application is changed, the same procedures in an opposite manner should be conducted. The binding procedures for the UDP application involve nine steps:

1. A server application initiates receiving the user data from UTs.
2. The SSM sets the application information (details are explained in the next subsection) for receiving the user data.
3. A UT application sends user data;
4. The USM verifies the user data belongs to the existing session IDs.
5. If the user data does not belong to any session ID, the USM begins to block this and subsequent user data, but otherwise, the user data are sent with the existing session ID.

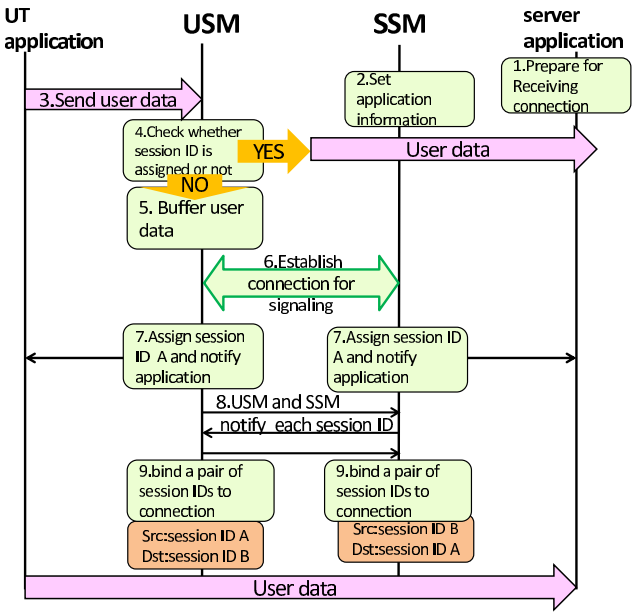


Figure 6.4: Binding procedures for UDP application.

6. The USM establishes a connection for the signaling process.
7. The SSM and USM assign session ID, A and B, and notify them to the individual applications.
8. The SSM and USM notify their individual session IDs each other.
9. The SSM and USM bind both session IDs, A and B to the connection.

The key of UDP case is how to distinguish the initial user data for assigning the session ID at the sender node (the UT in the Figure 6-4's assumption). The UDP application does not have the system call which the TCP application utilizes so as to indicate the SSM and USM to prepare the connection. The USM registers the session ID and connection information with its binding database when the application newly creates the session states with its database. The USM distinguishes whether the user data is sent from a new session state by referring to the binding database or not.

6.3.3 Multiple Migration Support

The function of *"multiple migration support"* is making a group for a set of the session IDs and the multiple connections. This function is executed at step 2 in Figures 6-3 and 6-4. This function enables the session states of which the session IDs are grouped to be simultaneously relocated, when the operators specify a session ID for the session state migration architecture.

When the server and UT application start to communicate and the session state is created in the server and UT, the SSM and USM memorize a set of the session IDs and one or more connection information which the session state uses for the communication. E.g., when the UTs use the video service which prepares the connections for the RTP, RTCP and RTSP, the SM needs to record a set of session IDs for the RTP, RTCP, and RTSP and the multiple connection information for the session state of the UT.

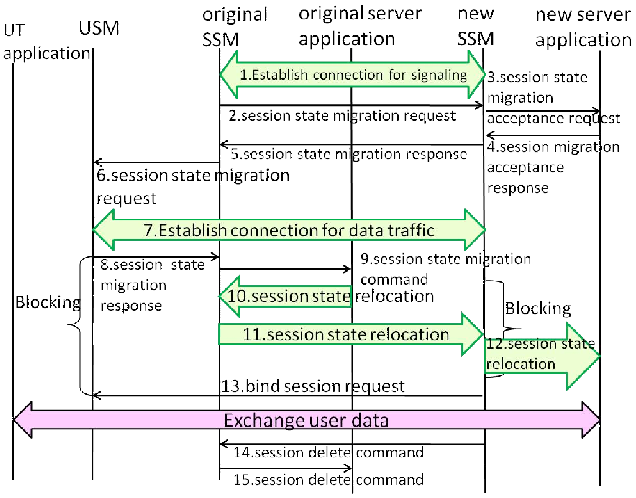


Figure 6.5: Procedures for session state migration architecture.

For this function, the applications need to report to the SSM and USM the application information i.e., which servers the application establishes the connections with, what data structures become the session states and what session states should be grouped. The application developers need to create a code for registering this information with the SSM and USM.

The function of *"switching connections"* switches over connections without disrupting services when the session state migration is executed. This function is executed at step (7) in Figure 6-1.

Figure 6-5 illustrates the procedures for the session state migration architecture. TCP/UDP applications can use the same procedures. However, in UDP applications, the sending and receiving applications can become both the server and UT applications. The thin arrows represent the signaling messages. These procedures are initiated by operators when the server is required to halt for some reason, such as maintenance or reducing the number of servers, and the SSM initiates requests to leave all session states to other servers.

Two connections (steps 1 and 7 in Figure 6-5) are established beforehand in the procedure. The first connection is used to exchange signaling messages and session states between the original and new SSM. This connection is established at the beginning of the procedure to inform the session ID of the session state which will be relocated, and the connection information.

6.3.4 Connection Switching

The second connection for the user data between the USM and new SSM is newly exchanged between the USM and new SSM (step 7). This connection is established before the session states are relocated to minimize the delayed user data caused by the session state migration architecture. This connection takes over the same connection information as the connection between the original SSM and USM except the IP address.

The original SSM in the initial phase of the procedure (steps 2-6) confirms whether the new SSM and USM can accept the session states or not. If the new server application or new SSM rejects them, the procedure terminates at step 5. If they accept them, the new SM check whether the session ID which is used for the relocated session states in the original SM overlaps the session ID in the new SM or not. If it overlaps, the value of the session ID in the original server is changed.

The USM hides the connection switching and forwards the following user data from the UT to the new server application after the session state migration is executed. When the original SSM obtains acceptance from the new server and USM, the procedure enters the middle phase where the session states are relocated.

The session states are relocated in the middle phase (steps 7-13) from the original server application. Then, the relocated session states include the session IDs in order that the new SSM binds those session states to the connections newly established. User data blocking is executed during this phase. The details are described in the next subsection. The procedure goes to the final phase where the user data are normally exchanged.

After the new SSM installs the relocated session states into the new server application, the new SSM joins the session states to the connection between the new SSM and USM by checking the session IDs of the session states. Then, the new server application starts sending user data, and the new SSM informs the USM that the session state migration has completed. Here, the USM releases the blocked user data, and the UT and new server application exchange user data. The new SSM signals the original SSM it has completed in the final phase (steps 14-15). Then, the original SSM shuts down the connection to the USM, and deletes the session states related to the UT.

When the multiple session states need to be relocated for the video services using RTP, RTCP and RTSP, the procedures in the initial phase are executed in parallel. The multiple session states are then simultaneously relocated in the middle phase, and the procedures in the final phase are executed as a group.

When the multiple connections with the multiple servers need to be re-established for the Web services where there are the direct and behind servers, the procedures between the direct server and UT become the same as Figure 6-5. For the procedures between the direct and behind serves, the behind server has a role of UT in Figure 6-5. Because the connections for the signaling and user data were established beforehand and the session states were relocated by the procedures between the UT and direct server, the steps 1, 7, 10, 11, and 12 are omitted.

6.3.5 User Data Blocking

The function of "user data blocking" prevents the packet loss caused by the session state migration architecture, and defines the trigger to block user data in the procedures on the session state migration architecture. This function is executed at step (7) in Figure 6-1. The server application stops sending user data when it is requested to relocate the session state. Then the following user data should be sent from the new server.

User data is blocked in the phase in the left and right curly brackets in Figure 6-5. The USM buffers the user data from its application at steps 8-13. The original SSM relays the session state to the new SSM, and the new SSM also relay it to the new server application.

From step 10 to 12, the user data is not sent to the UT application. The new server application complete installing the relocated session state at step 12, and start to send the user data to the UT application. The connection between the new SSM and USM is established beforehand at step 7. Then, the new SSM can begin to send user data even though the remaining procedure has not completed between the USM and new SSM. This also avoids the loss of user data.

We assumed that the last user data sent from the original server application would reach the UT application before step 13. This can be in time because there are four steps (actual steps 10-13). Further, the last user data that are sent from the UT just before user data are blocked (step 8) can reach the original server application before the request to begin the session state transfers (step 10). Therefore, the user data in both directions can be forwarded without

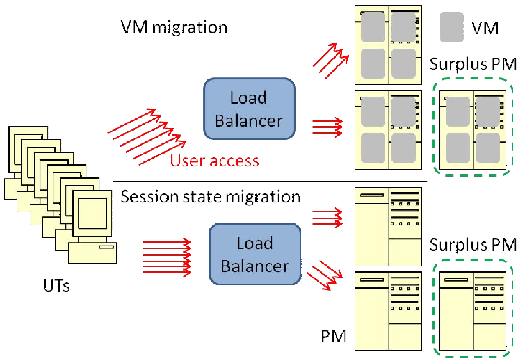


Figure 6.6: Network environment in simulation.

any loss.

6.4 Evaluation of Server Consolidation

6.4.1 Basic Assumption

This section evaluates how many servers are required to accommodate the UTs in some intervals, after the session state migration and VM migration are respectively executed. We compare two situations. In the first situation, a server is implemented in a VM as a guest host and VM migration is executed. One thread in a CPU is supposed to be dedicated for each VM. There are several VMs accommodating UTs in the server. The resource of each VM is isolated like Kernel-based Virtual Machine (KVM) [84] in this simulation. KVM guarantees the isolation of resources between the VMs, when the number of VMs is less than that of threads in a CPU. In the second situation, a server directly accommodating UTs relocates the session states with the proposed architecture.

Considering the current-deployed equipment of servers [85], one server can have 20 threads in a CPU. In this evaluation, this dissertation adopts 20 and 40 as the number of the threads in a CPU of a server. Here, a single server can prepare 19 or 39 VMs as the guest host, and the remaining thread is assigned for the host OS.

In this evaluation, the servers provide a VoD service. The user access is dispatched by a load balancer as drawn in Figure 6-6. Figure 6-6 shows the network environment in this simulation. When UTs are newly connected to the server, the load balancer selects the VM or server where the number of accommodated user is less than the maximum number of accommodated user in a round robin manner. In the simulation, one server is prepared as a surplus server which does not accommodate any UTs and a server is newly setup from the order of the load balancer when the UTs need to be accommodated in the surplus server.

As the maximum number of accommodated user in each VM, 40 and 60 are adopted. We supposed that the server that implements the session state migration architecture could accommodate the same number of UTs in the single server as those where VM migration was executed, preparing 19 or 39 VMs. Then, we compared the number of servers in executing the session state migration and VM migration. When the SM is implemented in the server, the maximum number of accommodated user may be degraded. Because it actually depends on the technique of implementation, anyone cannot know how well the server performs in accommodating users where the SM is installed.

We used the data for hourly user accesses to the VoD services taken from a literature [86] in

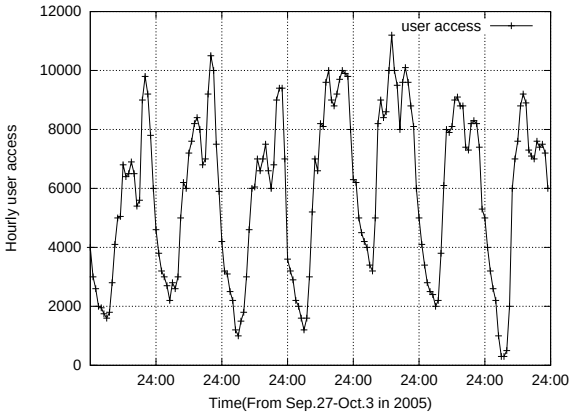


Figure 6.7: Hourly user access to VoD services.

the simulation. Figure 6-7 shows the hourly user access to VoD services. The X-axis means the date when the data is collected, and the Y-axis means the number of user accesses every hour. The simulation period is set to one week. We assumed that 1/60 times user accesses, which are the values for hour on the X-axis of Figure 6-7, would arrive at the servers every minute.

As the length of content in this evaluation, 120 min is adopted. We assumed from the literature [86] that the time 98% of clients spent watching videos excluded zapping and depended on a uniform distribution, where the viewing time ranged from 0 to 96 min. About 2% of clients watched the video content until the end.

In this simulation, the VM migration and session state migration are executed every certain interval. We termed this interval "migration interval". To investigate the influence of the length of migration interval, 60 and 120 minutes as the migration interval are adopted. It is difficult to execute the migration in real time based on user access, because the operators at the service operator also needs to grasp the number of accommodated user in the other servers in real time.

VM migration requires about 3 or 4 min [66],[67] to complete all procedures for the migration, although few packets are lost. Then a large volume of traffic and the large load in the server which accommodates the migrated VM may be generated, although it depends on the memory size of the VM. Thus, the operators at SPs avoid executing VM migration in parallel as they are concerned that multiple faults of the hardware might simultaneously occur. Therefore, we consider that it is not feasible to execute VM migration for a number of servers every less than 60 min.

The session state migration architecture aims to relocate few session states to the other server which has more session states in order that the number of executing the migration is reduced. After the session state migration or VM migration, the servers and VMs which do not accommodate any UT would be turned off in order to reduce the number of servers.

6.4.2 Simulation Results

Figures 6-8 through 11 show the cumulative distribution function (CDF) of required servers for the user access. In each Figure, four cases are evaluated. VM migration is executed in the first and second cases, and the session state migration is executed in the third and fourth cases.

In the first and second cases of Figure 6-8, each server accommodates 19 VMs which accommodate 40 and 60 UTs, respectively as the maximum number of accommodated users. In the third and fourth cases of Figure 6-8, each server accommodates 760 UTs for the former

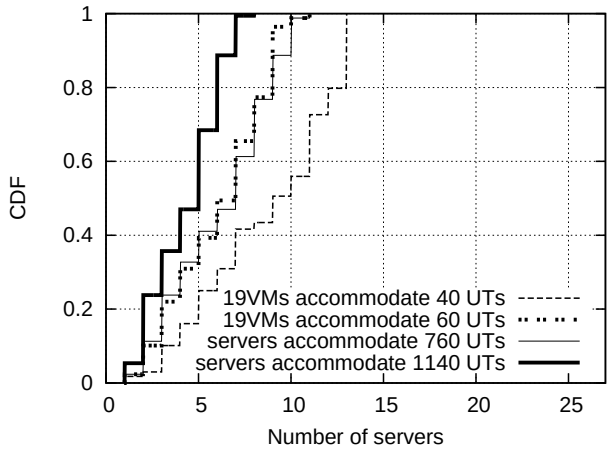


Figure 6.8: Preparing 19 VMs and servers which accommodate 760 and 1140 UTs (migration interval 60 min).

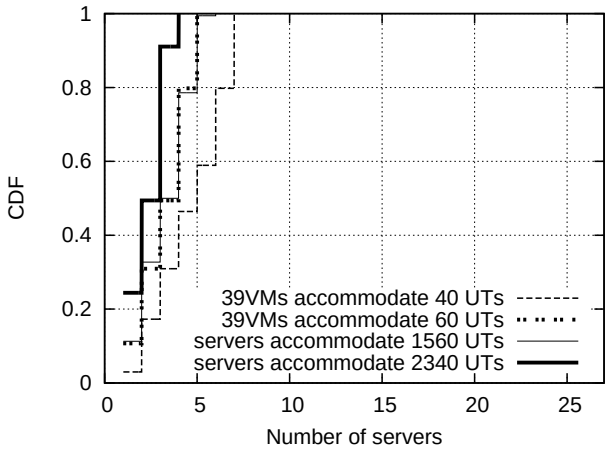


Figure 6.9: Preparing 39 VMs and servers which accommodate 1560 and 2340 UTs (migration interval 60 min).

and 1140 UTs for the latter as the maximum number of accommodated users. In the first and second cases of Figure 6-9, each server accommodates 39 VMs that can accommodate 40 UTs for the former and 60 UTs for the latter as the maximum number of accommodated users. In the third and fourth cases, each server accommodates 1560 UTs (former) and 2340 UTs (latter) as the maximum number of accommodated users. The evaluation in Figures 6-10 and 6-11 are executed in the same way as in Figure 6-8 and 6-9.

Figures 6-8 and 6-9 show that executing the session state migration every 60 minutes reduces the number of servers by about 33%, compared to VM migration, when the single server accommodates the same number of user in the both cases executing the session state migration and VM migration. Figures 6-10 and 6-11 show that executing the session state migration every 120 minutes reduces the number of servers by about 60%, compared to VM migration, when the single server accommodates the same number of user in the both cases. When the VMs and accommodated users in the VM and server increase in number as drawn in Figures 6-8 through 11, the number of servers is also reduced by the rate of increase.

Note that the number of servers does not decrease by the rate of decrease, when the migration interval decreases in the cases of VM migration. When the number of VMs and accommodated user is 39 and 40 respectively, the average number of servers where the migration interval is 60 and 120 min become 4.64 and 7.63, respectively from Figures 6-9 and 6-10. If the value of migration interval becomes half, the number of servers does not become half.

For the VM migration, decreasing the migration interval does not contribute to server consolidation, compared to the number of VMs and accommodated users. In contrast, the number of servers after the session state migration becomes almost the same although the value of the migration interval is different, compared to the cases where the server accommodates 760, 1140 1560, and 2340 UTs as drawn in Figures 6-8 through 11. Because the session state migration architecture distributes the session states into the multiple servers, the resources of all the servers are utilized at maximum.

The session state migration architecture increases the chance of servers consolidating, compared to VM migration. The server can accept accommodating the relocated UTs when the number of accommodate UTs in servers is less than the maximum number of accommodated user. In the session state migration architecture, the resources of the servers are utilized at maximum. On the contrary, VM migration cannot consolidate the servers even when one UT

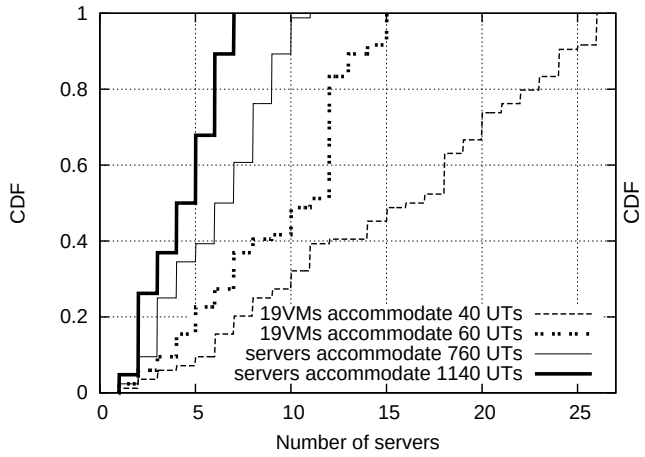


Figure 6.10: Preparing 19 VMs and servers which accommodate 760 and 1140 UTs (migration interval 120 min).

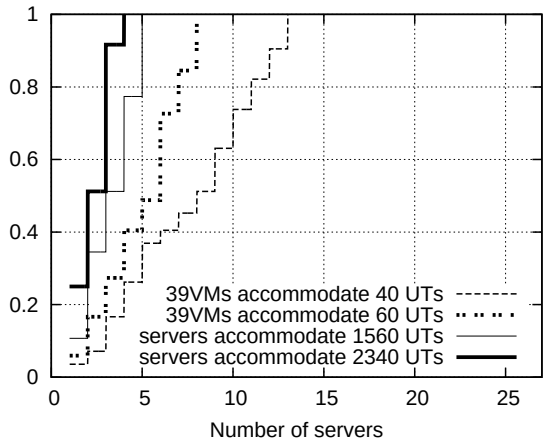


Figure 6.11: Preparing 39 VMs and servers which accommodate 1560 and 2340 UTs (migration interval 120 min).

watches the content in the VM. Here, the resource of the VM is used in vain.

6.5 Implementation

6.5.1 Overview

We implemented the SM as the middleware in Linux OS for a proof of concept. Figure 6-12 illustrates the relationship among an application, kernel and middleware (SM) in the server and client application. The server and UT can have the same implementation as the SM. We focus on two points: one is to minimize the modification to the existing applications and OS. The other is segregating the common procedures related to the session state migration architecture from the program in the application.

The session state migration architecture realizes that the communication between the server and client are continued, even if the IP address of the server is changed. Because the current application program interfaces (APIs) (e.g., socket) does not accept the change of the IP address during the communication, we have to modify the kernel related to those APIs.

The cost for deploying the session state migration architecture is high, and the possibility for it is low, when many parts of the kernel and application program needs to be modified. To reduce the volume of modifications in the kernel and application program, we implemented the common procedures among the various applications in the middleware as much as possible.

The communication between the applications in the server and UT is executed through the kernel in the same way as the existing application. Additionally, we modified the system call in order that the middleware intermediates the communication between the applications in the server and UT. The application in one server communicates with the application in another server through the middleware. As drawn in Figure 6-12, the traffic from the server application goes out through the kernel in the server,and middleware. In Figure 6-12, the unidirectional arrow shows the communication path of the traffic from the session states in the server and UT.

We drew the sockets ($A1$,2,3 and $M1$,2,3,4) used for the communications between application and kernel and middleware and kernel. First $A1$ is used by the application in the original server and $A2$ is used by the application in the UT. When the session state migration is executed, $A1$ is closed, but $A2$ is not closed. Therefore, the application in the UT is not terminated.

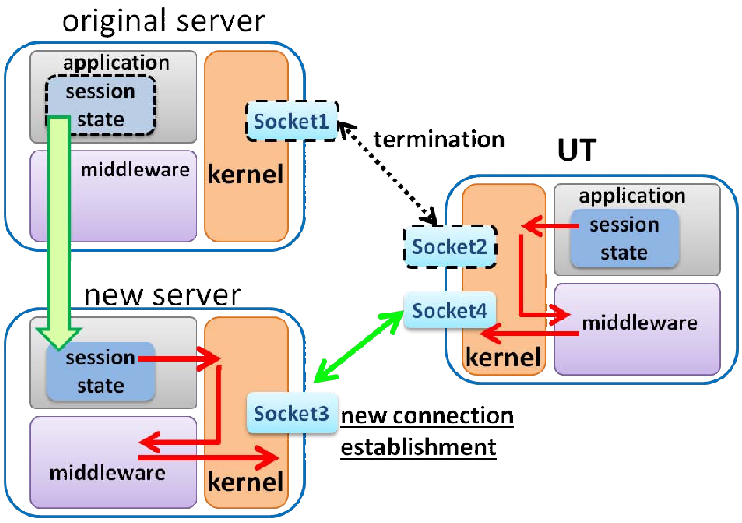


Figure 6.12: Relationship among application, kernel, and middleware (SM).

On the contrary, $M2$ in the original server is closed and $M4$ is newly open. $M4$ is used for the communication between the UT and new server. The application and middleware in the new server newly create $A3$ and $M3$, respectively.

The conventional server and client applications utilize socket API which the OS provides so as to communicate with the other hosts or applications. Beside the middleware, we extend this socket API and minimize the modification to the applications. As the new socket operation, we modify system calls: socket, bind, listen, connect, setsockopt, getsockopt, connect,fcntl and close. With these modified system calls, the communications between the server and UT applications are intermediated by the SSM and USM.

Binding of session state to connection:

We modified connect system calls to notify the initiation of the TCP applications to the SM. When the SM listens to these notifications, the SM assigns a session ID to the session state newly created. For the UDP applications, the SM needs to check whether the session ID has been assigned to user data or not.

When the SSM indicates a session ID to the application in the server so as to request the relocation of session state, the application has to transfer the session state to the SSM. The SSM provides the interface for the application developers to transfer the session states to the SM.

Multiple migration support:

The SM groups a set of session IDs and connection information in the binding database. Through the interface which the SM provides, the application developers need to prepare for reporting the following application information to the SM, i.e., with which servers the application establishes the connections, what data structures become the session states and what session states should be grouped.

Connection switching:

The modified system calls enable the UT application to utilize the same file descriptor even if the UTs change the server to which they are connected. Generally, when the file descriptor is closed, the communication is disrupted. After the session state migration, although the connection is switched over, the USM allows the UT application to continue using write and read system calls with the original file descriptor.

In this implementation, the application in the new server knows the execution of the session state migration by a new signal handler. By fcntl system call, the new signal handler which reports the request of the session state migration is registered with the file descriptor which the application uses for their communication. The developer needs to uses the fcntl system call to register this new signal handler for the session state migration.

User data blocking:

The function of *"user data blocking"* is implemented at the middleware so as to minimize the modification of the applications and kernel, because they do not normally have this function. The same implementation for buffering function in the mobile access gateway (MAG) of Fast Mobile IPv6 [87] and Proxy Fast Mobile IPv6 [88] can be applied for this function. The MAG prepares the buffer to store the user data until the handover procedures complete. As the MAG does, the SSM and USM execute the buffering.

6.6 Performance Evaluation

6.6.1 Migration Latency in Actual Procedures

This subsection analyzes the migration latency which the session state migration architecture caused, by comparing to M-TCP [71]. M-TCP relocates the session state and switches over the TCP connection. Figure 6-14 illustrates the procedures of session state migration in M-TCP. In the evaluation, we assume that M-TCP adopt the function of *"user data blocking"* for preventing packet loss in the same way as the session state migration architecture.

Figure 6-15 shows the topology for the analysis of the migration latency. There are two servers and one UT. The session state is relocated from original server to new server. We investigate the migration latency for the user data from the server and UT when the session state migration architecture and M-TCP are applied. We compare by taking into account the procedures as drawn in Figures 6-5 and 6-14. For the analysis, we define:

- T_{s1} as the transmission and processing delay of the signaling message related to session state migration between original server and UT,
- T_{s2} as the transmission and processing delay of the signaling message related to session state migration between new server and UT,
- T_{ss} as the transmission and processing delay of the signaling message related to session state migration between new and original servers, and
- T_{ssti} as the transmission, installing, and processing delay of the session state from original server to new server.

In M-TCP, a UT initiates a request for session state migration (step $\langle 1 \rangle$ in Figure 6-14) by sending the uncommon TCP option header. Then, the session state is relocated from original server to new server (steps $\langle 2,3 \rangle$) and installed into new server. Finally, the UT and new server check the completion of the session states migration and establishment of the new connection.

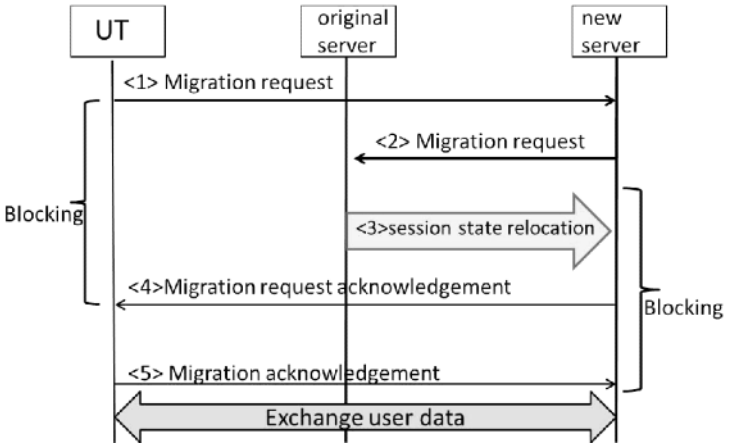


Figure 6.13: Procedures of session state migration in M-TCP.

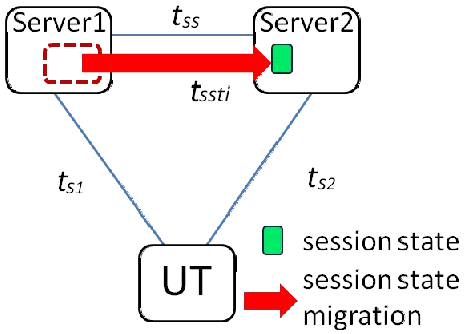


Figure 6.14: Topology for analysis of migration latency.

Here, the connection is switched over to the new connection between the UT and new server (steps ⟨4,5⟩). The migration latency for the user data from UTs in M-TCP (migration Latency for user data from UTs in M-TCP [LUM]) is simply expressed from Figures 6-14 and 6-15 as:

$$LUM = 2T_{s2} + T_{ss} + T_{ssti} \tag{6.1}$$

The migration latency for user data from servers in M-TCP (migration Latency for user data from Servers in M-TCP [LSM]) is simply expressed from Figure 6-14 and 6-15 as:

$$LSM = 2T_{s2} + T_{ssti} \tag{6.2}$$

During the session state migration, the user data from UTs is delayed from steps 8 to 14 in Figure 6-5. The migration latency for user data from UTs (migration Latency for user data from UTs in the session state migration architecture [LUS]) is simply expressed from Figures 6-5 and 6-13 as:

$$LUS = T_{s1} + T_{ssti} + T_{s2} \tag{6.3}$$

During the session state migration, the user data from new server are delayed from steps 10 to 12 in Figure 6-5. The migration latency for user data from servers (migration Latency

for user data from Servers in the session state migration architecture [LSS]) is simply expressed from Figures 6-5 and 6-13 as:

$$LSS = T_{ssti} \tag{6.4}$$

The migration latency in the session state migration architecture is less than that in M-TCP from Eqs. (1)-(4).

Considering the situation where the communication is executed through the wireless LAN, we assume that Ts1 and Ts2 are 100ms. We also assume that the transmission and processing delay of the signaling message related to the session state migration between original and new servers is 10ms. In the session state migration architecture, the values of the migration latency for user data from the UT and server are shorter than that in M-TCP, by 10 and 200 ms, respectively.

M-TCP needs to relocate the session state (step $\langle 3 \rangle$ in Figure 6-14) during the phase where the UT and new server establish the new connection (steps $\langle 3,4,5 \rangle$). Therefore, more duration is required to re-start the communication when M-TCP is adopted.

The session state migration architecture splits the procedures for switching connections and relocating session states, and prepares the connections for the session states before they are relocated. Thus, the migration latency from servers depends only on the duration during which the session states are relocated. We revealed that the splitting mechanism in the session state migration architecture is effective for preventing the degradation of the communication.

6.6.2 Migration Latency of Actual Application

This subsection demonstrates the influence on the real-time application which the session state migration architecture causes by using the implementation described in the previous subsection. We applied the middleware which implements the SM to an actual application, i.e., Video LAN Communication (VLC) [89]. VLC could utilize various protocols to transfer audio and video. We adopted RTP, RTCP, and RTSP as the protocol for transferring media traffic.

We used three PCs that had Ubuntu 11.04 (Linux kernel 2.6.39) installed as the OS, an Intel Core TM i5 3.33 GHz as the CPU, 4 Gb as the memory, and an Intel PRO/100MT as the physical network interface. The topology in this evaluation is same as drawn in Figure 6-13. We assumed that the transmission delay between the PCs would be sufficiently small because the PCs are directly connected.

We also assumed that the two VLC servers, between which the session state is relocated, would have the same content beforehand, and the content itself would not be transferred as the session state. The relocated session state was a pointer to read the content and the data structure for RTP, RTCP and RTSP at the new server.

We investigated the inter-packet gap in media traffic that arrived at the VLC client between the packets that the original server finally sent and those the new server initially sent by using the Wireshark. We carried out the demonstrations 100 times and obtained 11.4 ms as the average and 1.1 ms as the standard deviation. The normal inter-packet gap in media traffic from the VLC server was about 3 ms. User data were delayed by the session state migration architecture for about 11.4 ms.

This delay (11.4 ms) included the transmission, processing and installation delay of the session state from the original server to the new server, as expressed by Eq. (4). Because the transmission delay was sufficiently small, the total processing and installing delays became less than 11.4 ms.

Note that this delay (11.4ms) corresponds to the performance of the middleware, when the middleware receives the session state and bind the session state to the connection established. The acceptable delay in video services is generally required to be about less than 150 ms [90]. When the transmission delay increases between the server and UT, the rate of this delay (11.4 ms) which the middleware caused is sufficiently short at total.

The overhead of the middleware through which the signaling messages and user data are exchanged is not large, although we implement the functions of the SM as the middleware. Through this demonstration, there was no loss of user data and there was no noise in the replayed video. Client applications having packet receive buffers with depths of 11.4 ms or more can hide the migration latency from the user experience, even if the session state migration is executed. This buffer size is sufficiently small, compared to the requirements for decoding media traffic. Therefore, this demonstration shows that the session state migration architecture does not give large impact on the real-time application.

6.6.3 Binding Session IDs

This subsection evaluates the delay for binding the session IDs, when UTs are connected to the server. In the session state migration architecture, the session ID is assigned for each session state and exchanged between the server and UT. This procedure may give the additional delay in the communication between the server and UT, because the existing applications do not adopt this procedure.

By using our implementation, we evaluate the processing time for assigning the session IDs and exchanging them between the server and UT, as the delay for binding the session IDs. We use the two PCs having the same specification as the evaluation in Section 6.2. We prepare the sample TCP/UDP server and client application having the functions of the session state migration architecture. Here, we suppose that the client is connected to the server. Actually, we measured the durations from step 3 to 7 in Figure 6-3 and from step 3 to 9 in Figure 6-4 at 50 times.

We obtained 6.8 ms as the average and 0.2 ms as the standard deviation for the TCP application, 7.4 ms as the average and 0.2 ms as the standard deviation for the UDP application. These times (6.8 and 7.4 ms) include the transmission, and processing of the signaling message and assignment of session IDs. The delay for binding the session IDs in the UDP application is longer than the TCP application, because the SM stores the user data from the UDP application and checks whether the session ID was assigned or not. In the TCP application, the SM can know the initiation of TCP connection, because the TCP application explicitly indicates the initiation of the communication.

If the transmission delay between the server and client becomes longer, these time also becomes much longer. The six exchanges of messages as drawn in Figure 6-3 and 6-4 give the additional delay in the session state migration architecture, compared to the current TCP/UDP application. These exchanges make the UTs wait for the application initiation. However, seeing the current research for the user experience [91] on the waiting time, the user experience does not become worse, if the waiting time is less than 4 seconds. Even if the transmission delay between the server and client is 100ms, the additional waiting time becomes 600ms and the total waiting time is less than 4 seconds. Here, we can say that the cost of binding the session IDs does not give the large impact on the user experience.

6.6.4 Blocking Time

This subsection evaluates the blocking time. We use the two PCs having the same specification as the evaluation in Sections 6.2. Figure 6-16 shows the experimental environment where there

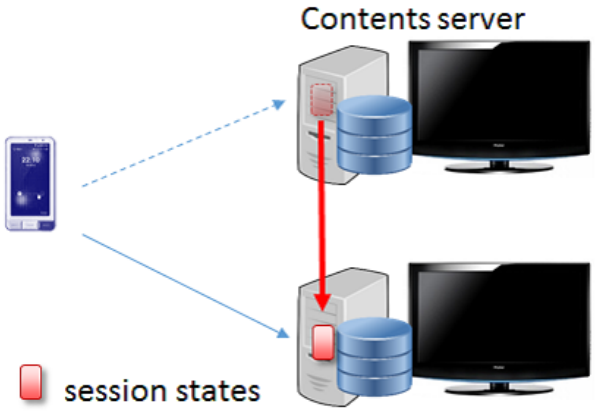


Figure 6.15: Experimental environment.

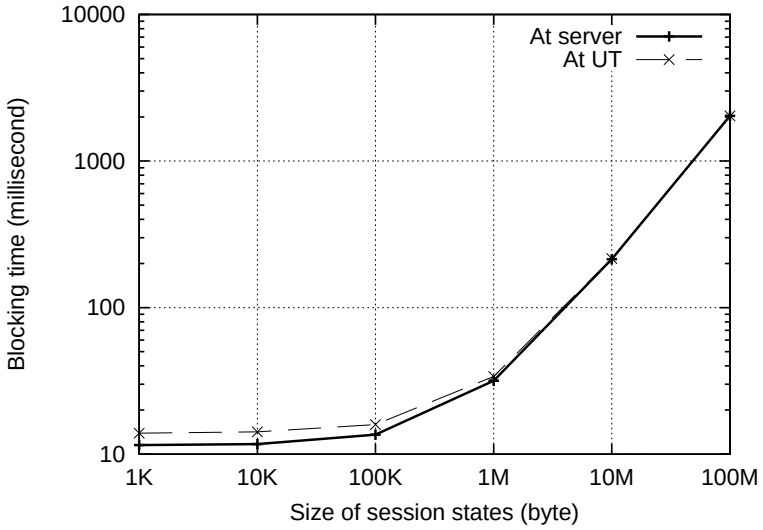


Figure 6.16: Blocking time.

are two content servers. We varied the size of the session states from 1K byte to 100 M byte (1K, 10K, 100K, 1M, 10M, 100M byte), and carried out the evaluation at 50 times. We prepare the only TCP server and client application, because the blocking time becomes same between the TCP and UDP applications.

Figure 6-17 shows the blocking time at the UT and server based on the size of the session states by using log scale in X-axis and Y-axis. When the size of session states is 1K byte, the blocking times at the server and UT become 11.5 and 13.9 ms, respectively. When the size of session states is 100M byte, the blocking times at the server and UT become 2031.7 and 2033.9 ms, respectively. The difference at the UT and server is always almost same (about 2 ms), because only two messages are additionally exchanged during the blocking time at the UT.

The blocking time increases depending on the size of the session states. When the size of the session states is 100 M byte, the throughput is about 393.9 M bps. This value is smaller than the throughput of physical link in our experiment, because this value includes the processing time in the middleware.

Blocking time 213.8 ms is too long for the real-time application as described in [90], when the size of the session states is 10Mbyte. The application program needs to minimize the relocated session states so as to prevent the degradation of the service quality. Then, the server needs to have the same files and contents as the other servers which may receive the session state migration. In this case, the server just needs to transfer the minimum information (e.g., name and location of the contents) as the session states. When the size of that minimum information is less than 100K byte, the blocking time becomes small enough not to give the influence on the service quality of the real-time application.

6.7 Summary

This disseration proposed the session state migration architecture for the flexible server consolidation. The one of technical challenges was to split a session state from a TCP/UDP connection and bind the session state to another connection in any servers, when the session state is relocate. We clarified common procedures and designed the required functions.

This disseration compared the session state migration architecture to VM migration in terms of how more efficiently servers were consolidated in the simulations. The results revealed that the session state migration architecture reduces the number of servers by about 33% and 60%, compared to VM migration when the migration interval is 60 and 120 min, respectively. We found that VM migration had fewer chances of offering the server consolidation than the session state migration architecture.

The migration latency which the session state migration architecture caused is investigated by analyzing the procedures. The analysis presented that the session state migration architecture provided less migration latency than that in existing studies. We reveal that the splitting mechanism in the session state migration architecture is effective for preventing the degradation of the communication.

This disseration demonstrated the implementation of the session state migration architecture in the VLC server and client as a proof of concept. The demonstration realizes binding of the session states to the connections within 11.4 ms with preventing the packet loss. The value does not give the large influence on the video services. Considering these results, the session state migration architecture enables SPs to conduct server maintenance at their own convenience, and conserves energy consumption at servers by consolidating them without disrupting the services, when the application implement the session state migration architecture.

Chapter 7

Conclusion

This dissertation proposed an advanced architecture to improve the availability of the services using the signaling protocol. It was designed to solve the operational and architectural issues, respectively, even when many devices including IoT devices use the signaling protocol.

This dissertation proposes a method that locates the lossy links from the limited information and restoring the servers processing these messages so as to solve the operational issues. The former method contributes to the reduction of the time for trouble shooting. The latter method contributes to the streamlines of the network operation. The analysis shows that the proposed methods satisfies the dependability by detecting the problems related to the signaling process and restoring the servers within the short duration that do not influence the quality of communication.

This dissertation proposes a solution that consists of transferring the traffic into the route having surplus bandwidth and connecting the users to the server having surplus capacity so as to solve the architectural issues. This solution ensures the dependable services enabling users to communicate continuously, even if the scale of the services becomes larger and the unpredictable volume of the traffic is concentrated. This dissertation also shows that the proposed method satisfies the quality required by users by flexibly selecting the routes and servers in the whole network. The former and latter method contribute to the improvement of the congestion tolerance regarding the traffic and signaling messages, respectively.

The result of this research proved that the advanced architecture and operational methods to improve the availability of the signaling-based dependable services in the point of view of the architecture and operational method. In addition, the proposed architecture and methods contribute to make it possible to easily utilize the signaling protocol in the services on the Internet. The dissertation contributes to providing the services using the large number of devices, e.g., IoT services.

7.1 Discussion

Actually, the signaling messages will be upgraded along with the evolution of the applications and services on the Internet,because some new parameters are required for the application and services. However, the basic functions of the signaling protocol will not change. E.g., the retransmission pattern as described in Chapters 3 and 4. The proposed operational methods and network architecture do not follow the particular parameters of the applications and services. Therefore, the dissertation aimed to provide the operational methods and network architecture which could be easily extended for the various applications and services in the various ways in the future.

The proposal described above was evaluated based on the performance requirement of the single network domain. The further study regarding the function to adjust the policy among the multiple network domains is required to provide the signaling-based dependable services on the global Internet. However, the proposed architecture and operational methods evaluated in this dissertation can be easily extended on the global Internet, because the global Internet consists of the multiple network domains and each network domain manages by itself. As the result, this dissertation contributes to a large foundation of the improvement of the availability of the services using the signaling protocol on the global Internet.

7.2 Future Works

The results of this dissertation utilize the numerical analysis, the simulation, and the prototype implementation in order to show the effect of the proposed operational methods and network architecture. When the results are utilized by many people and devices, the research can be finalized and learns to have the value. In this stage, the research regarding this dissertation is not completed.

As the next step, the implementation of the proposed operational methods and network architecture need to be activated in the commercial network. From now on, more and more IoT devices and users are connected into the Internet, and utilize the signaling protocol for their communication. To realize the dependable signaling-based services, I need to introduce the proposed operational methods and network architecture on the Internet.

As the further next step, how to interconnect the signaling protocol among the multiple service operators should be considered. Because the volume of the information on the Internet is too huge, all the information cannot be managed by the signaling protocol in the single domain. The autonomous decentralization of the signaling protocol is required on the Internet. Therefore, the extension for the interconnection of the signaling protocol is required to realize the communication through the whole Internet.

Bibliography

[1] Cisco Systems,"The Internet of Things, How the Next Evolution of the Internet Is Changing Everything,", Cisco Internet Business Solutions Group while paper, April 2011

[2] N. Okada, Y. Tao, Y. Kajitani,"The 2011 eastern Japan great earthquake disaster: Overview and comments,", International Journal of Disater Risk Science, March 2011

[3] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnson, and J. Peterson,"Session Initial Protocol", RFC3261,IETF,
June 2002.

[4] 3rd Generation Partnership Program(3GPP), "IP Multimedia Subsystem (IMS); Stage 2," TS 23.228, 2005.

[5] M. Handley, V.Jacobson and C. Perkins "SDP: Session Description Protcol", RFC 4566,IETF, July 2006

[6] Third Generation Partnership Project, "http://www.3gpp.org" (accessed 2015-02-02).

[7] Standards for M2M and the Internet of Things, "http://www.onem2m.org" (accessed 2015-02-02).

[8] 3GPP, "System Improvements for Machine-Type Communications,"TR 23.888 V1.3.0, June 2011

[9] 3GPP, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access,"TS 23.401.v. 10.4.0, June 2011

[10] P. Bellavista et al., The Future Internet Convergence of IMS and Ubiquitous Smart Environments: an IMS-Based Solution for Energy Efficiency, Elsevier J. Network and Comp. Apps., Special Issue on Intelligent Algorithms for Data-Centric Sensor Networks, May 2011

[11] K.McCloghrie and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", RFC1213, IETF, March 1991.

[12] K.Lingle, J-F. Mule, and J. Maeng, "Management Information Base for the Session Initiation Protocol", RFC4780,IETF, April 2007.

[13] S.Wanke, M.Scharf, S.Kiesel and S.Wahl, "Measurement of the SIP Parsing Performance in the SIP Express Router", in Lecture Notes in Computer Science, A. Pras and M.V. Sinderen eds., Springer, Vol.4606,pp.103-110, 2007.

[14] A.Tachibana, S.Ano, T.Hasegawa, M.Tsuru, and Y.Oie, "Locating Congested Segments on the Internet by Clustering the Delay Performance of Multiple Paths",ICC 2007, June 2007.

[15] M. Coates and R. Nowak, "Network loss inference using unicast end-to-end measurement," Proc. ITC Conference on IP Traffic,Modeling and Management, Monterey, CA, Sept. 2000.

[16] N.G. Duffield, F. Lo Presti, V. Paxson, and D. Towsley, "Inferring link loss using striped unicast probes," Proc. IEEE Infocom 2001, Anchorage, Alaska, April 2001bitemsipmon

[17] A. Kobayashi,H. Ishizuka, N. Tomoeda, T. Sone, E. Kosugi, and A. Iwatani, "VoIP Network Monitoring Solution for IP Telephony Service as Public Communication Infrastructure" Mitsubishi Electoronics technical paper, April 2006..

[18] ClearSight Analyzer, Product in ClearSight Networks "http://clearsightnet.org/productInfo.php?ProductID=3" (accessed 2015-02-02).

[19] R. Presuhn, J. Case, K. McCloghrie ,M. Rose, and S. Waldbusser, "Management Information Base for the Simple Network Management Protocol", RFC3418, IETF, December 2002.

[20] MRTG, "http://oss.oetiker.ch/mrtg/" (accessed 2015-02-02).

[21] "Telecom data book 2008" Telecommunications Carriers Association in Japan "http://www.tca.or.jp/databook/index.html" (accessed 2015-02-02).

[22] J. Rosenberg and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol", RFC3262, IETF, June 2002.

[23] M. Volk, I. Humar, and A. Kos, "Empirical performance evaluations of peer-to-peer VoIP telephony using SIP," IEEE EUROCOM, 2009

[24] J.L Gonzalez, and R. Marcelin Jimenez., "Phoenix: A Fault-Tolerant Distributed Web Storage Based on URLs," Parallel and Distributed Processing with Applications (ISPA), 2011

[25] F. Machida, M. Kawato, and Y. Maeno, "Redundant virtual machine placement for fault-tolerant consolidated server clusters," Network Operations and Management Symposium (NOMS), 2010

[26] A. Kamalvanshi and T. Jokiaho, "Using OpenAIS for Building Highly Available Session Initiation Protocol (SIP) Registrar," Service Availability, vol. 4328/2006, Springer Berlin/Heidelberg, pp.217-228, 2006.

[27] T. Renier, H. Schwefel, M. Bozinovski, K. Larsen, R. Prasad, and R. Seidl, "Distributed redundancy or cluster solution? An experimental evaluation of two approaches for dependable mobile internet services," In Proceedings of the First international conference on Service Availability, 2004

[28] T. Usui, Y. Kitatsuji, H. Yokota, and N. Nishinaga, "Restoring CSCF by Leveraging Feature of Retransmission Mechanism in Session Initiation Protocol," IARIA Emerging 2011, November, 2011

[29] E. Dubrova, "Fault-Tolerant Design," ISBN-10 1461421128, Springer, 2013

[30] M. Ohta, "Overload Protection in a SIP Signaling call flow," Internet Surveillance and Protection, 2006

[31] International Telecommunication Union, “ Network grade of service parameters and target values for circuit-switched services in the evolving ISDN, ” Recommendation E.721, Telecommunication.

[32] Free BSD 9.1-RELEASE Ping command

[33] A. K. SINGH, and A. KOTHARI, "HSRP (Hot Stand by Routing Protocol) Reliability Issues Over the Internet Service Provider's Network," ORIENTAL JOURNAL OF COMPUTER SCIENCE & TECHNOLOGY, 2011, Vol. 4, No. (2)

[34] J. Rosenberg, H. Schulzrinne, and G. Camarillo, "The Stream Control Transmission Protocol (SCTP) as a Transport for the Session Initiation Protocol (SIP)," IETF RFC 4168, Oct 2005

[35] The Open IMS Core Project, "http://www.openimscore.org" (accessed 2015-02-02).

[36] The SIP Express Router, "http://www.iptel.org/ser" (accessed 2015-02-02).

[37] Telecom Data Book 2010, Telecommunications Carriers Association (Japan), "http://www.tca.or.jp/databook/index.html" (accessed 2015-02-02).

[38] Oki Electric Industry Co., Ltd, CenterStage NX5100 series ,"http://www.oki.com/jp/centerstagenx/product/nx5000.html" (commercial product of P-CSCF) (accessed 2015-02-02).

[39] "Nokia Siemens Networks LTE-capable transport: A quality user experience demands an end-to-end approach," White paper, Nokia Siemens Networks

[40] E. Dahlman, A. Furuskar, A. Kangas, M. Lindstrom, and S. Parkvall, LTE: the evolution of mobile broadband, IEEE Communication Magazine, 2009

[41] ITU-T Y.2001 (2004),"General overview of NGN"

[42] 3GPP TS 23.228 (2005), "IP Multimedia Subsystem (IMS); Stage 2".

[43] ETSI ES 283 003, V1.1.1 (2001), "Telecommunications and Internet Converged Services and Protocols for Advanced Networking (TISPAN); IP Multimedia Call Control Protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP) Stage 3".

[44] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnson, and J. Peterson (2002),"Session Initial Protocol", IETF RFC3261.

[45] ITU-T Y.2111 (2006), "Resource and admission control functions in next generation networks".

[46] E. Rosen, A. Viswanathan, R. Callon (2001), "Multiprotocol Label Switching Architecture", IETF RFC3031.

[47] D. Awduche, J. Malcolm and J. Agogbua (1999), "Requirements for Traffic Engineering Over MPLS", IETF RFC2702.

[48] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan and G. Swallow (2001), "RSVP-TE: Extensions to RSVP for LSP Tunnels", IETF RFC 3209.

[49] F. Le Faucheur, W. Lai (2003), "Requirments for Support of Differentiated Service Aware MPLS Traffic Engineering",IETF RFC 3564.

[50] Aumatic Bandwidth Adjustment for TE tunnels, "QoS for IP/MPLS networks" Cisco Press ISBN 1587052334

[51] H. Ishi, K. Nagami (2004), "Proposal and evaluation for Automatic bandwidth setting in MPLS LSP" IEICE B ,J89-B,10, 1894-1901.

[52] K.Nicols, S.Blake, F.Baker (1998), " Definition of the Differentiated Services Field (DS Field)in the IPv4 and IPv6 Headers", IETF RFC2474.

[53] ITU-T Y.2174 (2008), "Distributed RACF Architecture for MPLS Networks"

[54] B.Martini, F.Baronceli, V.Martini, K.Torkman, P.Castoldi (2009), "ITU-T RACF implementation for application-driven QoS control in MPLS networks",Integrated Network Management(IM)'09. IFIP/IEEE international Symposium,422-429

[55] C. Srinvasan, Bloomberg L.P. (2004), "Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)", IETF RFC 3812.

[56] A. Medina and I. Matta (2000), "BRITE:A Flexible Generator of Internet Topologies", Tech. Rep. BU-CS-TR-2000-005, Boston University, Boston MA.

[57] R. Albert and A. Barabasi (2002), "Statistical mechanics of complex networks", Review of Modern Physics,74,47-97.

[58] 3GPP2 X.S0013-007-A (2005), "All-IP Core Network Multimedia Domain".

[59] H. Tamura, K. Kawahara, Y. Oie (2004), "Analysis of two-phase Path Management Scheme for MPLS Traffic Engineering", Proc. Of SPIE Performance Quality of Service, and Control of Next Generation Communication Networks2,194-203.

[60] S. Zaghloul, A. Jukan, W. Alanqar (2007), "Extending QoS from Radio Access to an All-IP Core in 3G Networks: An Operator's Perspective", IEEE Communications Magazine,45(9),124-132.

[61] F. Wegscheider (2006), "Minimizing unnecessary notification traffic in IMS presence system", 1st International Symposium on Wireless Pervasive Computing,6-12.

[62] Bernd Oliver Christiansen, and Klaus Erik Schauser, "Novel Codec for Thin Client Computing," Proceedings of the Conference on Data Compression, p.13, March 28-30, 2000.

[63] J.Rolica, A.Andrejak, M. Arlitt, "Automating Enterprize Application Placement in Resource Utilities," in Proc. IFIP/IEEE Distributed Systems Operations and Management, Heidelberg, Germany, pp. 118-129, 2003.

[64] K.Mizutani, T. Mano, O. Akashi, T. Kawano and H. Shimizu, "Streaming Server Management Scheme for Reducing Power Consumption," IEEE Globecom,2002.

[65] D. Gmach, S.Krompas, A.Schloz, M.Wimmer and A.Kemper, Adaptive Quality of Service Management for Enterprise Services, in ACM Transactions on the Web, vol.2, no.1 2008.

[66] C.Clark,K.Fraser,S.Hand, J.G. Hansen, E.Jul, C.Limpach , I.Pratt , and A.Warfield, "Live migration of virtual machines," Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation, p.273-286, May, 2005.

[67] M. R. Hines and M. Gopalan, "Post-Copy Based Live Virtual Machine Migration Using Adaptive Pre-Paging and Dynamic Self-Ballooning," ACM SIGPLAN/SIGOPS 2009.

[68] R. Bradford, E.Kotsovinos, and H. Schioberg, "Live wide-area migration of virtual machines including local persistent state," In Proc. of the International Conference on Virtual Execution Environments 2007.

[69] B. Gerofi, H. Fujita and Y. Ishikawa, "Live Migration of Processes Maintaining Multiple Network Connections", IPSJ online transaction, vold.3 13-24, Marchi, 2010

[70] F.Douglis and J. Ousterhout, "Transparent Process Migration: Design Alternatives and the Sprite Implementation," Software- Practice and Expericence, vol.21, no.8, pp.757-785, August 1991.

[71] F. Sultan, K.Srinivasan, D.Iyer, and L.Iftode, "Migratory TCP: Connection Migration for Service Continuity in the Internet," ICDCS 2002.

[72] F. Sultan, A. Bohra, and L.Iftode, "Service Continuations: An Operating System Mechanism for Dynamic Migration of Internet Service Sessions," SRDS 2003.

[73] F. Sultan, A. Bohra, and L. Iftode, "Autonomous Transport Protocols for Content-based Networks," Rutgers University Technical Report DCS-TR-479, March 2002.

[74] F. Sultan, K. Srinivasan, D. Iyer, and L. Iftode, "Migratory TCP: Highly Available Internet Services Using Connection Migration," Rutgers University Technical Report DCS-TR-462, December 2001.

[75] F. Ansari, and A. Sathyanath, "STEM: seamless transport endpoint mobility," ACM SIGMOBILE Mobile Computing and Communications Review, Vol.11 Issu2, April, 2007.

[76] V.C.Zandy, and B.P.Miller, "Reliable network connections," ACM Mobicom 2002.

[77] M. Takahashi, A.Kohiga, T.Sugawara, and A. Tanaka, "TCP Migration with Application-Layer Dispatching: A New HTTP Request Distribution Architecture in Locally Distributed Web Server Systems," Comsware 2006.

[78] C. Makaya, S. Das, D. Chee, J. Lin, S. Komorita, T. Chiba, H. Yokota, H. Schulzrinne, "Self organizing IP Multimedia Subsystem," IEEE IMSAA, 2009.

[79] T. Usui, K. Nakauchi, Y. Shoji, Y. Kitatsuji, H. Yokota, N. Nishinaga, "Design of Session State Migration Middleware for Disruption-free Services," Sixth International Conference on Next Generation Mobile Applications, Serivces and Technologies (NGMAST), September, 2012

[80] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol," IETF RFC 2326, April 1998

[81] H. Schulzrinne, G. Fokus, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," IETF RFC 3550, July, 2003

[82] C. Huitema, "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", IETF RFC 3605, October, 2003

[83] E. Laure, S. M. Fisher, A. Frohner, C. Grandi, P. Kunszt, A. Krenek, O. Mulmo, F. Pacini, F. Prelz, J. White, M. Barroso, P. Buncic, F. Hemmer, A. Di Meglio, A. Edlund, "Programing the Grid with gLite," COMPUTATIONAL METHODS IN SCIENCE AND TECHNOLOGY, vol 12, No. 1, pp. 33-45, 2006.

[84] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, "KVM: the linux virtual machine monitor,"in Proceeding of the 2007 Ottawa Linux Symposium, 2007.

[85] Intel CPU Comparison, "http://www.intel.com/content/www/us/en/processor-comparison/compare-intel-processors.html" (accessed 2015-02-02).

[86] H. Yu, D. Zheng, B.Y.Zha and W. Zheng, "Understanding User Behavior in Large-Scale Video-on-Demand Systems", ACM EuroSys 2006.

[87] R. Koodli, "Fast Handover for Mobile IPv6,", IETF RFC 4068,2007

[88] H. Yokota, K. Chowdhury, R. Koodli, B. Patil, and F. Xia, " Fast Handovers for Proxy Mobile IPv6," IETF RFC 5949, 2010.

[89] Video LAN Communication(VLC), "http://www.videolan.org/vlc" (accessed 2015-02-02).

[90] "Telecommunications and Internet Protocol Harmonization over networks release3: End to end quality of service in TIPHON systems Technical Report," ETSI TR 101 329-7 V2.1.1, Feburary 2002

[91] S.Niieda, S.Uemura, H. Nakamura, E.Hara, "Field study of a waiting-time filler delivery system," MobileHCI, 2011

Publication List

Referred Journal

1. Takeshi Usui, Takeshi Kubo, Yoshinori Kitatsuji, Hidetoshi Yokota, "A Study on Locating Lossy Links of Signaling Messages in SIP-based Services", IEICE Transactions on Communications, Vol. E94-B, No.1, pp. 118-127, 2011, January

2. Takeshi Usui, Yoshinori Kitatsuji, Hidetoshi Yokota, "A study on traffic management cooperating with IMS in MPLS networks", Telecommunication Systems Journal, Volume 52, Issue 2, pp671-680, 2013, Feburary

3. Takeshi Usui, Yoshinori Kitatsuji, Hidetoshi Yokota, Kiyohide Nakauchi, Yozo Shoji, Nozomu Nishinaga, "A Session State Migration Architecture for Flexible Server consolidation", IEICE transactions on Communications, Vol. E96-B, No.7, pp.1727-1741, 2013, July

4. Takeshi Usui, Yoshinori Kitatsuji, Hidetoshi Yokota, Kiyohide Nakauchi, Yozo Shoji, Nozomu Nishinaga, "An IMS Restoration System with Selective Storing of Session States", IEICE transactions on Communications, Vol. E97-B, No.9, pp.1853-1864 2014, September

International Conference

1. Takeshi Usui, Yoshinori Kitatsuji, Teruyuki Hasegawa, Hidetoshi Yokota, "On harmonizing IMS with MPLS-based traffic engineering", IWIN(International Workshop on Informatics) 2008, September

2. Takeshi Usui, Yoshinori Kitatsuji, Hidetoshi Yokota, "On Reducing Packet Buffering for Network-based Fast Handover Cooperating with IMS", IEEE IMSAA 2009, December

3. Takeshi Usui, Takeshi Kubo, Yoshinori Kitatsuji, Teruyuki Hasegawa, Hidetoshi Yokota, "On Locating Loss Links of Signaling Messages in SIP-based Services", IEEE Globecom 2009, December

4. Takeshi Usui, "Virtual Network Mobility:Advanced Mobility Management over Network Virtualization", 3rd European Research EU Japan, 2010, October

5. Takeshi Usui, Kiyohide Nakauchih, "Virtual Network Mobility Toward Future Networks", Wireless World Research Forum (WWRF) 2010, November

6. Takeshi Usui, "Designing Improved Traffic Control in Network-based Seamless Mobility Management for Wireless LAN", IARIA AFIN 2011, August

7. Takeshi Usui, Yoshinori Kitatsuji, Hidetoshi Yokota, Nozomu Nishinaga, "Restoring CSCF by Leveraging Feature of Retransmission Mechanism in Session Initiation Protocol", IARIA Emerging 2011, November

8. Takeshi Usui, Yoshinori Kitatsuji, Hidetoshi Yokota, Kiyohide Nakauchi, Yozo Shoji, Nozomu Nishinaga, "Design and Implementation of Session State Migration Middleware for Disruption-Free Services", Next Generation Mobile Applications, Services and Technologies (NGMAST) 2012, September